Jagiellonian University

Doctoral School of Exact and Natural Sciences
Faculty of Mathematics and Computer Science

Ph.D. Thesis

Learning to look and search: Efficient visual exploration for embodied agents

mgr Adam Pardyl

Under the supervision of
dr hab. Bartosz Zieliński, prof. UJ

Kraków, 2026

This thesis would not have been possible without the people who accompanied me throughout my doctoral journey.

To my family, for always being there, for their patience, encouragement, and unconditional support.

To my supervisor, Prof. Bartosz Zieliński, for his mentorship, confidence in me, and for helping me develop both as a scientist and as an independent researcher.

And to my co-authors, whose knowledge, curiosity, and collaboration shaped much of the work presented in these pages. It has been a privilege to learn and work alongside you.

Abstract

Modern computer vision is built on the premise that a model is handed all the visual information it needs in advance, as a complete, static, sufficiently high-resolution image. This assumption is reasonable for tasks such as classification or segmentation, but it breaks down the moment a model is placed inside an embodied agent, such as a robot or an unmanned aerial vehicle. An embodied agent perceives the world through a narrow window with a limited field of view and resolution. Hence, it must move physically to acquire more observations while operating under tight constraints on time, energy, and computation. Capturing and processing an entire environment at full resolution is not merely wasteful but often infeasible. Under these conditions the central question is no longer only *what we currently see*, but rather *where we should look or move to obtain the most informative observation*. By choosing its subsequent observations wisely, an agent can match the performance of one that senses everything indiscriminately, at only a fraction of the cost.

This dissertation concerns *active visual exploration* at two complementary scales: the *local scale* of deciding where to direct and focus a sensor within the surrounding scene and the *global scale* of deciding where to move through an environment to acquire new observations. The aim throughout is to build agents that are *efficient*, understanding an environment from as few observations as possible; *flexible*, exploiting the continuous capabilities of real sensors rather than artificial grids; and *scalable* toward the demands of real embodied agents. The dissertation addresses these challenges through four research hypotheses investigated in four publications.

The first contribution, *Active Visual Exploration Based on Attention-Map Entropy* (IJCAI 2023), asks how an agent should decide where to look next without a separate decision network. It shows that the internal uncertainty of a transformer, read directly from the entropy of its attention maps, is itself an effective signal for selecting the most informative next glimpse, simplifying training while improving reconstruction, classification, and segmentation from partial observations. This work, however, draws its glimpses from a fixed grid of patches of the same size. The second contribution, *Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers* (WACV 2025), lifts that constraint. It formalises the notion of input *elasticity* as the resilience of a model to patches of varying scale, position, and coverage. It then introduces a protocol to measure this resilience and proposes architectural and training modifications that let a vision transformer ingest patches sampled freely from an image, turning the rigid grid into a continuous sampling space. The third contribution, *AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale* (ECCV 2024), brings observation selection closer to the capabilities of real sensors. Building on an input-elastic transformer, it casts exploration as a Markov decision process and uses the Soft Actor-Critic algorithm to select glimpses of arbitrary continuous position and scale. Such degrees of freedom are offered by motorised pan-tilt-zoom cameras. The resulting agent discovers a coarse-to-fine observation strategy on its own, matching or surpassing prior methods while requiring fewer exploration steps to complete the task.

The first three papers progressively generalise *local-scale* exploration, moving from a fixed grid, through elastic patch geometry, to arbitrary continuous camera zoom and rotation. The fourth contribution, *FlySearch: Exploring how vision-language models explore* (NeurIPS 2025), turns to the complementary *global-scale* problem of deciding where to move in a large, open, three-dimensional world in order to find

something. It introduces a photorealistic benchmark for goal-driven object search, in which a simulated aerial vehicle must physically traverse a vast outdoor search area to acquire each observation, and uses it to ask whether today's vision-language models can carry out this kind of exploration. The evaluated vision-language models are able to perform short-horizon navigation but struggle with systematic search when the target is initially unseen. The tested fine-tuning improves short-horizon navigation, including transfer across environments, but does not resolve these systematic-search failures.

Taken together, the first three contributions connect an internal signal for observation selection, an architecture that processes flexibly sampled observations, and a learned policy that selects their position and scale. Within the evaluated settings, they progressively extend efficient local exploration beyond a fixed sampling grid. The fourth contribution introduces a benchmark that exposes the additional difficulty of organising observations and movements into sustained, goal-driven search in large outdoor environments. These findings motivate further research into sustained search strategies, transfer to physical environments, and explanations of the visual evidence guiding an agent's decisions.

All four publications appeared in the main tracks of leading conferences, three at CORE A* (200 MNiSW points) conferences and one at a CORE A (140 MNiSW points) conference, and I am the first author of each of them. The research was supported in part by the National Science Centre Preludium grant *Where to look next – guiding active visual exploration with internal model uncertainty*, of which I was the principal investigator. I also received the START scholarship of the Foundation for Polish Science. Moreover, during my doctoral studies I completed a research internship at the Computer Vision group of the Max Planck Institute for Informatics under the supervision of Prof. Bernt Schiele. Beyond the main line of research, I contributed to work on AI-assisted medical diagnostics (which led to a European patent application), on explainable artificial intelligence, and on benchmarking the spatial reasoning of vision foundation models. I also served as a reviewer for NeurIPS, IJCV, TPAMI, RA-L, WACV, and KDD, and co-organised three editions of the Machine Learning Summer School and the ELLIS Doctoral Symposium.

Keywords: active visual exploration, embodied vision, active perception, vision transformers, reinforcement learning, vision-language models, object-goal navigation.

Streszczenie

Współczesne widzenie komputerowe w znaczącym stopniu bazuje na założeniu, że model otrzymuje od razu wszystkie potrzebne informacje w postaci kompletnego, statycznego obrazu o odpowiednio wysokiej rozdzielczości. Podejście to sprawdza się w zadaniach takich jak klasyfikacja czy segmentacja. Przestaje jednak być wystarczające, gdy model ma sterować robotem lub bezzałogowym statkiem powietrznym, działającym w rzeczywistym otoczeniu. Taki agent w danym momencie widzi tylko część otaczającego go świata, a szczegółowość obserwacji zależy od rozdzielczości jego kamery i jej pola widzenia. Aby zdobyć kolejne informacje, musi aktywnie kierować kamerą lub zmieniać swoje położenie, dysponując przy tym ograniczonym czasem, energią i zasobami obliczeniowymi. Rejestrowanie i przetwarzanie całego otoczenia z najwyższą dostępną rozdzielczością jest w tych warunkach kosztowne, a często wręcz niemożliwe. Istotne staje się zatem nie tylko to, *co widzimy obecnie*, lecz także *gdzie spojrzeć lub dokąd się przemieścić, aby uzyskać najbardziej przydatne informacje*. Trafna selekcja kolejnych obserwacji pozwala osiągać wyniki porównywalne z rezultatami metod przetwarzających wszystkie dostępne dane, przy znacznie mniejszych kosztach.

Niniejsza rozprawa dotyczy *aktywnej eksploracji wizualnej* (ang. active visual exploration) w dwóch uzupełniających się skalach. W *skali lokalnej* agent decyduje, na którą część otaczającej go sceny skierować kamerę i jakie zastosować przybliżenie. W *skali globalnej* wybiera natomiast, dokąd się przemieścić, aby uzyskać nowe obserwacje. Celem badań jest opracowanie metod, które pozwolą agentom poznawać otoczenie na podstawie możliwie niewielu obserwacji, swobodnie korzystać z możliwości ruchu i zmiany przybliżenia oferowanych przez rzeczywiste kamery oraz sprostać wymaganiom eksploracji rozległych, otwartych środowisk. Rozprawa podejmuje te wyzwania poprzez cztery hipotezy badawcze analizowane w czterech kolejnych publikacjach.

Pierwsza publikacja, *Active Visual Exploration Based on Attention-Map Entropy* (IJCAI 2023), dotyczy wyboru kolejnej obserwacji bez uczenia osobnej sieci decyzyjnej. Pokazuje, że entropia map uwagi transformera, będąca miarą jego wewnętrznej niepewności, może skutecznie wskazywać, który fragment sceny warto obejrzeć w następnym kroku. Wykorzystanie tego sygnału upraszcza uczenie, a zarazem poprawia wyniki rekonstrukcji, klasyfikacji i segmentacji na podstawie niepełnych obserwacji. Metoda ta wybiera jednak fragmenty o jednakowym rozmiarze, rozmieszczone na ustalonej siatce.

Druga publikacja, *Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers* (WACV 2025), przedstawia sposób przezwyciężenia tego ograniczenia. Wprowadza pojęcie *elastyczności wejścia*, czyli zdolności modelu do zachowania jakości predykcji przy zmianach położenia i skali fragmentów wejściowych oraz stopnia pokrycia nimi obrazu. Proponuje protokół pomiaru tej właściwości, a także modyfikacje architektury i sposobu uczenia modelu typu Vision Transformer, dzięki którym może on przetwarzać fragmenty obrazu wybierane poza regularną siatką. Pozwala to przejść od sztywnego układu danych wejściowych do swobodnego doboru ich położenia i skali.

Trzecia publikacja, *AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale* (ECCV 2024), przybliża sposób wyboru obserwacji do możliwości rzeczywistych kamer. Korzystając z transformera o elastycznym wejściu, ujmuje eksplorację jako proces decyzyjny Markowa. Agent uczony za pomocą algorytmu Soft Actor-Critic wybiera położenie i skalę kolejnych obserwacji w przestrzeni ciągłej. Taki zakres swobody odpowiada zdolnościom kamer umożliwiających obrót w poziomie i pionie

oraz zmianę przybliżenia (pan-tilt-zoom). Agent samodzielnie uczy się strategii, w której najpierw ogląda scenę z szerszej perspektywy, a następnie skupia się na istotnych szczegółach. Dzięki temu dorównuje wcześniejszym metodom lub je przewyższa, wykonując mniej kroków eksploracji.

Pierwsze trzy prace stopniowo rozszerzają możliwości eksploracji w *skali lokalnej*: od wyboru fragmentów z ustalonej siatki, przez swobodny dobór ich geometrii, po płynne sterowanie kierunkiem obserwacji i przybliżeniem. Czwarta publikacja, *FlySearch: Exploring how vision-language models explore* (NeurIPS 2025), dotyczy *skali globalnej*, czyli przemieszczania się w rozległym, otwartym środowisku trójwymiarowym w celu odnalezienia wskazanego obiektu. Wprowadza fotorealistyczny benchmark, w którym symulowany bezzałogowy statek powietrzny przemieszcza się w otwartym terenie, pozyskując kolejne obserwacje. Został on stworzony do oceny zdolności współczesnych modeli wizyjno-językowych (ang. vision-language models) do prowadzenia takich poszukiwań. Badane modele potrafią wykonywać krótkie zadania nawigacyjne, lecz mają trudności z systematycznym przeszukiwaniem otoczenia, gdy poszukiwany obiekt nie jest początkowo widoczny. Zastosowany fine-tuning poprawia skuteczność krótkich działań nawigacyjnych, także w środowisku innym niż treningowe, ale nie rozwiązuje problemu systematycznych poszukiwań.

Łącznie pierwsze trzy prace wiążą ze sobą wewnętrzny sygnał wskazujący, co warto zaobserwować, architekturę zdolną przetwarzać obserwacje o różnym położeniu i skali oraz strategię ich wyboru wyuczoną przez agenta. W badanych warunkach pozwalają one prowadzić efektywną eksplorację lokalną bez ograniczenia do stałej siatki. Czwarta praca pokazuje, jak dużym dodatkowym wyzwaniem jest łączenie obserwacji i ruchów w konsekwentnie realizowaną strategię poszukiwań w rozległym, otwartym terenie. Wyniki te wyznaczają kierunki dalszych badań: uczenie długofalowych strategii poszukiwań, przenoszenie metod do rzeczywistych środowisk oraz wyjaśnianie, jakie informacje wizualne stanowią podstawę decyzji agenta.

Wszystkie cztery publikacje ukazały się w materiałach „main track" wiodących konferencji. Trzy z tych konferencji należą do kategorii A* w rankingu CORE (200 punktów MNiSW), a jedna do kategorii A (140 punktów MNiSW). Jestem pierwszym autorem wszystkich czterech publikacji. Badania były częściowo finansowane z grantu PRELUDIUM Narodowego Centrum Nauki *Where to look next – aktywna eksploracja sterowana wewnętrzną niepewnością modelu*, którym kierowałem. Za działalność naukową otrzymałem również stypendium START Fundacji na rzecz Nauki Polskiej. Ponadto w trakcie studiów doktoranckich odbyłem staż naukowy w zespole Computer Vision w Max Planck Institute for Informatics pod opieką prof. Bernta Schiele. Poza głównym tematem rozprawy uczestniczyłem w badaniach nad zastosowaniami sztucznej inteligencji w diagnostyce medycznej, które doprowadziły do europejskiego zgłoszenia patentowego, a także w pracach nad wyjaśnialną sztuczną inteligencją oraz oceną zdolności modeli typu vision foundation model do rozumowania przestrzennego. Recenzowałem również prace dla NeurIPS, IJCV, TPAMI, RA-L, WACV i KDD oraz współorganizowałem trzy edycje Machine Learning Summer School i ELLIS Doctoral Symposium.

Słowa kluczowe: aktywna eksploracja wizualna, widzenie ucieleśnione, aktywna percepcja, transformer wizyjny, uczenie ze wzmocnieniem, modele wizyjno-językowe, nawigacja ukierunkowana na obiekt.

List of Publications

[I] Adam Pardyl, Grzegorz Rypeść, Grzegorz Kurzejamski, Bartosz Zieliński, and Tomasz Trzciński. Active Visual Exploration Based on Attention-Map Entropy. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1303–1311, 2023. https://doi.org/10.24963/ijcai.2023/145.
CORE RANK A*, 200 MNiSW points.

[II] Adam Pardyl, Grzegorz Kurzejamski, Jan Olszewski, Tomasz Trzciński, and Bartosz Zieliński. Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8536–8545, 2025. https://doi.org/10.1109/WACV61041.2025.00827.
CORE RANK A, 140 MNiSW points.

[III] Adam Pardyl, Michał Wronka, Maciej Wołczyk, Kamil Adamczewski, Tomasz Trzciński, and Bartosz Zieliński. AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 112–129, 2024. https://doi.org/10.1007/978-3-031-72664-4_7.
CORE RANK A*, 200 MNiSW points.

[IV] Adam Pardyl, Dominik Matuszek, Mateusz Przebieracz, Marek Cygan, Bartosz Zieliński, and Maciej Wołczyk. FlySearch: Exploring how vision-language models explore. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 185979–186011, 2025. https://doi.org/10.52202/085713-5594.
CORE RANK A*, 200 MNiSW points.

Other works published during the PhD studies, but not included in this dissertation, are briefly presented in Chapter 5.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.1.1 The enduring dream of intelligent machines

The dream of creating artificial beings that perceive and act like us is far older than the invention of the semiconductor. From the ancient Greek myth of the bronze automaton Talos [78], through the mechanical contraptions of the Enlightenment [108, 125], to the very word *robot*, coined in Karel Čapek's 1920 play R.U.R. [20], and the *Maschinenmensch* (machine-human) of Fritz Lang's 1927 movie Metropolis [59], the idea of building mechanical creatures has been a recurring theme in our history. Following the development of modern electronics and digital computers, this dream seemed within reach. Throughout the mid-20th century, Isaac Asimov's tales of positronic-brain-powered robots [7] presented a vision of highly intelligent anthropomorphic machines that reason based on logical rules and hardcoded ethics to guide their actions. Stanisław Lem's stories [64] went a step further, giving them personalities, flaws, and desires, showcasing the complexity of artificial civilisations and their potential ethical dilemmas. Movies such as Forbidden Planet (1956) [132] and Star Wars (1977) [76] presented robots as beings with an almost human psychology and thinking capabilities, but clumsier and more limited in their actions, bringing the idea to the mainstream. By the mid-20th century, the dream of intelligent machines was also becoming a scientific research programme. At that time Alan Turing reframed the question of whether machines can think in operational terms [122]. A few years later the founders of artificial intelligence proceeded from the optimistic premise that the essential features of intelligence could be described precisely enough to be simulated by a computer algorithm [79]. Figure 1.1 illustrates this enduring cultural aspiration to create machines with human abilities.

1.1.2 Moravec's paradox

Reality proved to be quite different and more humbling. Machines surpassed even the best human experts at tasks such as chess [18], once regarded as a pinnacle of human intellect. Automated theorem proving and satisfiability solving also achieved substantial successes in formal reasoning [26, 27]. At the same time, everyday competences such as finding a path in a cluttered room, locating and moving objects from one place to another, or even just looking for a familiar face in a crowd, turned out to be surprisingly hard to reproduce even at the animal level. This inversion, known as Moravec's paradox [83], has a compelling

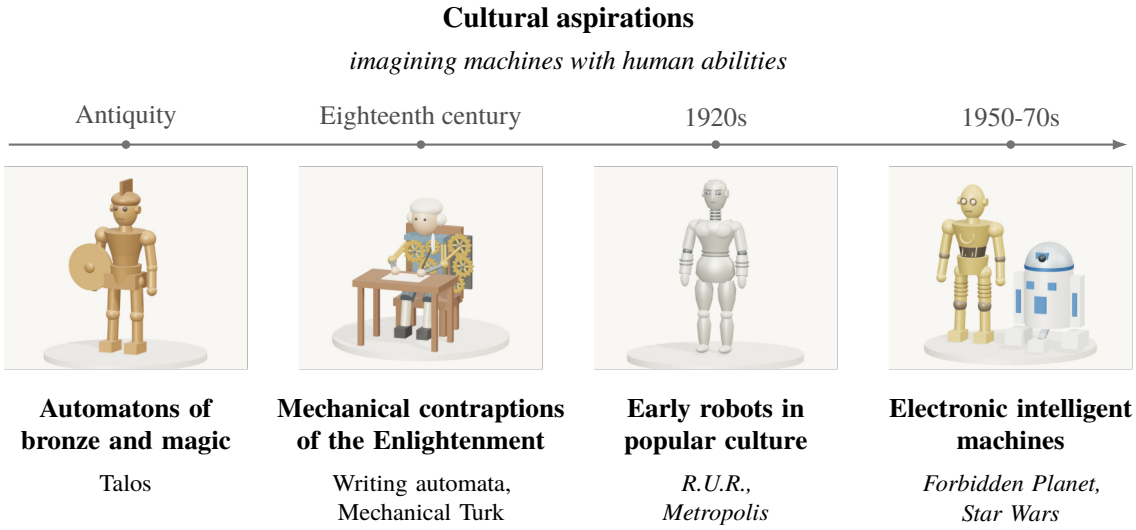


Figure 1.1: **Cultural aspirations for intelligent machines.** The idea of creating intelligent artificial beings dates back to antiquity. This figure traces the evolution of cultural depictions of intelligent machines through selected representative examples [59, 76, 78, 125]. The illustrations are schematic.

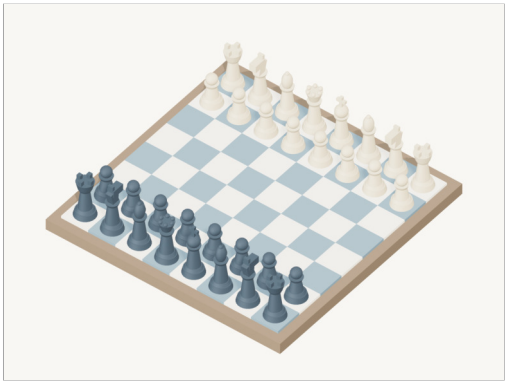
explanation in evolutionary terms. Those abilities are a product of hundreds of millions of years of natural selection. The sudden emergence of eyes is even credited with starting the Cambrian expansion of animal life [58, 98], and a large fraction of the primate cerebral cortex is dedicated to processing what we see [38]. We are, as a consequence, expert visual explorers by nature, able to efficiently use our limited perceptual and computational resources in a complex and dynamic world. What our brains do almost subconsciously – allocating gaze and gathering just enough visual evidence to act – remains as of today one of the hardest open problems in artificial intelligence [31]. Figure 1.2 illustrates the perceptual decisions hidden within an apparently simple everyday task.

1.1.3 The narrow window of perception

Every agent that acts in the physical world, whether it is a human, an animal, or a machine, does so through a narrow window. An eagle circling above a meadow searching for prey, a person looking for a familiar face in a crowded corridor, or a robot inspecting a factory floor for defects never perceives its surroundings all at once and in perfect detail. Instead, it is limited by its perceptual capabilities and environmental constraints. Furthermore, the environment is not static, but changes constantly, making a complete, up-to-date map difficult to maintain. Therefore, the agent must be able to reason based on partial information, while actively pointing its sensors toward the most informative regions to gather more data as needed. The perception process is not a passive act of reception, but rather an active, goal-driven process of exploration [4, 11, 13], illustrated in Figure 1.3. Human studies have shown that this exploration process happens on both a local and a global scale. On the local scale, we are experts at scanning our surroundings with our eyes, moving our gaze in a task-driven manner [46, 139]. On the global scale, we excel at exploring our environment by moving our body to different locations, navigating through complex scenes and avoiding obstacles without much effort [37]. The main motivation of this thesis is to enable

Formal intellectual task

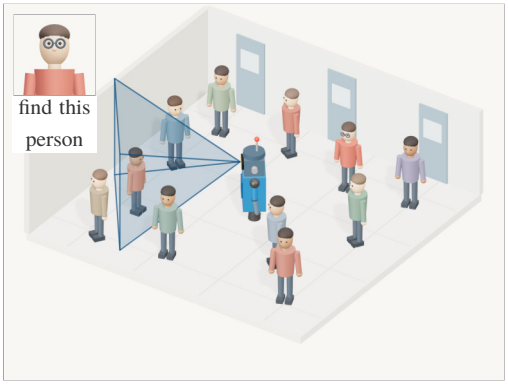
choose a move in chess



Explicit rules and objective
Relevant game state available
Evaluate possible moves
within a defined problem

Everyday visual task

find a familiar person



Occlusion and incomplete views
Limited time to gather evidence
Choose where to look, recognise the person,
and decide when enough has been seen

Figure 1.2: **The hidden difficulty of everyday perception.** Moravec's paradox contrasts artificial intelligence and robotics achievements in formal intellectual tasks with the difficulty of reproducing everyday perceptual competences [83]. In chess, the rules, objective, and relevant game state are available to the player. Finding a familiar person in a crowd requires acquiring missing visual evidence under occlusion and time constraints, recognising the person, and deciding when enough has been seen.

machines to do the same: to *learn to actively explore* their environment efficiently, perceiving only what they need in order to fulfil their goals.

The gap between this goal and the current state of machine learning and computer vision remains far from closed. Modern models achieve remarkable accuracy in passive perception tasks [30, 55, 87], yet the very assumption of complete access to visual information [41] breaks down the moment a model is placed inside an embodied agent with a limited field of view, a finite time budget, and a world that constantly changes [104, 131]. Understanding how an agent should efficiently acquire visual information, rather than just how it should process information it has already captured, remains an open problem [12, 32, 73, 104]. The remainder of this chapter traces how the field arrived at this point, explains why general-purpose exploration is still unsolved, and introduces the research direction pursued across the four publications that constitute this thesis.

1.2 A brief outline of the field

1.2.1 From hand-crafted features to foundation models

Computer vision is almost as old as computing itself, with the first works on machine perception dating back to the 1960s [109], epitomised by the so-called "blocks world" of simple polyhedral scenes. First attempts to apply machine vision to real-world robotics problems followed soon after, with the SRI Shakey project [86], developed in the late 1960s and early 1970s. Roughly a decade later, Moravec's stereo-vision-

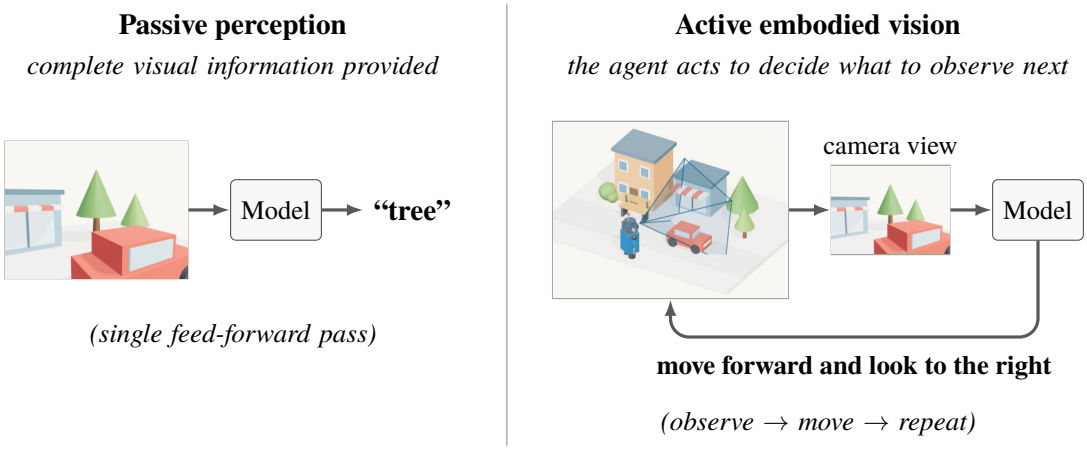


Figure 1.3: **Passive perception versus active embodied vision.** A passive perception system is handed a complete, static view of the scene to produce a prediction. An active embodied agent instead senses only a narrow view of the surrounding scene and closes the loop by actively choosing where to move for the next observation.

guided Stanford Cart [84] continued this path. These early systems were built on hand-crafted geometry and operated in carefully prepared evaluation environments. Their inability to operate reliably outside such controlled settings highlighted the difficulty of real-world visual perception.

For the next two decades, progress was incremental at best and heavily reliant on hand-crafted feature representations. The turning point came with the resurgence of neural networks, driven by the availability of large annotated datasets [29] and the development of more powerful hardware. Convolutional networks, pioneered for digit recognition [60], were scaled up and decisively outperformed hand-crafted pipelines [55]. Deeper and more trainable architectures such as residual networks [47] followed, and learned representations rapidly became the default across tasks such as classification, detection, and segmentation [61]. Computer vision had become, first and foremost, a problem of representation learning.

The next wave of progress came from the field of natural language processing, with the invention of attention-based transformer models [124]. The transformer architecture proved equally powerful for visual data, quickly surpassing convolutional networks in accuracy [30]. Moreover, transformers were able to learn rich feature representations with self-supervised and weakly supervised learning [21, 48, 101], enabling training of models on vast amounts of unlabelled or loosely labelled data from the web. Those so-called vision foundation models in turn not only proved to be highly accurate backbones for standard vision tasks, but also enabled the development of vision-language models [40, 87], which generalised and scaled up the ability of machines to describe and reason about images in language on an unprecedented scale.

This progress from the 1960s blocks world to today’s foundation models is, by any measure, a triumph. Yet nearly all of these advances share a quiet assumption: that the model is handed all needed visual information at once as a complete, static, sufficiently high-resolution image. Figure 1.4 summarises these advances in visual representation learning and parallel developments in embodied vision.

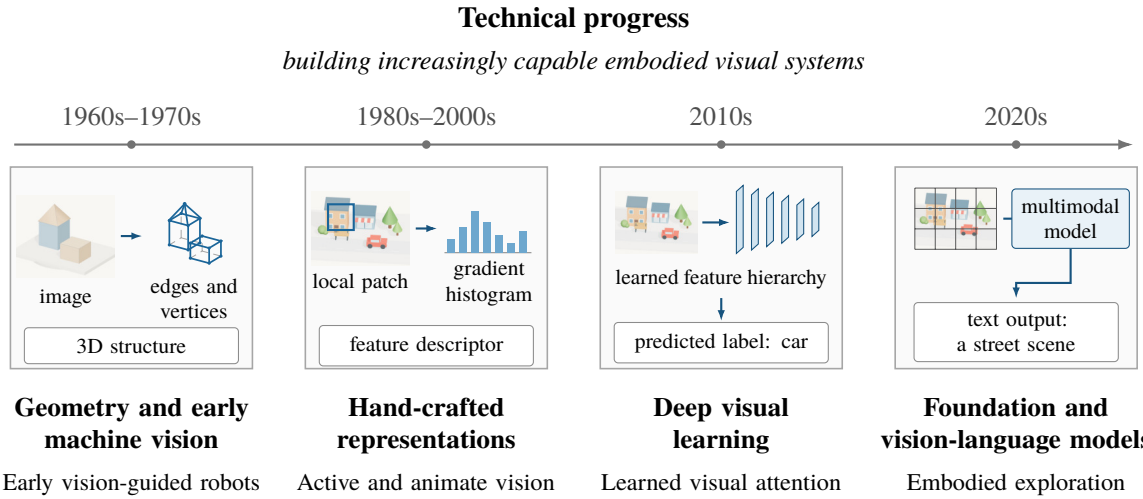


Figure 1.4: **Technical progress in visual learning.** Selected milestones in visual representation learning [30, 61, 101, 109] and parallel developments in embodied vision [4, 13, 81, 86, 104], as described in Section 1.2. The illustrations are schematic.

1.2.2 The passive-perception assumption

Standard computer vision solutions assume complete and unobstructed access to input data [41]. This is an entirely reasonable simplification for most studied tasks, such as image classification or segmentation, and even the most recent vision-language benchmarks [23, 141] follow this pattern. However, this assumption falls apart when the model is placed inside an *embodied* agent, such as a robot or an unmanned aerial vehicle (UAV), operating in the real world. Such an agent perceives the world through sensors with a restricted field of view and resolution, must move physically to acquire new observations, taking obstacles into account, and operates under tight constraints on time, energy, and computational power [131]. Capturing and processing every part of a large environment at maximum resolution is not only wasteful but often infeasible, especially in open-world settings [81, 104, 110].

Under these conditions, the question is no longer *what we currently see*, but rather *where we should look* in the first place. An agent that can choose its observations wisely can achieve strong performance at a fraction of the sensing and computational cost of one that processes everything indiscriminately. For active perception to work we need to move this decision of *where to look next* inside the model itself.

1.2.3 Active embodied vision

The idea that perception systems should be active has been around since the early days of embodied computer vision. The active perception [11], active vision [4], and animate vision [13] proponents stipulated decades ago that a perceiver which controls its own sensing can turn otherwise ill-posed problems into tractable ones. This perspective has been revisited in the modern learning era [12]. The inspiration is deeply biological. Human visual acuity is concentrated at the centre of gaze. Therefore, we explore scenes through rapid saccadic eye movements that direct this fovea to informative regions [46, 139].

Machine learning approaches to active vision followed the broader deep learning progress. Recurrent models of visual attention [81] learned to select a sequence of glimpses for recognition, drawing on the

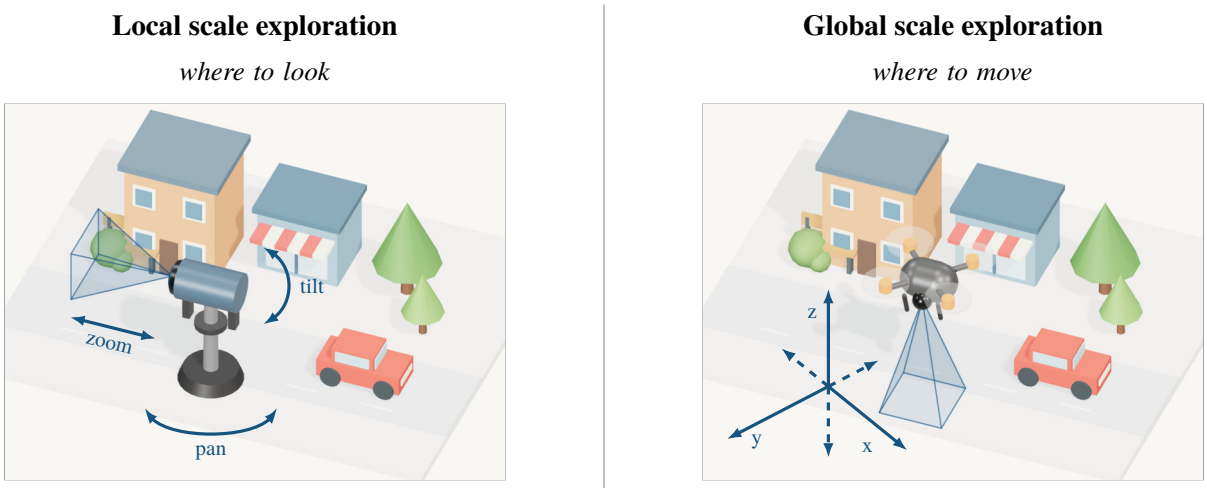


Figure 1.5: **The two scales of active exploration.** At the local scale, a stationary agent (e.g. a pan-tilt-zoom camera, or a robot's head) remains at a fixed position and redirects its gaze towards different regions of the surrounding scene. At the global scale, a mobile agent (e.g. a robot or a UAV) relocates itself through three-dimensional space.

reinforcement learning methods [120] that had begun to master control problems from raw pixels [82]. In parallel, the robotics and embodied-AI communities studied exploration in the context of simultaneous localisation and mapping (SLAM) [17], where an agent must build an explicit map of the environment. More recently, the task of embodied visual exploration [104] studied how an agent should build an internal model of an unfamiliar space to support downstream tasks such as reconstruction, localisation, and navigation. Another recent line of work on active visual exploration specifically studies how a model equipped with a limited field of view should select successive glimpses to reconstruct, classify, or segment a scene [115]. These threads share a common conviction: that intelligent sensing is a decision problem.

1.2.4 Why general exploration is still unsolved

Despite this long history of research and recent rapid progress, the problem of exploration in the general case remains far from solved. The difficulty is easy to underestimate because existing methods achieve strong performance within simplified settings. For example, some active-vision methods operate in small search spaces [115], where even random glimpse selection was found to perform on par with attention-based selection [52]. Moreover, the exploration is often limited to a grid game, where an agent uncovers a fixed area at each move [106], a simplification that mirrors the grid input format of vision transformers rather than the continuous movements of real-world agents. Embodied navigation is most often studied in constrained indoor simulators [54, 111, 136], where the agent can map the entire environment explicitly, without having to deal with the complexity and vast size of real-world environments. This allows relatively simple object-goal navigation methods [14, 44] to produce strong results on those benchmarks, but it does not address the problem of general exploration. A different body of work pushes this global-scale exploration outdoors, using simulators such as AirSim [116], vision-and-language navigation tasks such as AerialVLN [71] and CityNav [62], or, more rarely, open-ended object-goal search directly in such

settings [138]. Notably, most of this active-vision progress, indoors and outdoors alike, still operates at what we earlier called the local scale, or follows a fairly constrained navigation task: an agent at each step decides where to look within a scene it can fully scan, or move towards an object it can already see. Whether the same efficient, learned decision-making transfers to unconstrained, zero-shot, global-scale search remains largely an open question [12, 104]. Finally, it is not clear whether methods designed for simplified simulation environments can be transferred to the real physical world, bridging the sim-to-real gap [3].

1.3 Learning to explore efficiently

This dissertation investigates how active visual exploration (AVE) can become more *efficient* (understanding an environment from as few observations as possible), more *flexible* (exploiting the continuous capabilities of real sensors rather than artificial grids), and more *scalable* (toward the demands of embodied agents operating in the open world). It considers both the local scale of choosing where to look and the global scale of choosing where to move, contrasted in Figure 1.5. The four publications that make up the thesis form a progression along this path, each investigating one of the following research hypotheses.

1.3.1 Research hypotheses

H1. The uncertainty already encoded in a visual model's attention maps can be used to decide where to look next, without training a separate component for observation selection.

H2. A vision transformer can be made resilient to input patches sampled at varying positions, scales, or with incomplete coverage, allowing it to process observations that do not follow a rigid grid.

H3. An agent can learn to select both the position and scale of each observation in a continuous space, discovering an efficient coarse-to-fine exploration strategy that reduces the number of pixels it needs to observe.

H4. Current vision-language models can perform short-horizon visual navigation, but cannot reliably form and execute consistent multi-step search strategies in large, open, three-dimensional environments.

The first contribution, "Active Visual Exploration Based on Attention-Map Entropy" [I], investigates Hypothesis H1 by asking how an agent should decide where to look next without the machinery of a separate decision network. It shows that the internal uncertainty of a transformer, read directly from the entropy of its attention maps, is itself an effective signal for selecting the most informative next glimpse, simplifying training while improving reconstruction, classification, and segmentation from partial observations (see Figure 1.6a). This work operates, however, within the familiar setting of glimpses drawn from a fixed grid.

The second contribution, "Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers" [II], investigates Hypothesis H2 and addresses the constraint of fixed grid sampling. It formalises the notion of input *elasticity*, the resilience of a model to patches of varying scale, position, and coverage, introduces a protocol to measure it, and proposes architectural and training modifications that let a vision

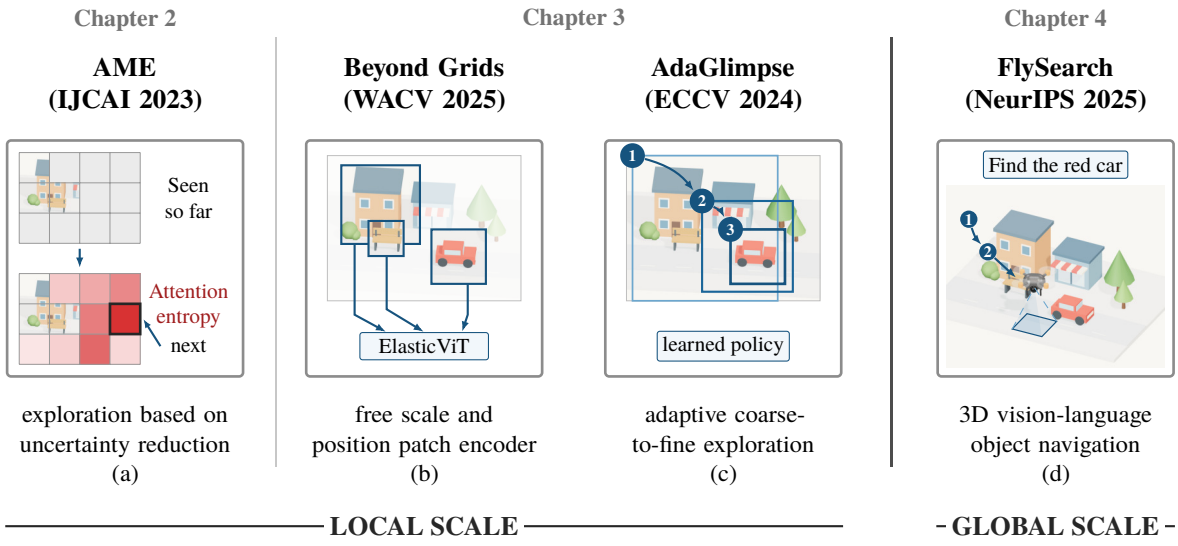


Figure 1.6: **The four contributions of this thesis as a progression.** The first three papers generalise local-scale exploration, deciding where to look within a scene the agent can already sense, from a fixed grid (AME), to elastic patch geometry (Beyond Grids), to arbitrary continuous zoom and pan (AdaGlimpse). The fourth, FlySearch, turns to the complementary global-scale problem of deciding where to move through an open, three-dimensional world.

transformer ingest patches sampled freely from an image. This turns the rigid grid into a continuous sampling space (see Figure 1.6b).

The third contribution, "AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale" [III], investigates Hypothesis H3 and brings observation selection closer to the capabilities of real sensors. Building on an input-elastic transformer, it casts exploration as a Markov decision process and uses the Soft Actor-Critic reinforcement learning algorithm to select glimpses of arbitrary continuous position and scale, the degrees of freedom offered by motorised pan-tilt-zoom cameras (see Figure 1.6c). The resulting agent rapidly establishes a coarse awareness of a scene before zooming in on detail, matching or surpassing prior methods while observing far fewer pixels.

The first three papers progressively generalise *local-scale* exploration: deciding where to look within a scene the agent can already sense, moving from a fixed grid (AME), to elastic patch geometry (Beyond Grids), to arbitrary continuous zoom and pan (AdaGlimpse). The fourth contribution, "FlySearch: Exploring how vision-language models explore" [IV], investigates Hypothesis H4 and turns to the complementary *global-scale* problem: deciding where to move next in a large, open, three-dimensional world in order to find something. The environment is no longer a bounded scene from which glimpses can be sampled, but a vast outdoor search area that the agent, a simulated UAV, must physically traverse to acquire each new observation. FlySearch introduces a photorealistic benchmark for goal-driven object search in such environments and uses it to ask whether today's vision-language models can execute this kind of exploration (see Figure 1.6d). It finds that state-of-the-art models fail to reliably solve even simple exploration tasks, with the gap to human performance widening sharply as tasks demand longer, consistent search strategies, and this gap persists even after fine-tuning. In mapping out these failure modes, FlySearch both charts the frontier of the problem and motivates further research in this field.

Taken together, these works trace an arc from deciding where to look within a single reachable scene, through freeing the model from rigid sampling and matching real sensor geometry, to deciding where to move through an open, three-dimensional world. They constitute concrete steps in pursuit of the goal that gives this thesis its title: learning to look and search, towards efficient visual exploration for embodied agents. Figure 1.6 summarises this progression.

Parts of this research were carried out within the National Science Centre Preludium grant *Where to look next – guiding active visual exploration with internal model uncertainty* (2024–2025), of which I was the principal investigator. My work was also recognised with the START scholarship of the Foundation for Polish Science (2026).

1.4 Thesis outline

The remainder of this thesis follows the progression outlined above. Chapter 2 studies exploration at the local scale and presents the first publication, "Active Visual Exploration Based on Attention-Map Entropy", which recasts glimpse selection as uncertainty reduction driven by the internal attention entropy of the task model itself. Chapter 3 overcomes the fixed-grid limitation underlying that work and covers two publications: "Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers", which formalises input elasticity and introduces the ElasticViT architecture, and "AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale", which trains a reinforcement learning agent to select glimpses of continuous position and scale on top of that backbone. Chapter 4 moves to the global scale and presents "FlySearch: Exploring how vision-language models explore", a benchmark for goal-driven object search in large outdoor environments. Each of these chapters opens with its research scope and the preliminaries needed to keep the thesis self-contained, then summarises the corresponding publication and its contributions. The full texts of the publications are provided in the appendix. Chapter 5 gives an overview of other research work and achievements from the author's doctoral studies, and Chapter 6 concludes the thesis with a synthesis of the results and a discussion of future research directions.

Chapter 2

Exploration as uncertainty reduction

2.1 Research scope

As established in the introduction, an embodied agent operating with a restricted field of view cannot rely on the passive-perception assumption that underlies most of modern computer vision [41]. It must instead decide, step by step, where to direct its sensor to build a sufficient understanding of a scene from as few observations as possible. This chapter studies this problem at what we earlier called the *local scale*: the agent is stationary and positioned in front of a single bounded scene (an image, a panorama, or a room), and must choose a sequence of *glimpses* – small regions of that scene it is allowed to observe one at a time, so as to reconstruct, classify, or segment the scene as accurately as possible within a limited budget of observations.

Prior work at this local scale has explored a range of strategies for deciding *where to look next*. Some methods frame the choice explicitly in terms of uncertainty. One approach hallucinates plausible completions of the unseen part of a scene with a variational autoencoder and derives an uncertainty score from the resulting hypotheses [105], while others pair a reinforcement-learning policy with a predictive-uncertainty signal to choose the parameters of the next observation [51, 103], or apply a related attention-driven reinforcement-learning policy to patch selection in video streams [22]. Another line of work, which is most directly relevant to the publication summarised in this chapter, instead learns the next-glimpse decision as a dedicated, supervised component trained alongside the target task. One such method uses a convolutional architecture with a probability-map output head for reconstructing 360° scenes from a sequence of glimpses [113]. This approach was later extended to segmentation [114] and to an attention-pooling variant trained through five parallel streams [115]. This methodology was subsequently transferred to a transformer-based masked autoencoder with a separate glimpse-decision head attached on top of it [52]. Despite their differences, all of these approaches converge on a common architectural pattern. A neural network performs the target task (e.g., reconstruction) from the glimpses observed so far, while a second, separately trained component (typically a dedicated decision head with an auxiliary loss, or a reinforcement-learning policy) decides where the next glimpse should be taken. This pattern introduces an additional training objective and can increase the complexity, memory footprint, and fragility of the training procedure.

The central research question of this chapter is whether such a dual-objective design is in fact necessary, or whether a network trained for the target task alone already contains, inside its own internal

representations, sufficient information about where it is most uncertain, and therefore where it should look next. If so, active exploration could be reduced to a form of *uncertainty reduction*: at every step, the agent samples the observation about which the model currently knows the least, without ever training a separate mechanism to make that choice. This idea is explored through the publication summarised in this chapter, "Active Visual Exploration Based on Attention-Map Entropy" [I], which identifies such a signal directly in the attention maps of a transformer-based model and shows that it can replace a dedicated decision module without loss, and, in fact, with a substantial gain in performance.

2.2 Preliminaries

Before describing the publication summarised in this chapter, to make this thesis self-contained, we introduce four background concepts required to understand it: the transformer-based masked autoencoder architecture on which it is built (Section 2.2.1), the notion of entropy as a measure of predictive uncertainty (Section 2.2.2), the glimpse-based exploration protocol used throughout the chapter (Section 2.2.3), and the retina-like sensor model evaluated as one of its variants (Section 2.2.4).

2.2.1 Vision transformers and masked autoencoders

The Vision Transformer (ViT) [30] adapts the transformer architecture [124] originally developed for language to images. An input image is divided into a regular grid of non-overlapping square patches, each of which is linearly projected into an embedding vector and augmented with a positional embedding encoding its location in the grid. The resulting sequence of tokens is processed by a stack of self-attention blocks, each of which lets every token gather information from every other token in the sequence, weighted by a learned compatibility score. Because a ViT operates on a variable-length sequence of tokens rather than on a fixed-size pixel grid, it is naturally suited to inputs where only a subset of patches is available at any given time.

Masked Autoencoders (MAE) [48] exploit this property for self-supervised representation learning. During training, a large fraction of the input patches (e.g. 75%) is removed. An encoder processes only the small set of remaining, visible patches, and a lightweight decoder is tasked with reconstructing the missing patches from the encoder's output, together with positional information about where the missing patches were located. Because the encoder never has to process masked tokens, this design is both computationally efficient and, crucially for active exploration, directly compatible with an agent that only observes a partial view of a scene.

2.2.2 Entropy as a measure of uncertainty

Given a discrete probability distribution $p = (p_1, \ldots, p_n)$, its Shannon entropy is defined as

$$H(p) = -\sum_{i=1}^{n} p_i \log p_i. \tag{2.1}$$

Entropy is maximised when p is uniform, i.e. when all outcomes are equally likely, and minimised, reaching zero, when p places all of its probability mass on a single outcome. In machine learning, the entropy of a model's output distribution is a standard proxy for the model's uncertainty: a confident

prediction concentrates its probability mass on one or a few outcomes and has low entropy, while an uncertain prediction spreads its probability mass broadly and has high entropy.

2.2.3 The glimpse-based exploration protocol

Throughout this chapter, an image is partitioned into a fixed grid of N candidate *glimpse positions*. A glimpse is the unit an agent acquires in a single step. Depending on the sensor regime, it is either a single patch or a small, fixed group of neighbouring patches. An agent is given a fixed budget of $T \ll N$ steps. At step t, it has already observed t glimpses and must select the position of the $(t+1)$-th glimpse from among the candidate positions not selected previously. Once the glimpse is revealed, the model performs the target task (reconstruction, classification, or segmentation) again, conditioned on all $t+1$ known glimpses. This process repeats until the budget T is exhausted, at which point the final prediction is produced.

2.2.4 Retina-like foveated sensing

The human retina offers a useful model for efficient partial observation: it delivers high spatial acuity only within a small central region called the *fovea*, while receptor density (and thus spatial resolution) decreases steeply towards the periphery [25, 110]. This non-uniform sampling is highly efficient – a wide field of view is maintained with relatively few photoreceptors, while fine detail is reserved for the region currently fixated. The publication summarised in this chapter evaluates a *retina-like* glimpse variant that mirrors this structure: each rectangular glimpse is rendered at full resolution at its centre but progressively degraded towards its edges (simulated through resampling filters), in contrast to a uniform-resolution glimpse of the same spatial extent.

2.3 Active Visual Exploration Based on Attention-Map Entropy

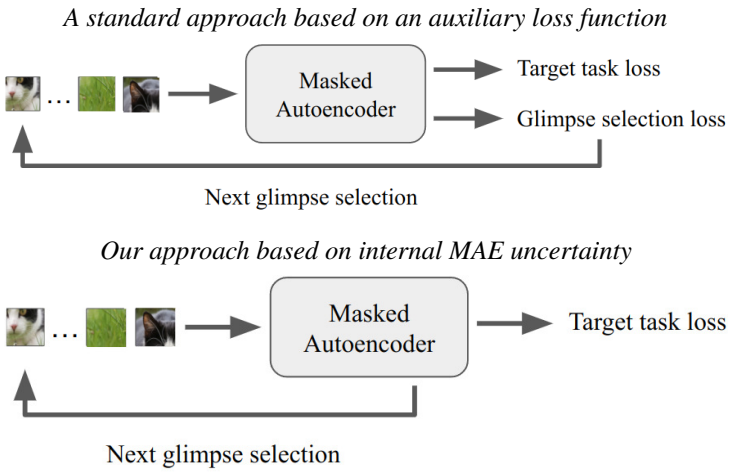


Figure 2.1: **Attention-Map Entropy (AME).** The proposed approach chooses the most informative observations (patches) by reusing the internal uncertainty already coded in the attention maps of the target-task model. In contrast to existing methods, it does not require any auxiliary loss function dedicated to active exploration, so that training can concentrate entirely on the target-task loss. Figure reproduced from [I].

2.3.1 Motivation

As established in Section 2.1, prior work on glimpse-based active exploration relies on a dedicated network component, trained separately from and in parallel to the target task, to decide where to look next. This *dual-objective* design has a tangible cost – it increases the number of network components, raises the memory footprint, and complicates training, since two training objectives (one for the perceptual task, one for exploration) may pull in opposite directions.

This publication investigates Hypothesis H1: the uncertainty already encoded in a visual model's attention maps can be used to decide where to look next, without training a separate component for observation selection. A transformer-based masked autoencoder processing a partial view of a scene already produces a rich description of what it is uncertain about as the entropy of its self-attention maps. The question is if such a signal alone is sufficient to drive effective glimpse selection, and whether eliminating the second objective actually improves rather than harms performance (see Figure 2.1).

2.3.2 Attention-Map Entropy

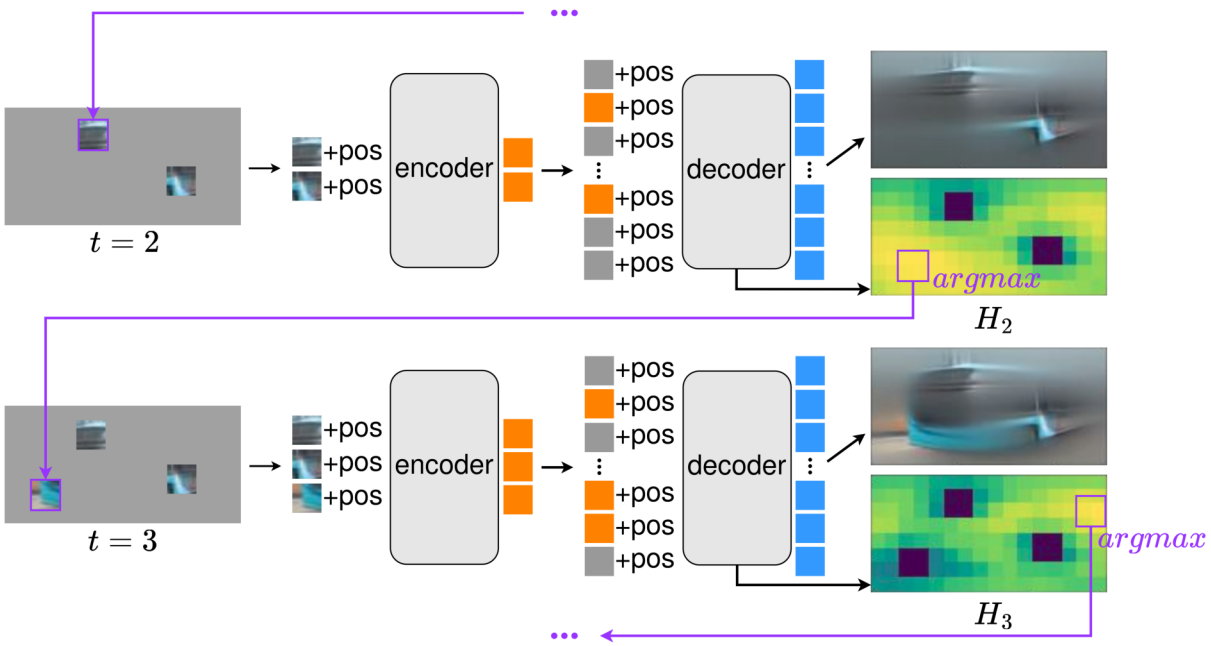


Figure 2.2: **Architecture for reconstruction.** The agent has observed two patches of the image, which are processed by the encoder to produce their feature representations (orange rectangles). These outputs are combined with the masked patches (grey rectangles) and passed through the decoder, which reconstructs the missing image patches. In addition, the method computes the entropy map for one of the decoder's multi-head self-attention layers and uses it to select the location of the third glimpse (the one with the highest entropy). The process repeats until the assumed number of glimpses is reached. Reproduced from [I].

Architecture. The proposed method, named *Attention-Map Entropy (AME)*, builds upon a masked autoencoder (Section 2.2.1) with a ViT-L encoder (24 transformer blocks and embedding size 1024) and a lightweight 8-block decoder (embedding size 512). At every step t, the encoder processes the t glimpses

observed so far, and the decoder combines their latent representations with mask tokens standing in for the unseen patches, producing an output for the target task. Figure 2.2 illustrates this process for the reconstruction task. For other tasks, the architecture is similar, with the exception of the output head.

Attention-derived uncertainty. The key idea is to reuse the decoder's own self-attention maps to decide on the next action, instead of training a separate module for this purpose. Each row of a self-attention map is produced by a softmax and therefore defines a probability distribution describing how strongly the corresponding token attends to every other token when computing its representation. High entropy in a row indicates that the model cannot commit to any particular token as informative, which in turn signals that the underlying image region is poorly understood and would benefit from being observed. Because these distributions arise as a by-product of the target task's own forward pass, their entropy provides an uncertainty signal at no extra computational cost, with no additional parameters or training objective required.

For the single-patch glimpse regime, given the patches $p_1, \ldots, p_t$ observed so far, the decoder operates on the full set of M tokens, one for every patch position of the image grid: the t observed patches plus mask tokens standing in for each position not yet observed. Each of its self-attention layers therefore computes an attention matrix

$$A = \text{softmax}\left(QK^T/\sqrt{d_k}\right) \in \mathbb{R}^{M\times M}, \tag{2.2}$$

where Q and K pack together the query and key projections of all M token embeddings produced by that layer, and d_k is the dimension of the keys and queries. Row i of A is the probability distribution used to compute a weighted average of all patch representations when producing the updated representation of patch i. A peaked row means the model relies on a few already well-understood patches, while a flat row means it must combine evidence from many patches because none of them alone provides a confident answer. Figure 2.3 illustrates this on a toy 2×2 example.

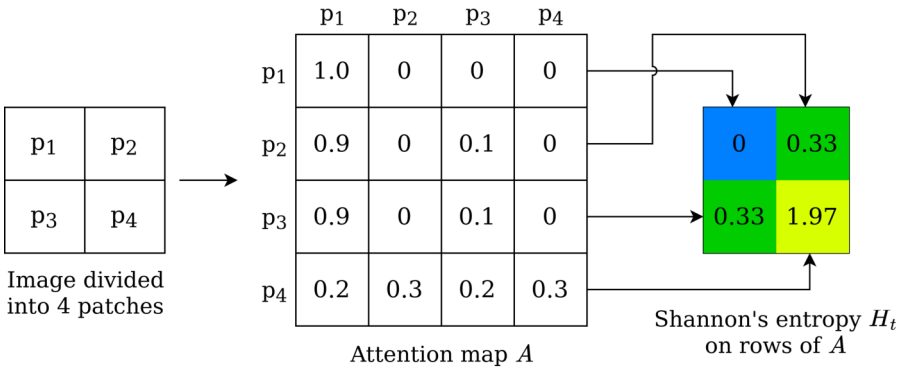


Figure 2.3: **Entropy map.** Consider an image divided into four patches (2×2), shown on the left. Its attention map is a 4×4 matrix, each row of which holds the attention weights used to compute the output of the corresponding patch in the next transformer layer. Computing the Shannon entropy (Equation 2.1) of each row yields a 2×2 entropy map; the patch with the highest entropy value is selected as the next glimpse. Figure reproduced from [I].

Glimpse selection and task outputs. Since transformers compute multiple attention maps in parallel (multi-head attention), the entropies contributed by all heads h are summed into a single entropy map at step t,

$$H_t[i] = \sum_h H(A_{h,t}[i]) \,, \tag{2.3}$$

where $A_{h,t}[i]$ is the i-th row of head h's attention matrix at step t, and $H(\cdot)$ is the Shannon entropy of Equation 2.1. Entries corresponding to already-known patches are set to zero so that they are never re-selected, and the next glimpse is placed following a greedy selection strategy, i.e. at the position of maximal remaining entropy,

$$i^* = \arg\max_i H_t[i]. \tag{2.4}$$

For multi-patch glimpses, the entropy map is averaged over the glimpse footprint before selecting a location. The very first glimpse, for which no patch has yet been observed, is chosen by running the same procedure with only mask tokens as input. For reconstruction and segmentation, the decoder's final linear layer directly outputs pixel values or per-class logits for every patch. For classification, an auxiliary head attached to the encoder's CLS token is trained jointly with the reconstruction objective. In every case, training uses only the losses of the perceptual tasks themselves, and no loss function or decision network dedicated to exploration is introduced.

2.3.3 Experimental results

The method is evaluated on three datasets: MS COCO [68], ADE20K [142], and SUN360 [137], against two families of baselines, each tested under its corresponding glimpse regime: (i) 8×48^2 *retinal* glimpses, matching the retina-like sensor simulation used by the convolutional methods AttSeg [114] and GlAtEx [115], and (ii) 37×16^2 regular, non-retinal glimpses of a single ViT patch each, matching the transformer-based SimGlim [52].

Reconstruction. As shown in Table 2.1, AME achieves the best reconstruction RMSE across every dataset and glimpse regime tested, improving over both convolutional and transformer-based baselines. Non-retinal glimpses further improve scores over retinal ones by more than 10%. The gain over SimGlim is particularly informative: since both methods share the same MAE backbone and differ only in the glimpse-selection rule, the improvement can be attributed to the entropy-based strategy itself rather than to the underlying architecture. Qualitative results on SUN360 in Figure 2.4 confirm that the reconstructions produced by AME are visibly sharper and more detailed than those of the convolutional baselines.

Figure 2.5 visualises the process step by step: as additional glimpses are selected, the entropy map concentrates on the regions of the scene that remain blurry or unexplained, and the overall reconstruction, not only in the immediate neighbourhood of the newest glimpse, becomes progressively sharper.

Classification. Classification results on SUN360 are summarised in Table 2.2. AME achieves 75.7% accuracy when training the full model end-to-end, compared to 56.4% for GlAtEx, and 70.1% when only the classification head is trained on top of frozen reconstruction features, still outperforming both convolutional baselines. The improvement in the head-only regime is particularly notable because the

Table 2.1: **Reconstruction results:** Comparison of our model (AME) in the reconstruction task against AttSeg [114], GlAtEx [115] and SimGlim [52] on SUN360, ADE20K and MS COCO datasets. The metric used is the root mean square error (RMSE; lower is better). For each experiment, we provide a training and evaluation regime defined by a number of glimpses of a specific resolution. Pixel % and area % denote respectively: the percentage of image pixels known to the model and the percentage of image area seen by the model. The two measures differ for retina-like glimpses, which have lower pixel densities in peripheral regions of the observation. Our method outperforms competing methods in all configurations.

Method	SUN360	ADE20K	MS COCO	Image res.	Glimpse regime	Pixel %	Area %
AttSeg	37.6	36.6	41.8	128×256	8×48^2 (retinal)	18.75	56.25
GlAtEx	33.8	41.9	40.3	128×256	8×48^2 (retinal)	18.75	56.25
AME (ours) (retinal)	**23.6**	**23.8**	**25.2**	128×256	8×48^2 (retinal)	18.75	56.25
AME (ours) (non-retinal)	37.9	40.7	43.2	128×256	8×16^2 (non ret.)	6.25	6.25
AME (ours) (non-retinal)	29.8	30.8	32.5	128×256	8×32^2 (non ret.)	25.00	25.00
AME (ours) (non-retinal)	**20.1**	**20.6**	**22.1**	128×256	8×48^2 (non ret.)	56.25	56.25
SimGlim (detached)	26.2	27.2	29.8	224×224	37×16^2 (non ret.)	18.75	18.75
SimGlim (end-to-end)	28.0	28.8	31.3	224×224	37×16^2 (non ret.)	18.75	18.75
AME (ours) (non-retinal)	**23.4**	**26.2**	**28.6**	224×224	37×16^2 (non ret.)	18.75	18.75

encoder was never exposed to any classification signal: the representations learned for reconstruction already provide a strong enough basis for scene-category decisions.

Semantic segmentation. Semantic segmentation on ADE20K follows the same pattern: AME raises pixel accuracy from 47.9% (AttSeg) and 52.4% (GlAtEx) to 70.3%. The attached publication text reports the full table.

Ablation studies. Ablation studies confirm that the entropy-based rule is a reasonable heuristic, which is measurably better than naive alternatives. AME outperforms both uniformly random glimpse placement and a fixed, non-overlapping checkerboard pattern on both reconstruction and classification. A further ablation over which decoder layer the attention map is drawn from shows a consistent trend that entropy computed from deeper decoder layers, closer to the final output, yields better glimpse-selection decisions than entropy computed from shallow ones. See the attached publication text for more details.

2.3.4 Contributions

The publication summarised in this chapter makes the following contributions to visual exploration:

- It introduces Attention-Map Entropy (AME), a glimpse-selection mechanism that reuses the entropy of a transformer decoder's own self-attention maps as a measure of the model's uncertainty about each unseen region. This removes the need for any auxiliary loss function or dedicated decision network for active exploration, in contrast to all prior approaches to this problem.

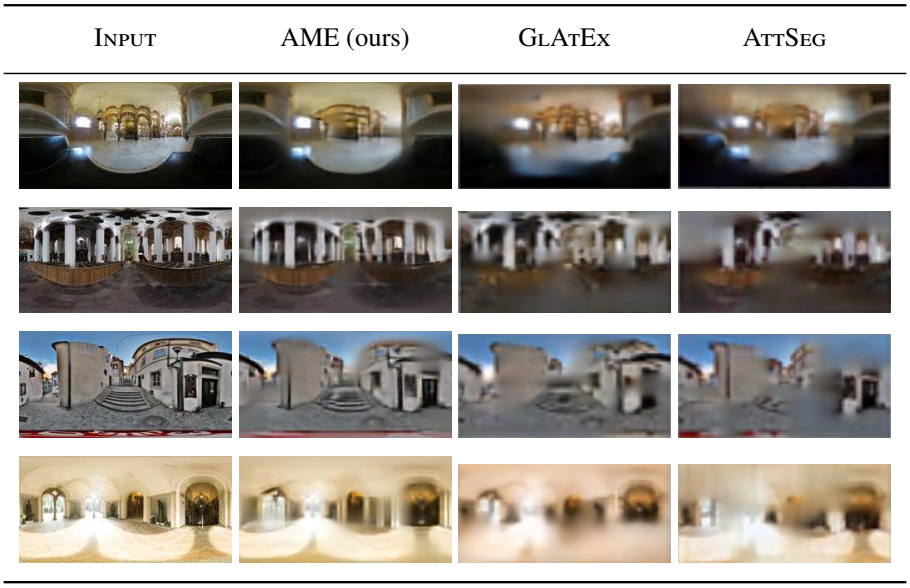


Figure 2.4: **Reconstruction quality for SUN360.** Qualitative reconstruction results comparing AME against AttSeg [114] and GlAtEx [115] on the SUN360 dataset using 8×48^2 retinal glimpses. Reconstructions produced by our method are visibly sharper and better preserve scene structure. Reproduced from [I].

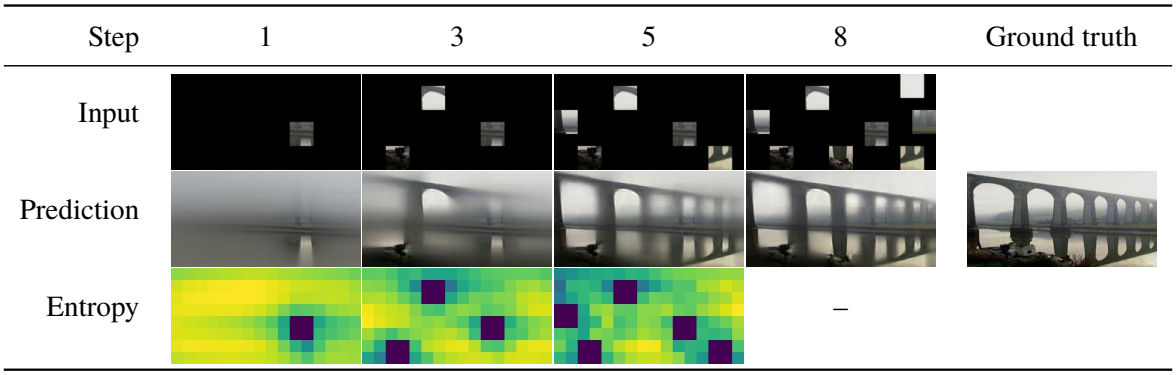


Figure 2.5: **AME-based reconstruction step by step.** Selected steps of an 8×32^2 glimpse-selection process on a sample 256×128 image from ADE20K, showing the model input (known glimpses), its prediction, and the decoder attention entropy driving the choice of the next glimpse (known areas are explicitly set to zero). AME explores the image where its own reconstruction is still uncertain. Adapted from [I].

- It shows that a single, task-only training objective is sufficient to obtain a strong exploration policy, simplifying the training procedure compared to dual-objective designs.
- It provides, to the best of the authors' knowledge, the first published empirical evaluation coupling transformer-based active visual exploration with retina-like glimpses, mimicking a foveated sensor.

The task results and sampling-rule ablations confirm Hypothesis H1 within the evaluated settings.

I was the first and primary author of this paper. I contributed to the conception of the idea and proposed using Shannon entropy as a measure of uncertainty, leading to the development of the AME method. I carried out the majority of the experimental design and implementation, conducted the majority of the

Table 2.2: **Classification results.** Multi-class classification accuracy on the SUN360 dataset using 8×48^2 retinal glimpses. *Train-all* trains the entire model jointly; *head-only* freezes the autoencoder weights and trains only the classification head on top. Our method (AME) outperforms competitive methods in both regimes. Reproduced from [I].

Method	Accuracy (%)
AttSeg	52.6
GlAtEx (full)	56.4
GlAtEx (no decoder)	67.2
AME (ours) (head-only)	70.1
AME (ours) (train-all)	**75.7**

experiments presented in the paper, and contributed significantly to writing the manuscript. I also presented the research at the International Joint Conference on Artificial Intelligence (IJCAI) 2023 in Macau. The code, data and further visualisations are available on my website at `https://io.pardyl.com/AME/`.

Chapter 3

Adaptive exploration

3.1 Research scope

The previous chapter describing AME confirmed that, in the settings evaluated, active visual exploration (AVE) can be guided by uncertainty reduction. At each step, the agent looks at the region about which the current model is most uncertain, using the entropy of its own internal representations. AME outperformed prior methods that required separately trained exploration modules. However, that work inherited a structural assumption shared by all prior methods in the field: glimpses are sampled from a *fixed-scale, grid-aligned* partition of the scene. Each observation covers a predetermined rectangular region at a predetermined resolution, and the agent's role reduces to choosing cells of a fixed grid (see Figure 1.6).

Nevertheless, the rigid grid is not unique to active exploration. It is inherited from the underlying vision transformer, and several lines of work have tried to relax it, each from a different angle. Hierarchical backbones such as Swin [75] and PVT [128] extract features at several fixed resolutions internally, merging tokens as depth increases, but the input tokenisation itself is still a single, uniform grid. FlexiViT [15] and NaViT [28] allow a whole image to be tokenised at a different, but still uniform, patch size or aspect ratio, and multi-grid training schedules [66, 135] vary this resolution across training iterations rather than within a single image. However, none of these approaches let individual patches of the same image differ in scale or be positioned off-grid, which is the capability an exploration agent would need once some regions of a scene require a closer look than others.

Grid dependence also limits existing active-exploration policies. As described in the previous chapter (Section 2.1) most methods use only fixed-size glimpses picked from a grid. However, a notable exception is a family of methods that first captures a single low-resolution overview of the entire scene and then selects a handful of fixed-scale, higher-resolution glimpses on top of it. Several methods were proposed to solve this task by training a reinforcement-learning policy to pick glimpse locations, from the original recurrent attention model [8] to variants specialised for pretrained hard attention [33], remote sensing optimisation [123], and early-exiting classification [129]. Others replace the learned policy with a differentiable saliency-based warping layer [107] or a top-down traversal of image scale-space [91]. Both families fix the scale of every glimpse in advance and only vary its position.

This grid-based assumption does not reflect the physical reality of the platforms that motivate active visual exploration. A simple pan-tilt-zoom (PTZ) camera can freely vary its optical zoom factor, capturing a wide, low-resolution overview of a room or a narrow, high-resolution crop of a single object. An

unmanned aerial vehicle (UAV) can achieve the same effect by changing altitude, and even a robot with a fixed-focal-length camera can emulate variable scale by moving in the environment. What these systems share is a continuous three-dimensional action space over position *and* scale, not a discrete set of fixed-size grid cells.

Beyond the hardware argument, there is an intuitive, information-theoretic case for variable-scale sampling. When an agent begins exploring an unknown scene, it may not yet know where informative content is located. A useful first action is therefore to observe the scene at the *widest possible field of view*. A single low-resolution glimpse that spans the whole scene establishes a global context – rough object locations, dominant regions, salient colour or texture blobs. Then, the agent can identify sub-regions that are most likely to improve its current prediction, and concentrate subsequent, higher-resolution observations there. This *coarse-to-fine* strategy can reduce the number of observations needed to reach a target accuracy because it avoids wasting high-resolution observations on uninformative background before the agent knows where the most informative regions are. This pattern mirrors how biological visual systems combine rapid saccades with a high-acuity fovea [25, 139]. Fixed sampling grid methods cannot utilise this strategy because every glimpse, regardless of when it is taken in the exploration sequence, carries the same spatial extent and the same pixel density.

This chapter addresses this discrepancy between the hardware capabilities and the assumptions embedded in existing methods. Two research questions guide the work:

1. Can a vision transformer backbone be made *elastic* – robust and accurate when patches are of varying scales and non-grid-aligned positions, rather than rigid 16×16 tokens drawn from a fixed grid?

2. Given such an elastic backbone, can a reinforcement-learning policy discover an effective coarse-to-fine exploration strategy that selects glimpses in the fully continuous three-dimensional space of position and scale to achieve more efficient exploration than grid-constrained methods?

The two publications covered in this chapter answer these questions in sequence. “Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers” [II] tackles the first question. It formalises what *input elasticity* means for a ViT and shows that standard architectures are fragile under the perturbations that arise in real-world embodied scenarios. It introduces ElasticViT, a modified architecture and training regime that substantially improves robustness, together with two adaptive patch-sampling strategies that demonstrate how to leverage this elasticity in real tasks. “AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale” [III] then builds on ElasticViT to answer the second question. It formulates glimpse selection over a continuous position-scale action space as a Markov decision process and trains a Soft Actor-Critic agent to solve it, demonstrating state-of-the-art performance on reconstruction, classification, and segmentation while needing to observe substantially fewer pixels than previous methods.

3.2 Preliminaries

Before describing the two publications, we introduce three background concepts that are not covered in the previous chapter, but are necessary to make this thesis self-contained: positional encodings in vision

transformers (Section 3.2.1), reinforcement learning with continuous action spaces (Section 3.2.2) and the Soft Actor-Critic algorithm (Section 3.2.3).

3.2.1 Positional encodings in vision transformers

The transformer architecture processes its input as an unordered *set* of tokens. The self-attention mechanism is permutation-equivariant and contains no built-in notion of where a token sits in the sequence [124]. For language, positional encodings inject order information by adding a position-dependent vector to each token embedding before the first transformer block. For images, they serve the same purpose but must encode spatial location rather than sequence order.

The original ViT [30] uses *learnable* positional embeddings – a separate trainable vector is assigned to each cell of the fixed 14×14 patch grid (for a 224×224 image with 16×16 patches), and these vectors are added to the corresponding patch embeddings. The learned embeddings can capture spatial proximity, relative distances, and even some 2D structure, but only for the specific grid size seen during training. Applying a model with 14×14 learned positional embeddings to a different grid size typically requires interpolation of the embeddings, which can degrade accuracy.

An alternative used in the original Transformer [124] and adapted to images is the *fixed sinusoidal* encoding. For a one-dimensional sequence position pos and embedding dimension index i, the encoding is:

$$\begin{aligned} \mathrm{PE}_{(\mathrm{pos},\,2i)} &= \sin\Big(\mathrm{pos} \,/\, 10000^{2i/d}\Big)\,, \\ \mathrm{PE}_{(\mathrm{pos},\,2i+1)} &= \cos\Big(\mathrm{pos} \,/\, 10000^{2i/d}\Big)\,, \end{aligned} \tag{3.1}$$

where d is the embedding dimensionality. For 2D images, row and column positions can be encoded independently and concatenated (or summed). Because sinusoidal encodings are deterministic functions of position rather than learned parameters, they generalise somewhat better to unseen sequence lengths. A model trained on 196 tokens can be applied to more tokens by simply evaluating the same function at new positions, without any interpolation.

Both approaches, however, share a fundamental limitation for the work presented in this chapter. They map from a discrete, finite set of grid positions to embedding vectors. As Section 3.3 will show, overcoming this limitation, enabling a ViT to process patches at any pixel-level position and at any scale, requires a genuinely continuous positional encoding.

3.2.2 Reinforcement learning with continuous actions

Many sequential decision problems can be formalised as a *Markov decision process* (MDP) [120], a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$ consisting of a state space $\mathcal{S}$, an action space $\mathcal{A}$, a transition distribution $\mathcal{T}(s' \mid s, a)$ specifying the probability of reaching state s' after taking action a in state s, a reward function $\mathcal{R}(s, a)$, and a discount factor $\gamma \in [0, 1]$. The goal of a reinforcement learning (RL) agent is to find a *policy* $\pi(a \mid s)$ (a distribution over actions conditioned on the current state) that maximises the expected cumulative discounted reward over timesteps t:

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \, r_t\right]. \tag{3.2}$$

A central measure used during training is the *action-value function* (Q-function),

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} r_t \mid s_0 = s,\, a_0 = a\right], \tag{3.3}$$

which estimates the expected return when starting from state s, taking action a, and then following policy π thereafter. The Q-function satisfies the *Bellman equation*:

$$Q^{\pi}(s,a) = \mathcal{R}(s,a) + \gamma\, \mathbb{E}_{s' \sim \mathcal{T},\, a' \sim \pi}\left[Q^{\pi}(s',a')\right], \tag{3.4}$$

which expresses the value of a state-action pair as the immediate reward plus the discounted value of all future states. This recursive structure is the basis of most practical RL algorithms: a parameterised critic network $Q_{\phi}(s,a)$ is trained to satisfy Equation 3.4 by minimising the *Bellman residual*, and the policy is then improved by acting greedily with respect to the current critic.

When the action space $\mathcal{A}$ is continuous (as it is in our case), selecting the greedy action requires maximising $Q_{\phi}(s,\cdot)$ over a continuous domain, which cannot be done by exhaustive enumeration. The standard solution is the *actor-critic* architecture. A separate actor network $\pi_{\theta}(a \mid s)$ is trained to produce actions that maximise Q_{ϕ}, while the critic Q_{ϕ} is updated independently via Bellman residual minimisation. Both networks are trained end-to-end with gradient descent.

On-policy algorithms such as REINFORCE [133], used by early recurrent attention models [81], update the actor only from trajectories sampled under the *current* policy. This leads to high variance and poor sample efficiency, particularly problematic when the reward is sparse or delayed, as in exploration tasks where performance only improves after several observations have been made. Experience collected under an older policy cannot be reused directly in the same on-policy update without accounting for the changed sampling distribution.

3.2.3 Soft Actor-Critic

The *Soft Actor-Critic* (SAC) algorithm [45] addresses these challenges through two key ideas. First, it is *off-policy* and maintains a *replay buffer* of past transitions (s,a,r,s') collected under any previous version of the policy, updating both actor and critic by sampling mini-batches from this buffer. This decouples data collection from learning, allows each transition to be reused many times, and greatly improves sample efficiency. Second, SAC augments the reward with an entropy bonus, replacing the standard objective (Equation 3.2) with the *maximum-entropy* objective:

$$J_{\text{SAC}}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t} \gamma^{t}\left(r_t + \alpha\, H(\pi(\cdot \mid s_t))\right)\right], \tag{3.5}$$

where $H(\pi(\cdot \mid s_t)) = -\mathbb{E}_{a\sim\pi}[\log \pi(a \mid s_t)]$ is the differential entropy of the policy at state s_t and $\alpha > 0$ is a temperature coefficient that balances reward maximisation against entropy maximisation. Maximising entropy encourages the agent to maintain a diverse, spread-out distribution over actions rather than prematurely collapsing to a single mode. The agent is therefore penalised for becoming overly deterministic before it has sufficient evidence that one action is clearly superior, which encourages it to explore the action space more.

In practice, the actor network $\pi_{\theta}(a \mid s)$ outputs the parameters (mean and log-variance) of a Gaussian distribution over actions. The continuous action $a \in \mathbb{R}^{d}$ is sampled from this distribution and passed

through a tanh non-linearity to map it into a bounded action range. Actor and critic are trained alternately: the critic minimises the soft Bellman residual against a bootstrapped target computed with a slowly updated *target network*, and the actor maximises the expected soft Q-value (i.e. Q-value plus entropy bonus) via the reparameterisation trick, which allows gradients to flow through the sampling operation.

3.3 Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers

3.3.1 Motivation

As described in Section 3.2.1, a standard ViT relies on learnable positional embeddings indexed by grid position. This design choice is valid in standard benchmarks where every image is processed by the same fixed (most commonly) 14×14 grid of 16×16 patches, but it creates a fundamental brittleness in any deployment where the patch positions or sizes deviate from those seen during training. In real-world settings, a PTZ camera may not snap to discrete zoom levels, and a UAV may fly at an altitude that does not correspond to any training configuration. More importantly, an active vision system may benefit from being able to select a glimpse at a position not aligned to the standard patch grid. This publication asks, as an empirical question, how fragile standard ViTs are under such conditions, and what architectural and training changes are needed to make them robust.

It thereby investigates Hypothesis H2: a vision transformer can be made resilient to input patches sampled at varying positions and scales or providing incomplete coverage, allowing it to process observations that do not follow a rigid grid.

3.3.2 Formalising input elasticity

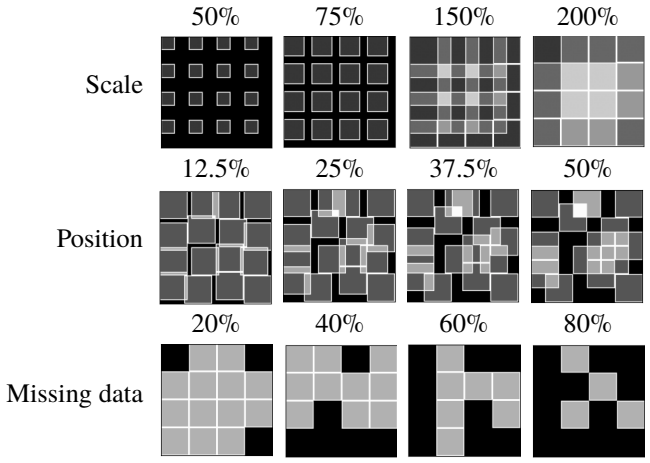


Figure 3.1: **Elasticity perturbations.** Each row illustrates one type of perturbation applied to an example image at increasing intensity. *Scale*: patches are sampled at sizes ranging from 50% to 200% of the native 16×16 size (undersampling introduces gaps while oversampling introduces overlap). *Position*: patches are displaced from their grid positions by up to 50% of the patch size (patch shake). *Missing data*: an increasing fraction of patches is dropped from the input. Adapted from [II].

The publication introduces the concept of *input elasticity* as resilience to particular forms of input data configurations that can occur in real-life applications. Three distinct types of elasticity are defined, each corresponding to a perturbation commonly encountered in embodied AI applications and illustrated in Figure 3.1.

Scale elasticity. This measures resilience when individual patches are sampled at a scale different from the native 16×16 pixel resolution. Each patch $p = (x, y, s)$ is characterised by the pixel coordinates of its top-left corner (x, y) and a relative scale s. The patch covers a $16s \times 16s$ pixel region in the original image, which is then resampled to the standard 16×16 input size. A scale of $s = 2$ means the patch covers four times the usual area at half the resolution, whereas $s = 0.5$ covers one quarter the area at twice the resolution, potentially leaving uncovered gaps between adjacent patches. This corresponds to the zoom ability of a camera.

Positional elasticity. Also called *patch shake*, this measures resilience when patch positions are displaced from the regular grid. The coordinates (x, y) of each patch are independently perturbed by an offset drawn uniformly from $[-r \cdot q,\ r \cdot q]$, where r is the patch size and q is the shake magnitude as a fraction of the patch size. This simulates the non-grid-aligned sampling that arises whenever an image is captured at a position that does not correspond exactly to a discrete training configuration, e.g. due to imperfect camera alignment or agent drift in real space.

Missing-data elasticity. This measures resilience when a fraction d of patches are simply removed from the input, leaving the corresponding image regions unobserved. This is the most directly relevant perturbation for active visual exploration, where an agent has only observed a subset of the scene, or has lost parts of it due to, e.g., occlusion or communication failure.

The three perturbations are composed into a pipeline. For image I, an initial patch set P (standard grid, $s = 1$, no displacement) is processed by

$$P' = \mathcal{E}_{\text{miss}(d)} \circ \mathcal{E}_{\text{pos}(q)} \circ \mathcal{E}_{\text{scale}(s_1,s_2)}(P), \tag{3.6}$$

where $\mathcal{E}_{\text{scale}(s_1,s_2)}$ samples each patch's scale independently from $[s_1, s_2]$, $\mathcal{E}_{\text{pos}(q)}$ shifts each patch's top-left corner by at most q patch-widths in each direction, and $\mathcal{E}_{\text{miss}(d)}$ removes a fraction d of the patches at random. Elasticity is high if predictive performance on the perturbed set P' remains close to that on the original P.

3.3.3 ElasticViT

The modifications proposed in this publication to increase input elasticity are twofold: a new continuous positional encoding that can represent patches at any position and scale, and a change to the training augmentation regime.

Continuous positional encoding. As established in Section 3.2.1, both learnable and standard sinusoidal positional embeddings operate on a finite, discrete set of grid positions. Hence, they cannot represent a patch sampled at an arbitrary pixel-level position or at a variable scale. Our publication addresses

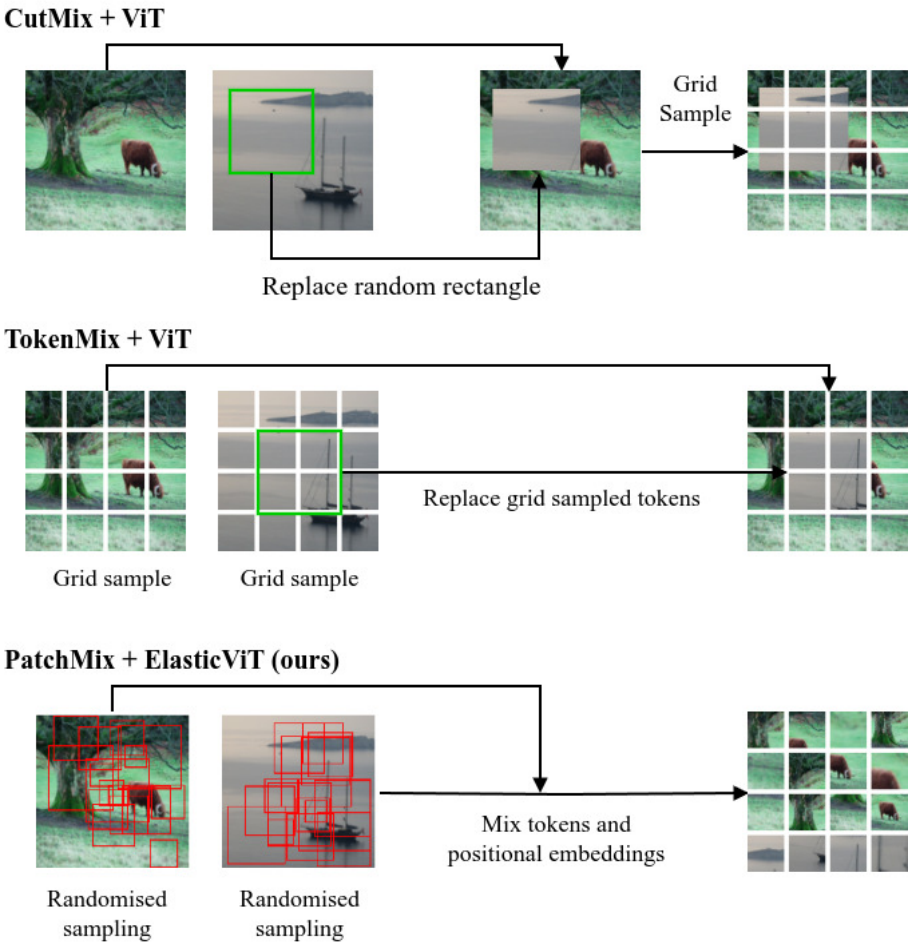


Figure 3.2: **PatchMix.** Unlike CutMix, which replaces a rectangular region of one image with another, PatchMix operates on individually sampled patches and mixes them at the token level. Unlike TokenMix, it does not assume grid-aligned patches, making it compatible with elastic sampling. Reproduced from [II].

this by replacing the learnable embeddings of standard ViT with a deterministic *four-dimensional* sine-cosine encoding. Each patch p is characterised by the pixel coordinates of its upper-left corner (x, y) and its relative scale s (where $s = 1$ corresponds to the native 16×16 patch size). The lower-right corner of the patch is then at $(x + 16s,\ y + 16s)$, and the full spatial extent of p is described by the four scalar values $(x,\ y,\ x + 16s,\ y + 16s)$. Each of these four values is encoded independently with a 1D sine-cosine encoding (Equation 3.1), and the resulting vectors are concatenated to form the patch's positional embedding:

$$\mathrm{PE}(p) = \big[\mathrm{pe}(x) \parallel \mathrm{pe}(y) \parallel \mathrm{pe}(x + 16s) \parallel \mathrm{pe}(y + 16s)\big], \tag{3.7}$$

where $\mathrm{pe}(\cdot)$ is the 1D sine-cosine encoding and $\parallel$ denotes vector concatenation. Because this function is defined over the reals, it generalises without modification to any position-scale combination at inference, including configurations never seen during training. The ViT input is also modified accordingly. Instead of a full rasterised image, the model receives a list of cropped patches together with their corresponding coordinate tuples, which are fed into the positional encoding rather than looked up in a table.

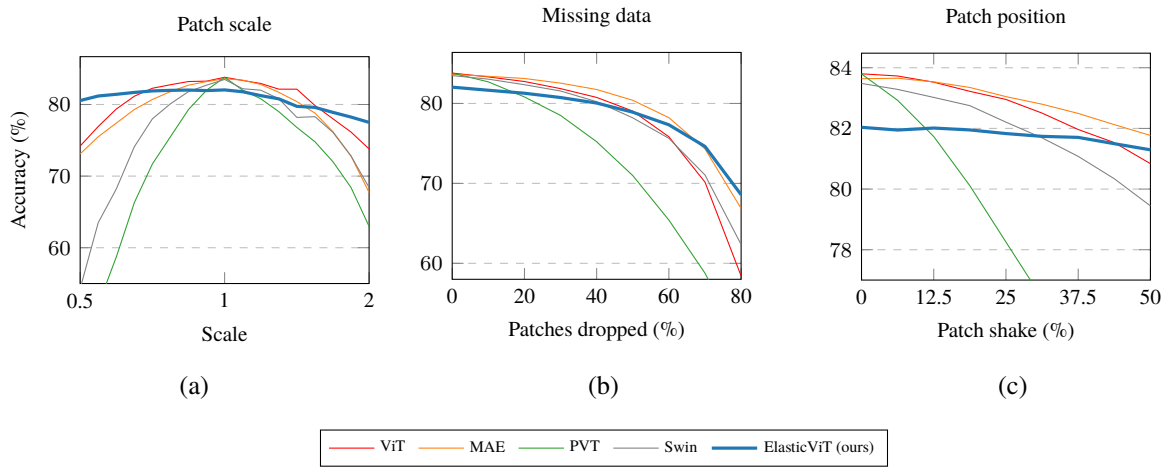


Figure 3.3: **Elasticity under isolated perturbations** (ImageNet-1k top-1 accuracy). **(a)** Scale elasticity: all baselines degrade sharply when patches deviate from scale 1, while our ElasticViT maintains near-constant accuracy across the measured spectrum and outperforms all baselines at extreme scales. **(b)** Missing-data elasticity: ElasticViT surpasses MAE from 70% dropout onward, despite never being trained with patch dropout. **(c)** Positional elasticity: ElasticViT is nearly immune to patch shake and outperforms all but MAE at the largest perturbations. Adapted from [II].

Elastic training augmentations. The baseline training regime follows DeiT III [121]. To this, the publication adds the elasticity functions as augmentations during training. Patch sizes are drawn uniformly between 16×16 and 48×48 pixels, resampled to the 16×16 token size, and patch positions are sampled uniformly throughout the image. This forces the model to generalise across a wide distribution of patch configurations, rather than overfitting to the regular grid it would otherwise always see.

PatchMix augmentation. The standard CutMix augmentation, which mixes two images by replacing a rectangular region of one image with the corresponding region from another, is incompatible with variable-scale patch sampling. The mixing ratios used to blend the class labels assume a fixed spatial layout that no longer holds when patches can overlap or leave gaps. The publication introduces PatchMix, an alternative that performs mixing at the level of individual *patches after sampling*. Patches from two images are randomly interleaved in the token sequence, and the label blending ratio is computed from the actual proportion of patches drawn from each image. Unlike the previously proposed TokenMix [70], PatchMix does not assume grid-aligned sampling and is therefore fully compatible with the elastic augmentation regime (see Figure 3.2).

3.3.4 Experimental results

Elasticity evaluation. Figure 3.3 shows the three single-perturbation elasticity curves, comparing our ElasticViT against ViT, MAE, Swin-B, and PVT-v2-B5, all of comparable size and trained on ImageNet-1k.

Under standard conditions (centre of each curve, scale 1, no dropout, no shake), ElasticViT trails ViT by $\approx$1.8 pp. However, as perturbation intensity increases, the gap reverses and widens considerably. At $2\times$ scale, ElasticViT outperforms ViT by $\approx$3.7 pp and at 80% missing data, it outperforms MAE (the

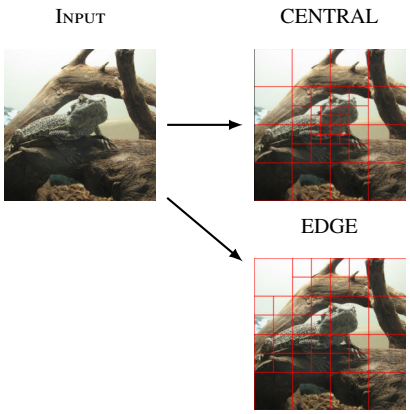


Figure 3.4: **Adaptive sampling strategies.** CENTRAL iteratively subdivides patches closest to the image centre into four smaller ones, producing higher resolution in the foveal region (analogous to retinal sampling). EDGE uses a Canny edge detector [19] to concentrate small patches on high-frequency regions. Reproduced from [II].

strongest baseline in that regime) by ≈1.7 pp. When perturbations are combined, the advantage grows even further: under simultaneous scale and position perturbation at their maxima, ElasticViT leads ViT by nearly 20 percentage points. Standard ViTs, in effect, overfit to the specific grid configuration used during training, while our ElasticViT trades a small amount of peak accuracy on that configuration for robustness across the whole spectrum of input perturbations.

Adaptive sampling. To test the usefulness of ElasticViT for visual exploration problems, the publication proposes two simple, non-learnable baseline sampling strategies: CENTRAL (retina-like sampling) and EDGE (edge-detection-based sampling), illustrated in Figure 3.4. Figure 3.5 shows their effect on accuracy as a function of the number of input patches.

At 16 input patches (a severely limited observation budget directly analogous to the earliest stages of active visual exploration), EDGE sampling with ElasticViT achieves 52.6%, compared to 40.6% for the same model on a standard grid and only 19.8% for rigid ViT on a grid. The rigid ViT does not benefit from CENTRAL sampling in this low-patch regime, because its learnable positional embeddings cannot generalise to the non-grid-aligned positions CENTRAL produces. ElasticViT, by contrast, handles these positions natively through its continuous positional encoding.

Further experimental results are presented in the original paper [II].

3.3.5 Contributions

The publication "Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers" [II] makes the following contributions to the field:

- It formalises the notion of *input elasticity* for vision transformers across three dimensions (scale, position, and missing data) and introduces an evaluation protocol that enables direct comparison of models on each dimension and demonstrates empirically that standard ViTs are fragile under these perturbations, with accuracy dropping sharply even for moderate scale or dropout levels.

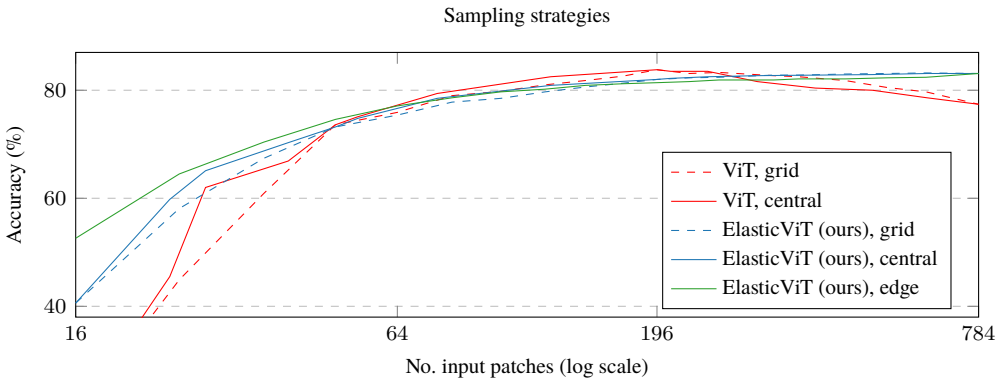


Figure 3.5: **Adaptive sampling strategies vs. standard grid** (ImageNet-1k accuracy). At 16 patches, ElasticViT with EDGE sampling achieves 52.6%, versus 40.6% for ElasticViT on a grid (+12 pp) and 19.8% for ViT on a grid (+32 pp). The rigid ViT does not benefit from CENTRAL sampling because it cannot interpret non-grid patch positions. Adapted from [II].

- It proposes ElasticViT, combining a continuous four-dimensional positional encoding (replacing learnable embeddings) with elastic training augmentations and the new PatchMix augmentation. ElasticViT achieves near-constant accuracy across the full range of scale and position perturbations, and surpasses MAE on missing-data resilience despite never being trained with patch dropout.
- It introduces the CENTRAL and EDGE adaptive patch sampling baseline strategies, and shows that these strategies provide large accuracy gains at low patch counts (up to $+32$ pp over rigid ViT at 16 patches) *only* when combined with an elastic backbone.

The robustness of ElasticViT across the three evaluated perturbation types, together with its gains from adaptive sampling at low patch counts, confirms Hypothesis H2 within the evaluated settings.

I was the first and primary author of this paper. I contributed to the conception of the idea and designed the training regime, positional encoding, and augmentation strategy. I carried out the majority of the experimental design and implementation, conducted the majority of the experiments presented in the paper, and contributed significantly to writing the manuscript. I also presented the research at the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2025 in Tucson, Arizona, USA. The code and data are available on my GitHub at `https://github.com/apardyl/beyondgrids`.

3.4 AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale

3.4.1 Motivation

Beyond Grids (Section 3.3) showed that a ViT can be made elastic enough to accept patches of any position and scale without significant accuracy loss, and that adaptive non-uniform sampling provides a measurable performance advantage at low patch budgets even with simple heuristic sampling strategies. The next step is to replace these hand-designed heuristics with a learned policy that, at each exploration

step, observes the partial information accumulated so far and selects the position and scale of the next glimpse so as to maximise task performance.

As reviewed in Section 3.1, prior AVE methods, whether they pick glimpses directly from a full-scene grid or refine a low-resolution overview with fixed-scale crops, are constrained to discrete, grid-aligned action spaces and cannot exploit variable-scale observations even on hardware that natively supports zoom. AdaGlimpse bridges this gap by formulating glimpse selection over a fully continuous three-dimensional action space of position and scale, and training a Soft Actor-Critic agent [45] to navigate it.

It thereby investigates Hypothesis H3: an agent can learn to select both the position and scale of each observation in a continuous space, discovering an efficient coarse-to-fine exploration strategy that reduces the number of pixels it needs to observe.

3.4.2 Adaptive glimpse sampling

Glimpse parameterisation. Let X be the scene to be explored, represented as a rectangle. A glimpse G is a square region of X captured by a camera with a fixed sensor resolution of $d_{\text{cam}} \times d_{\text{cam}}$ pixels. A glimpse is specified by its normalised top-left coordinates (x, y) and its *scale* $z = (d - d_{\text{min}})/(d_{\text{max}} - d_{\text{min}})$, where d is the physical size of the glimpse in the scene, and d_{min}, d_{max} are the minimum and maximum camera field of view. Intuitively, scale $z = 0$ corresponds to maximum camera zoom (smallest, highest-detail glimpse) and $z = 1$ to the widest possible view (full-scene overview at lowest detail level). Glimpse coordinates $C = (x, y, z) \in [0, 1]^3$ form the continuous action space.

Exploration as a Markov decision process (MDP). The exploration process unfolds over at most T steps. At step t, the agent captures glimpse G_t at coordinates C_t selected by the policy, yielding a $d_{\text{cam}} \times d_{\text{cam}}$ image patch (simulated by cropping from a high-resolution image and rescaling). The agent maintains a growing sequence of observed patches $G_1, \ldots, G_t$ and their coordinates $C_1, \ldots, C_t$. Exploration terminates when either the maximum step count is reached or a task-specific confidence threshold is exceeded (early stopping). Figure 3.6 illustrates the resulting coarse-to-fine exploration behaviour.

3.4.3 AdaGlimpse

AdaGlimpse consists of two interacting components: a vision transformer backbone that processes the accumulated glimpses and produces both task predictions and representations for the RL agent, and a Soft Actor-Critic policy that selects the next glimpse coordinates. The overall design is shown in Figure 3.7.

Glimpse encoder. The encoder is a ViT-B (12 transformer blocks and embedding size 768) with the ElasticViT positional encoding from Section 3.3: each patch is positioned using the four-dimensional continuous sine-cosine encoding of its corner coordinates (Equation 3.7). At step t, the encoder receives all patches from glimpses $G_1, \ldots, G_{t-1}$ and their coordinates, concatenated into sequences $\widehat{G}_t$ and $\widehat{C}_t$. Additionally, the encoder computes an attention-rollout importance score for each patch [2], forming a sequence $\widehat{I}_t$. The transformer blocks produce a sequence of latent tokens $\widehat{H}_t$ (one per patch) plus a class token.

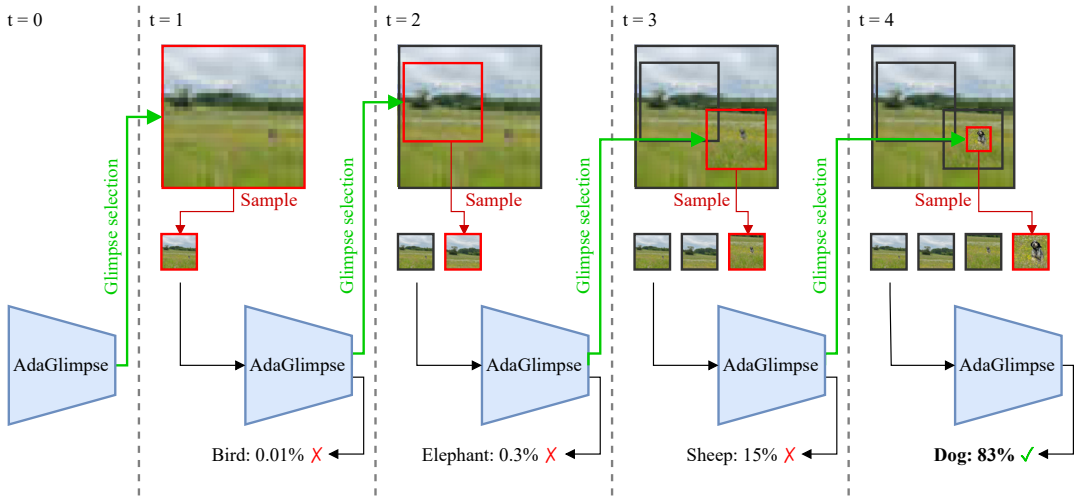


Figure 3.6: **AdaGlimpse in action.** The agent first captures a wide, low-resolution view of the entire scene (scale ≈ 1). Since no confident prediction can be made from this overview alone, it selects a second, smaller glimpse at a higher zoom level focusing on the most informative region. This coarse-to-fine process continues until a class probability exceeds a confidence threshold. Reproduced from [III].

Task-specific decoder. For classification, a linear layer acting on the class token suffices. For dense prediction tasks (reconstruction and segmentation) a MAE-style decoder of 8 transformer blocks (embedding size 512) is used. Unlike standard MAE, which inserts mask tokens only for the unobserved image regions, AdaGlimpse inserts mask tokens for *all* grid positions in the output. This is necessary because variable-scale glimpses can overlap, meaning no fixed partition of "seen" and "unseen" regions exists. Any output pixel may have been observed by multiple glimpses at different resolutions, and the decoder must reconcile these into a coherent full-image prediction.

Training objectives. The backbone is optimised on the task loss computed at the *final* step $t = T$ only (not at intermediate steps). For reconstruction, the loss is RMSE. For classification and segmentation, knowledge distillation against a teacher model (DeiT ViT for classification and DeepLabV3/ResNet-101 for segmentation) is used via KL divergence. This matches the training setup of STAM [106] and enables a fair comparison.

State, action, and reward. The RL agent's state at step t is the tuple $s_t = (\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$: the raw patches, their coordinates, their importance scores, and their backbone latent representations. The action is $a_t = (x, y, z) \in [0, 1]^3$, the continuous coordinates of the next glimpse. The reward is defined as the improvement in the task loss: $r_t = L_{t-1} - L_t$, where L_t is the task loss at step t. This direct signal encourages the agent to select glimpses that maximally reduce the residual task error.

RL agent. The agent uses Soft Actor-Critic with separate actor and critic networks. Each component of the state receives its own encoder: a small CNN for the raw patches $\widehat{G}_t$, and small MLPs for the coordinate, importance, and latent sequences $\widehat{C}_t$, $\widehat{I}_t$, $\widehat{H}_t$. The four resulting per-patch embeddings are concatenated and aggregated across patches via an attention-pooling layer [50]. The resulting fixed-size vector is passed

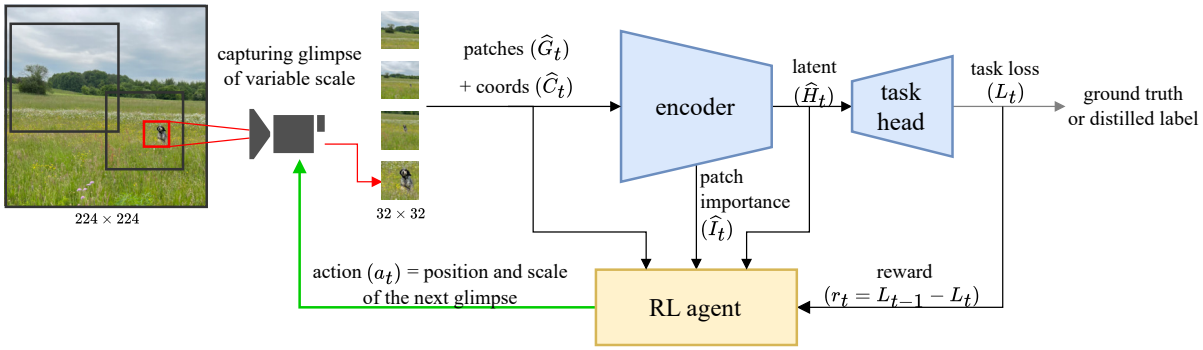


Figure 3.7: **Architecture.** AdaGlimpse combines an ElasticViT encoder with a task-specific decoder (top) and a Soft Actor-Critic RL agent (bottom). At each step, the RL agent selects the position and scale of the next glimpse using the previously observed patches, their coordinates, importance scores, and encoder latent representations. Reproduced from [III].

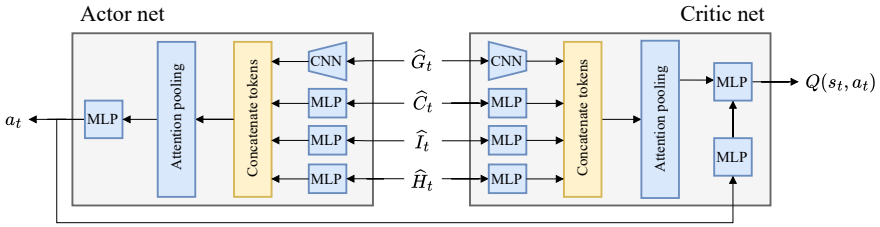


Figure 3.8: **RL agent.** The actor selects the next glimpse position and scale (x, y, z) from state $s_t = (\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$, while the critic estimates the expected cumulative reward $Q(s_t, a_t)$ for that action. Actor and critic share the same input encoding architecture but have separate parameters. Reproduced from [III].

through a final MLP to produce either the action distribution (actor) or the Q-value estimate (critic). The maximum-entropy SAC objective (Equation 3.5) prevents premature convergence to a deterministic greedy policy and encourages the agent to explore diverse glimpse sequences during training. Figure 3.8 shows the RL module.

Training schedule. Pretraining consists of 600 epochs with 196 random glimpses per image sampled from a uniform position-scale distribution. Then, the model is trained for 100 epochs with early stopping: for the first 30 epochs, only the RL agent is updated (allowing it to learn an initial policy), while from epoch 31 onward, backbone and RL agent are optimised in alternating epochs.

3.4.4 Experimental results

Reconstruction. Table 3.1 compares AdaGlimpse with AME [I], AttSeg [114], GlAtEx [115], and SimGlim [52] on image reconstruction (RMSE, lower is better) across four datasets (ImageNet-1k, SUN360 [137], ADE20K [142], and MS COCO [68]) at three pixel budgets. A notable property of

Table 3.1: **Reconstruction results.** RMSE (lower is better) for AdaGlimpse against baselines across four datasets at three pixel budgets. Pixel % denotes the fraction of scene pixels seen by the model. †: reproduced result not published in the relevant paper. *: zero-shot performance (trained on ImageNet-1k only). AdaGlimpse outperforms all baselines at every pixel budget. Adapted from [III].

Method	IMNET	SUN360	ADE20K	COCO	Image res.	Glimpses	Regime	Pixel %
AME	30.3†	29.8	30.8	32.5	128×256	8×32^2	simple	25.00
AdaGlimpse (ours)	**14.5**	**11.1***	**14.0***	**14.5***	224×224	12×32^2	adaptive	24.49
AttSeg	–	37.6	36.6	41.8	128×256	8×48^2	retinal	18.75
GlAtEx	–	33.8	41.9	40.3	128×256	8×48^2	retinal	18.75
AME	–	23.6	23.8	25.2	128×256	8×48^2	retinal	18.75
SimGlim	–	26.2	27.2	29.8	224×224	37×16^2	simple	18.75
AME	–	23.4	26.2	28.6	224×224	37×16^2	simple	18.75
AdaGlimpse (ours)	**14.7**	**11.1***	**14.2***	**14.7***	224×224	9×32^2	adaptive	18.36
AME	–	37.9	40.7	43.2	128×256	8×16^2	simple	6.25
AdaGlimpse (ours)	**20.9**	**17.6***	**20.5***	**21.5***	224×224	12×16^2	adaptive	6.12

AdaGlimpse is that it is trained on ImageNet-1k only. The SUN360, ADE20K, and COCO results represent zero-shot transfer without any per-dataset fine-tuning.

At the ≈6% pixel budget, AdaGlimpse achieves a lower RMSE (17.6–21.5 across datasets) than AME does at ≈18% pixels (23.4–28.6). The coarse-to-fine strategy allows the model to first establish a global scene context with a single wide glimpse and then concentrate remaining observations on the most informative regions, which is impossible with a fixed-scale policy.

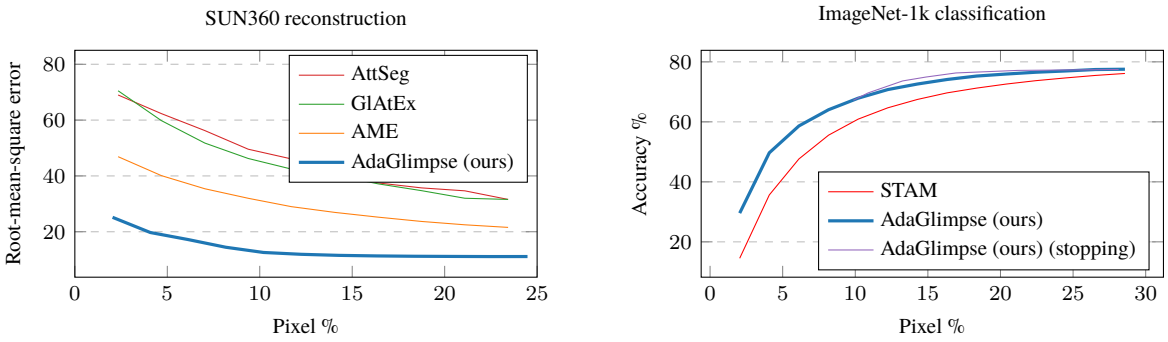


Figure 3.9: **Pixel % vs. performance.** *Left*: SUN360 reconstruction RMSE (lower is better) as a function of the fraction of pixels observed. AdaGlimpse surpasses all baselines at any pixel budget and achieves better reconstruction with 5% of pixels than AME achieves with 25%. *Right*: ImageNet-1k classification accuracy. AdaGlimpse exceeds STAM at every budget, and the early-stopping variant matches STAM's accuracy using ∼40% fewer pixels. Adapted from [III].

Table 3.2: **Classification results.** Top-1 accuracy on ImageNet-1k. AdaGlimpse achieves the best accuracy at the same pixel budget as STAM (28.57%), and with early stopping at 85% confidence, reaches STAM-level accuracy using ∼40% fewer pixels. Adapted from [III].

Method	Accuracy %	Glimpses	Regime	Pixel %
DRAM	67.50	8×77^2	full+simple	94.53
GFNet	75.93	5×96^2	full+simple	91.84
Saccader	70.31	6×77^2	full+simple	70.90
TNet	74.62	6×77^2	full+simple	70.90
STN	71.40	9×56^2	full+simple	56.25
PatchDrop	76.00	$\sim 8.9 \times 56^2$	full+simple+stopping	∼55.63
STAM	76.13	14×32^2	simple	28.57
AdaGlimpse (ours)	**77.54**	14×32^2	adaptive	28.57
AdaGlimpse (ours)	**76.30**	$\sim 8.3 \times 32^2$	adaptive+stopping	∼**16.94**

Classification. Table 3.2 shows classification results on ImageNet-1k against a broad set of baselines including methods that take a low-resolution overview plus fixed-scale glimpses (DRAM [8], GFNet [129], Saccader [33], TNet [91], STN [107], and PatchDrop [123]) and the state-of-the-art grid-based method STAM [106].

Pixel efficiency. Figure 3.9 shows the relationship between the fraction of image pixels observed and task performance, sweeping over the number of glimpses. AdaGlimpse consistently outperforms baselines across the full pixel-budget spectrum. In reconstruction, it achieves lower error with 5% of pixels than competing methods achieve with 25%.

Qualitative analysis. Figure 3.10 shows two representative classification sequences. The policy takes an initial wide-field glimpse, then progressively zooms in on discriminative regions, and halts as soon as the confidence threshold is exceeded, illustrating the coarse-to-fine strategy emerging from the learned policy without any explicit inductive bias towards it.

Further experimental results are presented in the original paper [III].

3.4.5 Contributions

The publication "AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale" [III] makes the following contributions to the field of active visual exploration:

- It introduces AdaGlimpse, a novel AVE method that selects glimpses in a fully continuous three-dimensional position-scale action space, overcoming the grid constraint shared by prior approaches. AdaGlimpse is able to learn a coarse-to-fine exploration strategy without any explicit inductive bias towards it.

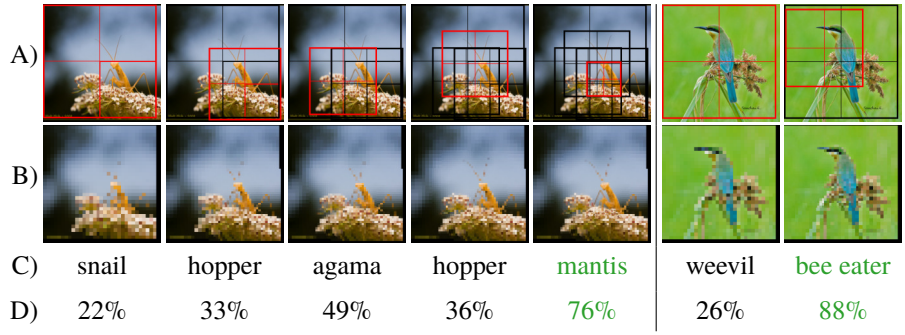


Figure 3.10: **Glimpse selection step-by-step.** AdaGlimpse explores 224×224 ImageNet images using 32×32 glimpses of variable scale, stopping once the predicted probability exceeds 75% (a lower threshold than the 85% used for the quantitative results in Table 3.2, chosen here to keep the illustrated episodes short). The vertical bar separates two independent examples. Row A: glimpse locations overlaid on the original image. Row B: pixels visible to the model (interpolated for display). Row C: current predicted label. Row D: prediction confidence (green = correct and above threshold). Reproduced from [III].

- It formulates AVE as a Markov decision process with a four-component state space $(\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$ and trains an exploration policy with the Soft Actor-Critic algorithm, whose maximum-entropy objective encourages diverse, exploratory policies. The policy operates on a single task-agnostic transformer backbone, in which only the decoder head changes between reconstruction, classification, and segmentation.

- It demonstrates state-of-the-art performance across three visual tasks and four datasets. In reconstruction, AdaGlimpse achieves better results with 6% of pixels than competing methods achieve with 18%. In classification, it surpasses the best prior method (STAM) in absolute accuracy and can match STAM's accuracy using 40% fewer pixels with early stopping. In segmentation, it matches AME while observing 35% fewer pixels.

The learned coarse-to-fine behaviour and the improvements in pixel efficiency across reconstruction, classification, and segmentation confirm Hypothesis H3 within the evaluated settings.

I was the first and primary author of this paper. I contributed to the conception of the idea and proposed the reinforcement learning formulation, including the state and action spaces and reward function, as well as the agent architecture. I carried out the majority of the experimental design and implementation, conducted roughly half of the experiments presented in the paper, and contributed significantly to writing the manuscript. I also presented the research at the European Conference on Computer Vision (ECCV) 2024 in Milan, Italy. The code, data and further visualisations are available on my website at `https://io.pardyl.com/AdaGlimpse/`.

Chapter 4

Open-world exploration

4.1 Research scope

Chapter 3 demonstrated that, in the evaluated settings, a learned policy can perform efficient active visual exploration in a fully continuous observation space. AdaGlimpse, built on the ElasticViT backbone, enables an agent to direct its attention anywhere in a scene at any resolution, and through reinforcement learning it discovers coarse-to-fine strategies that maximise information gain for a specified number of observed pixels. Yet the same geometrical assumption runs through all three preceding publications – each scene is bounded and visible from a single position. The agent's task is to decide where to look within it. The entire environment is, in principle, reachable without physical relocation.

Real-world embodied agents rarely operate under these conditions. A UAV conducting a search-and-rescue mission over a forest, a robot looking for a misplaced item in a large warehouse, and a rover mapping unfamiliar terrain all share a challenge that is qualitatively different from glimpse selection. The target may not be within the current field of view at all, and acquiring a new observation requires physical movement through three-dimensional space. The agent therefore must form and execute a search strategy in a world that extends well beyond what it can sense from any single position, while each movement consumes time and energy that cannot be recovered.

Vision-language models (VLMs) are a natural candidate for meeting this challenge. Pretrained on internet-scale image-text corpora, they encode broad world knowledge. This also includes human-intuitive knowledge such as where objects tend to occur and what visual cues to look for. Therefore, they are increasingly deployed as the reasoning core of general-purpose embodied and agentic systems [16, 85], without requiring the task-specific training that a dedicated navigation policy would need.

Existing evaluation practice does not fully answer the question of whether today's models are equipped for this kind of search. Most published assessments of VLMs rely on static benchmarks, such as single-image visual question answering (VQA) or visual entailment tasks, that never require the model to decide what to observe next [43, 74]. The comparatively few interactive benchmarks that place a model in a control loop tend to focus on game-like environments or instruction-following in indoor simulators, where the state space is small and strong structural cues are available [72, 90]. The object-goal navigation task offers a closer analogue, but the field's dominant indoor simulators were designed to evaluate purpose-built navigation agents rather than general-purpose foundation models. Their structured, room-bounded layouts reward methods that build an explicit floor map rather than genuinely exploring unknown

spaces [54, 111, 136]. On the other side, the handful of outdoor aerial navigation environments that exist, such as OUTDOOR [138], restrict the agent to two-dimensional horizontal movement, omitting the altitude dimension that, similarly to the coarse-to-fine exploration problem studied in Chapter 3, is central to the search problems with which this chapter is concerned.

This chapter takes a step back and asks a core question: do the models evaluated in the publication possess the capabilities required for this kind of open-world, goal-directed visual search? The publication presented in this chapter, "FlySearch: Exploring how vision-language models explore" [IV], describes a benchmark rather than a method. It presents a realistic evaluation environment with a structured set of challenges that make the question concrete and measurable, demonstrating that even the strongest proprietary models remain well below human performance, with the gap widening as tasks demand longer, more consistent exploration.

4.2 Preliminaries

Before describing the publication, for completeness, we introduce four background concepts not addressed in previous chapters: vision-language models (Section 4.2.1), Group Relative Policy Optimisation (Section 4.2.2), the object-goal navigation task that FlySearch is based on (Section 4.2.3), and the sensing properties of UAVs that shape the search problem (Section 4.2.4).

4.2.1 Vision-language models

Vision-language models (VLMs; often called MLLMs or multimodal large language models) are neural networks trained to reason over joint visual and textual inputs. The field is young and the space of architectures is still being actively explored. Alternatives to the standard transformer language backbone exist, including ones built on Mamba state-space models [99] or on diffusion rather than autoregressive decoding [140]. A full survey of these architectural choices is beyond the scope of this section. The distinction most relevant to this chapter is not the backbone itself but *when* multimodal data enters the language backbone's training. Under the *text-first, then aligned* recipe, a language model and an image encoder are pretrained independently and only brought together afterwards, connected by a learned module and jointly fine-tuned to align the two modalities. Under the *jointly pretrained* recipe, the language backbone itself is trained on multimodal data from the very start of its own pretraining, rather than being retrofitted after the fact.

Models following the text-first, then aligned recipe are the historically dominant choice in practice. A vision transformer (ViT) [30], typically pretrained with a contrastive image-text objective such as CLIP [101], encodes the input image into a sequence of visual tokens. A separate connector module then maps these tokens into the embedding space of a large language model separately pretrained on text data. The visual tokens are concatenated or interleaved with text tokens. The complexity of this connector varies widely across models. LLaVA [69] uses the simplest form – a single linear projection bridging a frozen CLIP encoder to a pretrained Vicuna language model, with only the projector trained during an initial alignment stage. Qwen2.5-VL and Qwen3-VL [9, 10] push the same paradigm further. The vision encoder natively supports variable input resolutions rather than resizing to a fixed size, an MLP merger performs the projection, and a rotary positional scheme (M-RoPE) [126] encodes the joint spatial and

temporal structure of the visual tokens. The latest generation additionally injects visual features from multiple layers of the vision encoder into corresponding layers of the language model, rather than only at the input. These designs reuse a pretrained language backbone and vision encoder. Multimodal alignment can update the connector and vision encoder while the language model is frozen. Moreover, during later fine-tuning stages, all components can be trained jointly.

Models following the jointly pretrained recipe are comparatively rare among openly documented systems, since assembling a large multimodal pretraining corpus from the outset is a substantially larger undertaking than reusing an existing text-only checkpoint. Gemini [40] is a documented example. It is described as natively multimodal, trained jointly across interleaved text, image, audio, and video data from the beginning of pretraining, rather than adapted from a text-only model afterwards. The recipe is a property of how a specific model was trained rather than a fixed trait of the organisation or model family behind it. Whereas Qwen2.5-VL and Qwen3-VL, described above, are built by bridging a pre-existing text-only Qwen checkpoint to a vision encoder, their successor Qwen3.5 [100] instead trains its language backbone from scratch on text, image, and video data simultaneously. Because multimodal data shapes the language backbone's representations throughout pretraining rather than only during a later alignment stage, this recipe can shape language representations around multimodal tasks from the start.

In all the autoregressive VLMs considered here, once visual and textual information share a single token sequence, the decoder attends over it and generates output text (one token at a time). Modern VLMs extend this basic mechanism to multi-turn conversations containing multiple images, maintaining a record of all past visual and textual context in their context window and generating responses that draw on the full history of what has been seen and said [67]. Because these models can generate text, no trained task-specific head is required for issuing action commands. Any structured command an application calls for, such as a navigation instruction, can be generated directly as text and parsed downstream.

A further *post-training* stage, applied regardless of pretraining paradigm, is what turns a raw next-token predictor into a model that reliably follows instructions rather than merely continuing text. The standard recipe begins with supervised fine-tuning on curated instruction-response pairs, followed by a preference-based stage that further shapes behaviour using feedback on candidate outputs [24]. Reinforcement Learning from Human Feedback (RLHF) [89] trains a reward model on (usually human) judgements between pairs of candidate responses and then optimises the policy against it. Direct Preference Optimisation (DPO) [102] reformulates this same objective as a closed-form classification loss over preference pairs, letting a model be tuned directly on pair-wise preferences instead. An alternative recipe, Reinforcement Learning with Verifiable Rewards (RLVR) [57, 117], replaces pair-wise preference input with a directly computable, rule-based reward for tasks with a checkable outcome, such as a correct answer or a successfully completed action. Section 4.2.2 describes Group Relative Policy Optimisation, the RLVR algorithm used in the presented publication to fine-tune a VLM for exploration.

4.2.2 Group Relative Policy Optimisation

Standard actor-critic policy-gradient methods for language models, such as Proximal Policy Optimisation (PPO) [112], estimate the advantage of a sampled action relative to a learned value baseline produced by a separate critic network (Section 3.2.2). For a large language or vision-language policy, this critic is typically a network of comparable size to the actor itself, which roughly doubles the memory and

compute required for training and adds a second model that must itself be kept accurate throughout training. *Group Relative Policy Optimisation* (GRPO) [117] removes this requirement by leveraging the generative property of LLMs, estimating the baseline empirically from a group of samples drawn for the same prompt or state, rather than learning it.

Given a state (prompt) q, the current policy $\pi_{\theta_{\text{old}}}$ samples a group of G candidate outputs $\{o_1, \ldots, o_G\}$, each scored by a reward function to produce $\{r_1, \ldots, r_G\}$. Each sample's advantage is then computed by normalising its reward against the group's own statistics, rather than against a separately learned baseline:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}. \tag{4.1}$$

For a group with zero reward variance, the normalised advantage is set to zero to avoid division by zero. Samples that score above the group average receive a positive advantage and have their probability increased; samples that score below it receive a negative advantage and have their probability decreased. The policy is updated using a clipped surrogate objective, following PPO, which discourages large changes in sampled-action probability ratios between π_θ and $\pi_{\theta_{\text{old}}}$, together with a KL-divergence penalty that further regularises π_θ towards a fixed reference policy (typically the model before RL fine-tuning began), reducing the risk of drifting towards degenerate outputs that exploit the reward.

In practice, each group member o_i corresponds to a complete rollout – a full generated response for a single-turn language task, or, in a sequential decision setting, an entire multi-step conversation/trajectory. Because the advantage is computed empirically within each group, no critic network is trained at all, and the only network updated during training is the policy itself.

4.2.3 Object-goal navigation

Object-goal navigation (ObjectNav) [14] is a task in which an embodied agent, starting from a random initial position in a simulated environment, must navigate to an instance of a specified object. The agent receives egocentric visual observations and a goal specification (a category label or more recently a natural-language description), and must issue movement commands until it is sufficiently close to the target, at which point it signals success with a STOP or FOUND action. The task is different from passive visual search because the agent must actively move through the environment to acquire new observations, making decisions about where to go next based on incomplete, egocentric information.

The dominant simulators for ObjectNav (Habitat-Sim [111], AI2-THOR [54], and Gibson [136]) are almost exclusively indoor. These environments feature relatively small, structured layouts bounded by rooms and corridors, where the total navigable area is limited and explicit floor plans can be constructed and used to guide search. Language-conditioned ObjectNav [39, 77] extends the standard formulation by replacing discrete category labels with free-form natural language descriptions, requiring the agent to reason about object identity without a pre-defined vocabulary of targets. This is necessary for any deployment in which the set of possible objects is not known at design time. FlySearch builds on this language-conditioned formulation, but moves it out of the small, structured indoor environments described above and into the large, unstructured outdoor setting.

4.2.4 Aerial visual search

Unmanned aerial vehicles (UAVs) offer a distinctive sensing modality. By controlling altitude, the UAV agent controls the trade-off between field of view and spatial detail. At high altitude, the camera captures a large ground area in a single image but at low resolution, making it possible to survey a wide area quickly at the cost of fine detail. At low altitude, the field of view narrows but small or visually ambiguous targets become identifiable. An effective search strategy needs to manage this trade-off dynamically, flying high to establish a broad situational picture and descending to confirm candidate locations.

This dynamic has a structural parallel to the coarse-to-fine exploration studied in Chapter 3, where the agent controlled the scale of its glimpses to alternate between wide-context overviews and high-resolution inspections of specific regions. In the UAV setting, however, scale variation is achieved through physical movement rather than optical zoom. The agent must fly to a different altitude to change its effective resolution, which introduces additional complexity. Changing altitude takes steps, and descending to obtain detail over one region means sacrificing coverage of other regions. A rational search strategy must therefore reason about when the evidence gathered at high altitude is sufficient to justify the cost of a closer inspection, and when it is more valuable to survey from above.

The 3D action space means that the challenge is as much conceptual as computational. The agent must form and maintain a mental model of what has been explored at which altitude and what regions remain uncertain. Whether VLMs, whose architectures and training objectives were not designed with this kind of spatial bookkeeping in mind, can perform these computations effectively is a central empirical question addressed by FlySearch.

4.3 FlySearch: Exploring how vision-language models explore

4.3.1 Motivation

Despite the rapid development of vision-language models as zero-shot tools across a wide range of perception and reasoning tasks, their ability to perform goal-directed exploration in open three-dimensional environments has received little systematic attention. As discussed in Section 4.1, static benchmarks do not require active observation selection, while interactive benchmarks typically focus on games or structured indoor environments. Even outdoor aerial navigation benchmarks such as OUTDOOR restrict movement to the horizontal plane, excluding altitude changes that are essential to coarse-to-fine search. FlySearch addresses these gaps directly – it provides a simulated photorealistic outdoor environment with a three-dimensional action space, a zero-shot language-conditioned search task, and a suite of standardised challenges that isolate distinct aspects of exploration capability. By evaluating VLMs directly rather than as components within a larger navigation system, FlySearch measures their exploration capability in isolation.

The evaluation investigates Hypothesis H4: current vision-language models can perform short-horizon visual navigation, but cannot reliably form and execute consistent multi-step search strategies in large, open, three-dimensional environments.

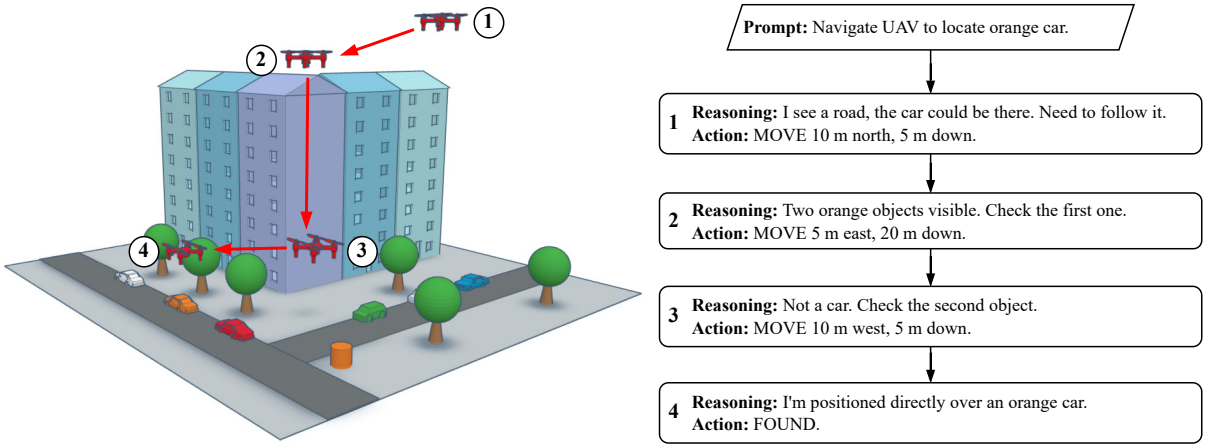


Figure 4.1: **FlySearch** evaluates exploration capabilities of vision-language models in a realistic outdoor setting. The agent controls a UAV by observing images captured from successive positions and providing text commands describing the next movement. To complete each scenario, the model must locate an object specified in natural language and signal success with a `FOUND` command. Reproduced from [IV].

4.3.2 Benchmark design

Task formulation. In FlySearch, the agent controls a simulated multirotor UAV equipped with a downward-facing camera. The UAV serves as a proxy for evaluation of spatial exploration capability, rather than low-level control of a physical robot, and is therefore controllable with high-level relative movement commands, resembling autopilot inputs. The agent is tasked with finding a target object described in natural language or with a visual example. The interaction follows a fixed protocol designed to be equally natural for human testers and VLMs, outlined in Figure 4.1.

At the start of each episode, the model receives a *starting prompt* that contains a brief textual description of the object to be located (e.g. *a red pickup truck*) and in some cases a visual example of the target object. The starting prompt also specifies the communication format, and an instruction that the model may preface each response with a chain of reasoning before issuing its action. This allows the model to "think aloud" before committing to a movement [130].

At each subsequent step, the agent receives an *observation* consisting of a 500×500 pixel RGB image from the UAV camera, overlaid with a coordinate grid [63] to help the model judge distances and directions, together with the current altitude above the ground.

The agent responds with an *action*: a text command `<action>(X, Y, Z)</action>` specifying a relative displacement in metres in the three spatial directions, or the terminal command `FOUND` when it believes the target is in view. Low-level flight control (collision avoidance, trajectory execution) is handled by a simulated autopilot layer, so the model focuses entirely on exploration decisions.

An episode is considered successful if, at the moment the agent signals `FOUND`, the object's centre point lies within the camera's field of view *and* the agent is at most 10 m above the object's highest point. This metric, adapted from the standard ObjectNav criterion [14], requires the agent to be close enough for meaningful visual confirmation rather than merely in the neighbourhood of the target.

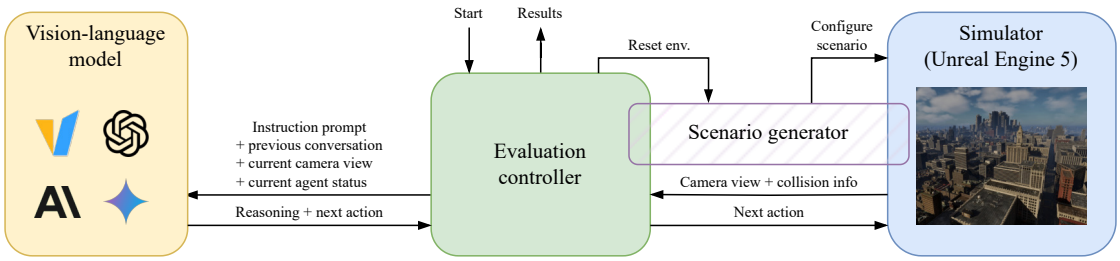


Figure 4.2: **Evaluation pipeline.** FlySearch consists of three main components: the Unreal Engine 5 simulator that renders near-photorealistic views of the open-world map, the evaluation controller that manages communication between the VLM and the simulator, and the scenario generator that procedurally creates new evaluation scenarios. Reproduced from [IV].

Evaluation pipeline. The FlySearch system, shown in Figure 4.2, consists of three main components: a simulator, an evaluation controller, and a scenario generator. The *simulator* is built on Unreal Engine 5 [36], which provides near-photorealistic graphics through real-time ray tracing and dynamic global illumination. Large, open-world maps are supported natively, and the procedural content generation system can place tens of thousands of object meshes in seconds at runtime, enabling an unlimited number of distinct evaluation scenarios. The *evaluation controller* is implemented in Python and handles the full lifecycle of the benchmarking process: managing the communication between the simulator and the VLM under test, enforcing episode termination conditions, and computing performance metrics. The controller supports a wide range of VLMs through a simple adapter interface, and additional models can be integrated using the open-source vLLM inference server [56]. The *scenario generator*, functionally integrated with the other two components, procedurally generates new evaluation scenarios: it builds the forest map entirely at runtime and spawns the distractor assets of the city environment, so that no two scenarios share the same layout.

Environments. The benchmark provides two distinct evaluation environments. The *forest environment* is based on the Electric Dreams UE5 demo [35] and features sparse forest scenery with randomly placed rock formations, procedurally generated vegetation, and dynamic wind. The entire map is generated at runtime, ensuring that no two scenarios share the same layout. The *city environment* is based on the City Sample UE5 demo [34], a large, semi-procedurally generated American-style city of approximately 4×4 km, populated at runtime with random distractor assets (parked cars, walking pedestrians) drawn from Unreal's asset marketplace. The visual complexity of the city environment is substantially higher than that of the forest environment, with many objects of similar appearance and strong visual clutter. Figure 4.3 shows example scenes from both environments.

Challenge sets. In addition to the procedurally generated environments, three standardised challenge sets are defined, each designed to stress a different aspect of exploration. To ensure reproducibility, the scenarios in those sets are pre-generated to ensure all models are evaluated in the same conditions.

FS-1 contains 400 scenes across forest and city environments. The agent has at most 10 actions to find the target object. The starting altitude is between 30 and 100 m, the search area is bounded to

(a) Forest environment.

(b) City environment.

Figure 4.3: **Evaluation environments.** Each column shows a wide-angle environment overview (top) and a selection of top-down views of target objects from the UAV camera (bottom), illustrating the visual fidelity of the simulation and the variety of objects the agent must locate. Reproduced from [IV].

$400 \times 400 \times 120$ m, and crucially, the target is always within the camera's field of view at the starting position (though it may be too small to identify from above). Models may not move beyond their current field of view in FS-1. This setting tests basic perception, spatial reasoning, and short-horizon navigation. City targets include road construction works, crowds, large trash piles, fires, and vehicles of specified types and colours. Forest targets include campsites, trash piles, persons, forest fires, and buildings.

FS-Anomaly-1 contains 200 scenes using the same setup as FS-1 but with a twist: the agent must find an object that does not belong in its environment (for example, a giraffe wandering through a city or a flying saucer in a forest) without being told which specific object is out of place. The agent must first determine what constitutes an anomaly and then locate it. This tests contextual world knowledge on top of basic search capability.

FS-2 contains 200 harder city scenes in which the starting altitude is raised to 100–125 m and dynamic lighting simulates different times of day. Most importantly, the target object may be completely outside the initial field of view. The agent may also move beyond its current field of view at each step. It has a budget of 20 actions, a maximum altitude of 300 m, and horizontal search bounds of $2h \times 2h$, where h is the starting altitude. This setting requires the agent to form and execute a systematic multi-step search strategy, exploring the environment methodically until the target is found.

4.3.3 Experimental results

Evaluation protocol. Ten VLMs are evaluated on FS-1 and FS-Anomaly-1, with a subset also evaluated on FS-2. Three proprietary models are included: GPT-4o [87], Claude 3.5 Sonnet [6], and Gemini 2.0 Flash [42]. Three large open-weight models (above 11B parameters) are evaluated: Qwen2-VL-72B [126], LLaVA-OneVision-72B [65], and Pixtral Large (124B) [80]. Four small open-weight models (below 11B parameters) are evaluated: Phi-3.5-vision [1], InternVL2.5-8B-MPO [127], LLaVA-NeXT-Interleave-

Table 4.1: **FlySearch benchmark results:** Success rates (± standard errors) on FS-1 and FS-2. Gemini 2.0 Flash achieves the highest overall score on FS-1, while all VLMs collapse on FS-2 relative to the human baseline. The last row shows a model fine-tuned on the Forest environment. Its Forest result is shown in grey because it may benefit from distribution overlap.

Model	FS-1			FS-2
	Overall (%)	Forest (%)	City (%)	Overall (%)
Human (untrained)	–	–	66.7 ± 4.5	60.8 ± 6.9
GPT-4o	39.5 ± 2.4	45.5 ± 3.5	33.5 ± 3.3	3.5 ± 0.9
Claude 3.5 Sonnet	41.2 ± 2.5	$\mathbf{52.0 \pm 3.5}$	30.5 ± 3.3	$\mathbf{6.5 \pm 1.2}$
Gemini 2.0 Flash	$\mathbf{42.0 \pm 2.5}$	42.5 ± 3.5	$\mathbf{41.5 \pm 3.5}$	6.0 ± 1.1
Phi-3.5-vision	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	–
InternVL2.5-8B-MPO	2.0 ± 0.7	2.5 ± 1.1	1.5 ± 0.9	–
LLaVA-NeXT-Interleave-7B	0.8 ± 0.4	0.0 ± 0.0	1.5 ± 0.9	–
Qwen2.5-VL-7B	3.8 ± 1.0	6.0 ± 1.7	1.5 ± 0.9	0.0 ± 0.0
Qwen2-VL-72B	17.2 ± 1.9	16.5 ± 2.6	18.0 ± 2.7	–
LLaVA-OneVision-72B	9.5 ± 1.5	12.5 ± 2.3	6.5 ± 1.7	–
Pixtral Large	29.8 ± 2.3	38.0 ± 3.4	21.5 ± 2.9	3.0 ± 0.8
Qwen2.5-VL-7B, GRPO on Forest	–	57.0 ± 3.5	27.0 ± 3.1	0.0 ± 0.0

7B [67], and Qwen2.5-VL-7B [10]. Human baselines are collected via an online study, comprising 111 recorded episodes for FS-1 City and 51 for FS-2.

Model success rates are computed separately for each target class and then averaged across classes. Within each class, the standard error is the sample standard deviation of binary episode outcomes divided by the square root of the number of episodes. The reported model uncertainty is the mean of these per-class standard errors.

FS-1 results. Table 4.1 reports success rates for FS-1 and FS-2. On FS-1, the best-performing model is Gemini 2.0 Flash at 42.0%, closely followed by Claude 3.5 Sonnet at 41.2% and GPT-4o at 39.5%. Human participants, evaluated on the city environment, achieve 66.7%, approximately 25 percentage points above the best VLM on the same split. Among large open-weight models, Pixtral Large performs best at 29.8%, with Qwen2-VL-72B at 17.2% and LLaVA-OneVision-72B at 9.5%. The small open-weight models essentially fail, with no model exceeding 4%. Investigation reveals that small models largely fail to follow the instruction format, often neglecting to issue the `FOUND` command even when the target is within view at the end of an episode.

FS-2 and the exploration gap. The most striking result concerns FS-2. Human participants score 60.8% (a modest drop from their FS-1 City score of 66.7%), reflecting that systematic search is a natural cognitive strategy that humans execute reliably when given sufficient steps. VLMs collapse dramatically. The best-performing model on FS-2 is Claude 3.5 Sonnet at just 6.5%, and GPT-4o and Gemini achieve 3.5% and 6.0% respectively. The relative gap between the best VLM and humans grows from roughly 60% on FS-1 to more than 800% on FS-2.

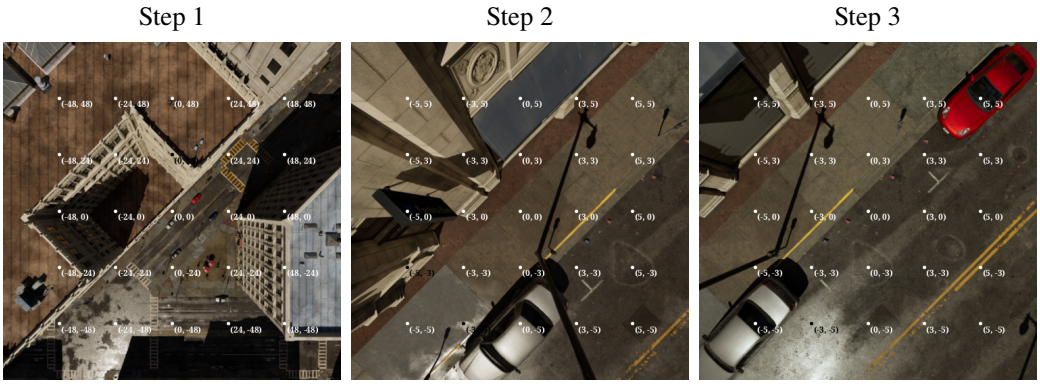


Figure 4.4: **Example successful trajectory on FS-1.** GPT-4o navigates to a red sports car by first descending (step 1) and then adjusting laterally (step 2) before claiming success. Note the coordinate grid overlay on the camera images, which helps the model estimate distances. Reproduced from [IV].

This collapse cannot be attributed solely to the harder detection conditions. The core problem is that FS-2 requires a genuine search strategy, covering the environment systematically and tracking which areas have been visited. Human participants intuitively begin following streets, grid-scanning districts, and ascending to survey larger areas when uncertain. VLMs in contrast tend to move in inconsistent, locally reactive patterns, often repeating previously visited regions, failing to track coverage, and being easily derailed when they lose sight of a candidate object. This behavioural gap is further amplified by context-length degradation. Models given 20 steps instead of 10 score lower, which was isolated in an ablation study.

Qualitative analysis. Manual inspection of failure trajectories reveals consistent failure modes across all models. When a model loses sight of an object it was tracking, it often backtracks or begins hallucinating the object's presence at an incorrect location, rather than reasoning about where the object was last seen and moving towards it. Collision handling is universally poor, with all models occasionally attempting to repeat an action that was rejected by the collision avoidance system, rather than navigating around the obstacle. In several recorded trajectories, Gemini 2.0 Flash abandons the task entirely after a collision, noting "I am unable to navigate without hitting the building." Figure 4.4 shows an example of a successful trajectory, illustrating the descent-and-approach strategy that works well on FS-1 but is insufficient when the target is not visible from the starting position in FS-2.

FS-Anomaly-1 results. Table 4.2 shows results on FS-Anomaly-1. Scores are substantially lower than on FS-1: the best model, Gemini 2.0 Flash, achieves only 35.5% overall, compared with 42.0% on FS-1, and performance is notably weaker in the city than in the forest. The gap reflects an additional challenge beyond search: the agent must first determine *which* object in the scene is out of place before it can locate it. VLMs consistently make errors in this judgement, often flagging visually distinctive but contextually normal objects (a yellow taxi in a city scene) while ignoring semantically obvious anomalies (a tank standing next to it). Naming the anomaly object explicitly, eliminating the need for contextual judgement,

Table 4.2: **FS-Anomaly-1 results:** Success rates (± standard errors) of the evaluated models. Scores are substantially lower than on FS-1, reflecting the additional challenge of identifying which object is out of place.

Model	Overall (%)
GPT-4o	27.0 ± 3.1
Claude 3.5 Sonnet	27.5 ± 3.2
Gemini 2.0 Flash	$\mathbf{35.5 \pm 3.4}$
Phi-3.5-vision	0.0 ± 0.0
InternVL2.5-8B-MPO	3.5 ± 1.3
LLaVA-NeXT-Interleave-7B	0.0 ± 0.0
Qwen2.5-VL-7B	2.8 ± 1.2
Qwen2-VL-72B	7.5 ± 1.9
LLaVA-OneVision-72B	8.5 ± 2.0
Pixtral Large	15.0 ± 2.5

raises Gemini's score from 35.5% to 46.5%, confirming that the primary bottleneck is semantic reasoning rather than visual search.

Fine-tuning analysis. To investigate whether fine-tuning can close the gap, Qwen2.5-VL-7B is fine-tuned using the GRPO algorithm [117] on a set of 67,500 synthetically generated training examples derived from flight trajectories in the Forest environment. The reward function incentivises the model to move closer to the target at each step, produce explicit reasoning, and correctly issue `FOUND` when the target is in view. Training is conducted in offline, step-wise mode using LoRA [49] on four H100 GPUs.

The resulting fine-tuned model is evaluated on both the Forest environment it was trained on (achieving 57.0% on FS-1 Forest) and the held-out City environment (achieving 27.0% on FS-1 City). The transfer to City is notable: the base model scored only 1.5%, providing evidence of transfer beyond the training environment. However, on FS-2, the fine-tuned model scores 0.0%, identical to the base model. The fine-tuning addresses surface-level failures (instruction format adherence, short-horizon spatial manoeuvring) but does not instil the systematic multi-step coverage strategy that FS-2 requires, suggesting that the exploration gap is a deeper limitation of current models rather than an artefact of missing task-specific training.

Additional experimental results are provided in the paper [IV].

4.3.4 Contributions

The publication "FlySearch: Exploring how vision-language models explore" [IV] makes the following contributions to the field:

- It introduces two high-fidelity photorealistic outdoor evaluation environments, a procedurally generated forest and a large semi-procedural city, both built with Unreal Engine 5 and capable of generating an unlimited number of reproducible scenarios with diverse conditions (time of day, object placement, distractor density).

- It defines three standardised benchmark challenge sets (FS-1, FS-Anomaly-1, FS-2) with increasing difficulty, designed to isolate distinct aspects of exploration capability: basic object detection and short-horizon navigation, contextual anomaly identification, and systematic multi-step coverage of an area in which the target is not initially visible.
- It provides an evaluation of ten VLMs across all three challenges, with human baselines, revealing consistent failure modes (hallucination when losing a target, inability to plan systematic coverage, sensitivity to long contexts) shared across proprietary and open-weight models alike.
- It demonstrates through GRPO fine-tuning that surface-level failures (instruction following, spatial grounding) are correctable with synthetic data, but that the deeper inability to form and execute multi-step exploration strategies persists after fine-tuning, identifying this as an open problem for current VLMs.

The best evaluated models were able to perform short-horizon navigation, but the gap to human performance widened as the tasks required longer strategies. The systematic-search failures persisted even after fine-tuning. This confirms Hypothesis H4 within the evaluated settings. FlySearch makes the problem of open-world exploration precise. By characterising where current models fail and under what conditions the gap to human performance is largest, it establishes a concrete set of open problems and a reproducible benchmark against which future progress can be measured.

I was the first and primary author of this paper. I contributed to the conception of the idea, implemented the entire Unreal Engine component of the benchmark, and prepared the benchmark evaluation environments together with their Unreal Engine procedural content generation component. I conducted roughly half of the experiments presented in the paper and contributed significantly to writing the manuscript. I also presented the research at the Conference on Neural Information Processing Systems (NeurIPS) 2025 in San Diego, USA. The code, data and further visualisations are available on the project website at `https://flysearch.gmum.net/`.

Chapter 5

Other works & achievements

This chapter opens with a brief overview of other papers I co-authored during my PhD studies. These works either fall outside the thematic scope of this thesis or are part of ongoing research and are therefore omitted from the main text. Nevertheless, as they are relevant to my broader research area, they are included here. The chapter then concludes with a summary of my other achievements, including complementary research activities and outreach initiatives aimed at advancing deep learning.

5.1 Publications

- Turhan Can Kargin, Wojciech Jasiński, Adam Pardyl, Bartosz Zieliński, and Marcin Przewięźlikowski. SpaRRTa: A Synthetic Benchmark for Evaluating Spatial Intelligence in Visual Foundation Models. *International Journal of Computer Vision*, 2026. URL `https://arxiv.org/abs/2601.11729`. Accepted for publication

  Summary:

  > Visual foundation models recognise semantic content in images, but their ability to represent spatial relations between objects remains less well understood. The study introduces the Spatial Relation Recognition Task (SpaRRTa), a benchmark that evaluates this ability independently of semantic classification and precise geometric prediction. A controllable synthetic data pipeline generates photorealistic scenes with varied object arrangements and spatial annotations. The evaluation compares visual foundation models using different probing methods to assess how spatial information can be extracted from their representations. Results show that spatial relations are primarily encoded in patch-level features and can be obscured by global pooling, causing standard linear evaluation to underestimate spatial awareness. Relations defined relative to another object remain challenging across model families. These findings establish SpaRRTa as a diagnostic benchmark for distinguishing limitations of visual representations from limitations of the methods used to read out spatial information.

- Agnieszka Sroka-Oleksiak, Adam Pardyl, Dawid Rymarczyk, Aldona Olechowska-Jarząb, Katarzyna Biegun-Drożdż, Dorota Ochońska, Michał Wronka, Adriana Borowa, Tomasz Gosiewski, Miłosz Adamczyk, et al. AI-Driven Rapid Identification of Bacterial and Fungal Pathogens in Blood Smears of Septic Patients. *Computers in Biology and Medicine*, 199:111328, 2025. `https://doi.org/10.1016/j.compbiomed.2025.111328`
  Summary:

  > Identifying pathogens from blood samples is an important step in sepsis diagnosis, but conventional microbiological procedures can be time-consuming. The study develops a deep learning pipeline for species-level identification from microscopic images of Gram-stained smears of positive blood samples. It combines Cellpose 3 for microbial cell segmentation with attention-based deep multiple instance learning for classification, allowing information from multiple cells to contribute to each prediction. The evaluation uses 16,637 images covering 14 bacterial and three yeast-like fungal species. The model achieves classification accuracies of 77.15% for bacteria and 71.39% for fungi. Errors are associated with morphological similarities between species and variation within individual species. The results support the potential of microscopy-based microbial identification as an aid to diagnosis, while indicating that further optimisation and more diverse training data are needed to improve classification reliability.

- Adam Pardyl, Dawid Rymarczyk, Joanna Jaworek-Korjakowska, Dariusz Kucharski, Andrzej Brodzicki, Julia Lasek, Zofia Schneider, Iwona Kucybała, Andrzej Urbanik, Rafał Obuchowicz, et al. CompLung: Comprehensive Computer-Aided Diagnosis of Lung Cancer. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 1835–1842, 2023. `https://doi.org/10.3233/FAIA230471`
  Summary:

  > Computer-aided analysis of lung computed tomography (CT) scans often addresses individual processing stages, leaving their integration into patient-level assessment unresolved. The study introduces CompLung, a pipeline combining lung segmentation, suspicious-region detection, patch representation learning, and patient-level classification. It uses multiple instance learning to aggregate information from candidate regions into a prediction for the complete scan. The evaluation uses LIDC-IDRI, extended with radiologist-produced lung segmentation masks made available for further research. For patient-level classification using a malignancy-score threshold greater than three, CompLung achieves an area under the receiver operating characteristic curve of 0.90, compared with 0.83 for the strongest reported baselines. Qualitative analyses also examine the regions contributing to predictions. The study demonstrates the benefit of integrating the processing stages for prediction of radiologist-derived labels and provides segmentation annotations to support subsequent work on automated CT analysis.

- Łukasz Struski, Adam Pardyl, Jacek Tabor, and Bartosz Zieliński. ProPML: Probability Partial Multi-label Learning. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–8, 2023. `https://doi.org/10.1109/DSAA60987.2023.10302620`
  Summary:

  > Partial multi-label learning addresses training data in which each instance has a candidate label set containing both correct and incorrect labels. Existing approaches often require a separate procedure to identify the relevant labels, increasing training complexity. The study introduces ProPML, a probabilistic loss that adapts binary cross-entropy to this setting. It encourages predictions within the candidate set while penalising predictions outside it, and can be used with deep networks through a change to the training objective. The evaluation covers datasets with naturally ambiguous labels and artificially corrupted labels, including image classification on VOC2007 and COCO2014, across several noise levels. ProPML compares favourably with existing methods, with particularly strong results when candidate sets contain more incorrect labels. These findings support a simple loss-based approach to learning from ambiguous annotations without a separate label-disambiguation procedure.

- Adam Pardyl, Dawid Rymarczyk, Zbisław Tabor, and Bartosz Zieliński. Automating Patient-Level Lung Cancer Diagnosis in Different Data Regimes. In *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, pages 13–24, 2022. `https://doi.org/10.1007/978-981-99-1648-1_2`
  Summary:

  > Patient-level diagnostics from lung computed tomography (CT) scans requires models that can operate with different amounts of annotation. The study introduces MilLung and AutoLung and compares them with the existing DeepLung baseline to examine the effect of supervision on automated assessment. DeepLung uses nodule locations and malignancy annotations, MilLung combines nodule-level supervision with multiple instance learning, while AutoLung learns representations with an autoencoder and requires only patient-level malignancy labels. The evaluation uses LIDC-IDRI under two malignancy thresholds. At a threshold greater than three, the best MilLung variant matches DeepLung's mean area under the receiver operating characteristic curve of 0.83, while AutoLung reaches 0.80. AutoLung performs less well at the lower threshold, showing that the effect of reduced supervision depends on the task. The study clarifies the trade-off between annotation requirements and predictive performance when selecting a patient-level screening approach.

5.2 Patents

- **European Patent application EP24461637.1** "A computer implemented method for identifying a microorganism in a blood and a data processing system therefor". European Patent Office, 2024, patent pending

5.3 Ongoing research

- **Reinforcement learning for visual exploration in real-world environments.** Adam Pardyl, Jan Mikołajczyk, Jan Matusiak, Hubert Zientata, Dominik Gaweł, Ignacy Alwasiak, Bartosz Zieliński, Marek Cygan, Maciej Wołczyk

  Summary:

  > Training agents for visual exploration requires diverse navigation experience and a way to transfer learned behaviour to physical environments. This project extends the FlySearch [IV] benchmark towards reinforcement learning for vision-language model-based agents operating in the real world. The planned approach uses the GRPO algorithm [117] to fine-tune models with 4–11 billion parameters on FlySearch-RealTime, a simplified version of the benchmark, and Gaussian splat scenes reconstructed from real-world flight data. Procedurally generated tasks will provide varied navigation experience, while the reconstructed scenes are intended to reduce the visual gap between simulation and deployment. Evaluation will use the original FlySearch benchmark and a physical UAV in a real-world setting. The intended contribution is a training framework for learning exploration policies that transfer from simulated tasks to physical navigation.

- **Self-supervised concept discovery.** Adam Pardyl, Siddhartha Gairola, Sukrut Rao, Bernt Schiele, Bartosz Zieliński, Dawid Rymarczyk

  Summary:

  > Self-supervised visual representations support a range of downstream tasks, but their features can be difficult to interpret. This project aims to learn discrete, spatially grounded concepts corresponding to visual features such as object parts or texture patterns without manual concept annotations. Inspired by PDiscoFormer [5] and CFM [134], the proposed approach extends the DINOv2 family [88] with a spatial concept learning pathway and a concept attribute quantisation stack to represent both general object characteristics and their variations. The resulting decoded concept embedding is intended to replace the CLS token of a vision transformer in downstream tasks. The project will examine whether the learned concepts capture coherent visual features while retaining useful information for downstream tasks. Its intended contribution is an interpretable vision foundation backbone that connects image-level representations to discrete concepts and their spatial locations.

5.4 International research internships

- **Max Planck Institute for Informatics**, June–September 2026, Computer Vision Group, supervised by Prof. Dr. Bernt Schiele

5.5 Scientific grants

5.5.1 As a principal investigator

- **National Science Centre Preludium** "Where to look next – guiding active visual exploration with internal model uncertainty", 2024–2025, grant number: 2023/49/N/ST6/02465
- **Foundation for Polish Science START scholarship**, 2026–2027, grant number: 59.2026

5.5.2 As a participant / co-investigator

- **National Science Centre Sonata Bis** "Sustainable computer vision for autonomous machines", 2024–2027, grant number: 2023/50/E/ST6/00469 (principal investigator: Bartosz Zieliński)
- **Foundation for Polish Science Proof of Concept** "Intelligent Fight Against Sepsis - Innovative Use of Artificial Intelligence in Rapid Microbiological Diagnosis of Sepsis", 2024–2025, grant number: FENG.02.07-IP.05-0068/23 (principal investigator: Monika Brzychczy-Włoch)
- **European Commission Horizon Europe** "European Lighthouse of AI for Sustainability (ELIAS)", 2024–2027, grant number: 101120237 (principal investigator: Massimiliano Pontil)
- **Foundation for Polish Science Team-Net** "Bio-inspired artificial neural networks", 2022–2023, grant number: POIR.04.04.00-00-14DE/18-00 (principal investigator: Jacek Tabor)
- **National Centre for Research and Development Fast Track** "X-rAI: Diagnostic viewer for computer-assisted radiology using Artificial Intelligence", 2021–2023, grant number: POIR.01.01.01-00-1666/20 (principal investigator: Zbisław Tabor)

5.6 Other activities

5.6.1 Acting as a reviewer

- Annual Conference on Neural Information Processing Systems (NeurIPS) 2026
- International Journal of Computer Vision (IJCV) 2026
- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2026
- IEEE Robotics and Automation Letters (RA-L) 2026
- CVPR 2025 Workshop on Computer Vision for Drug Discovery (CVDD)
- IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2025
- ACM International Conference on Knowledge Discovery and Data Mining (KDD) 2024
- IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2024

5.6.2 Doctoral schools and scientific events

As an organiser

- Machine Learning Summer School on Reliability & Safety (MLSS$^{R\&S}$) 2026, Kraków, Poland
- ELLIS Doctoral Symposium (EDS) on Robust AI 2025, Warsaw, Poland
- Machine Learning Summer School on Drug and Materials Discovery (MLSS^{D}) 2025, Kraków, Poland
- Machine Learning Summer School on Applications in Science (MLSS^{S}) 2023, Kraków, Poland

As a presenter

- ML in PL Conference 2025, Warsaw, Poland, "FlySearch: Exploring how vision-language models explore"
- ML in PL Conference 2024, Warsaw, Poland, "AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale"
- SFI IT Academic Festival 2024, Kraków, Poland, "AI for autonomous agents"

As a participant

- Mediterranean Machine Learning (M2L) Summer School 2025, Split, Croatia
- International Computer Vision Summer School (ICVSS) 2025, Scicli, Italy
- ELIAS-ELLIS-VISMAC Winter School 2025, Brunico, Italy
- ELLIS Doctoral Symposium (EDS) on AI & Sustainability 2024, Paris, France
- Mediterranean Machine Learning (M2L) Summer School 2023, Thessaloniki, Greece

5.6.3 Acting as a PhD Student Representative

- Member of the Audit Committee of the PhD Student Association of the Jagiellonian University, 2025–2026
- Member of the PhD Student Council of the Doctoral School of Exact and Natural Sciences of the Jagiellonian University, 2024–2026
- PhD Student Representative to the Standing Rector's Committee on Statutory and Legal Affairs, 2022–2026
- PhD Student Representative to the Council of the Faculty of Mathematics and Computer Science of the Jagiellonian University, 2025–2026

Chapter 6

Synthesis and future directions

This dissertation investigated how visual agents can select informative observations under limited sensing budgets and where their ability to explore breaks down as search problems become more complex. The four publications on which the thesis builds confirm the hypotheses introduced in the first chapter (Section 1.3) within the models, datasets, tasks, and environments evaluated.

Hypothesis H1 concerns the use of internal uncertainty to decide where to look next. AME [I] shows that attention-map entropy provides an effective glimpse-selection heuristic without a separately trained decision network. Improvements over baseline methods in reconstruction, classification, and segmentation, supported by ablations against random and fixed sampling, demonstrate that a model trained for its visual task can also guide observation selection. This first step operates within a fixed grid of candidate glimpses.

Hypothesis H2 addresses the restriction imposed by that grid. Beyond Grids [II] demonstrates that a vision transformer can be made resilient to changes in patch position, scale, and coverage. ElasticViT retains useful predictive performance under these perturbations and supports adaptive sampling, achieving high accuracy even at low patch counts. It provides the architectural support for processing observations whose geometry changes during exploration, leaving their selection to an observation policy.

Hypothesis H3 asks whether an agent can learn to exploit this flexibility by selecting both glimpse position and scale. AdaGlimpse [III] learns a coarse-to-fine strategy in a continuous action space, first acquiring a broad view and then concentrating on informative details. Its substantial gains in pixel efficiency over baseline methods across reconstruction, classification, and segmentation demonstrate the value of adaptive observation selection.

Hypothesis H4 concerns exploration through large, open, three-dimensional environments. FlySearch [IV] shows that the evaluated vision-language models can perform short-horizon navigation but struggle with systematic search when the target is initially unseen. Their gap to human performance widens as tasks require longer strategies. The tested fine-tuning improves short-horizon navigation, including transfer across environments, but leaves the fine-tuned model struggling with systematic search over a long horizon. The benchmark makes this limitation explicit and provides a reproducible measure of progress.

Together, the first three contributions connect the signal that guides observation selection, the architecture that processes flexible observations, and the policy that chooses them. FlySearch identifies the additional demand of organising observations and movements into sustained, goal-driven search, presenting a direct challenge for further work.

Parts of this research were supported by an NCN Preludium grant, for which I was the principal investigator. My work was also recognised with the FNP START scholarship.

Future research should investigate how agents can retain information about visited regions, identify unexplored areas, and revise their search as evidence arrives. The fine-tuning results motivate investigating whether training on complete search trajectories can develop capabilities that training on individual navigation steps did not provide. Evaluation should measure systematic-search success and generalisation to unseen environments.

A second direction is transfer to physical environments. My ongoing follow-up to FlySearch plans to combine procedurally generated navigation tasks with scenes reconstructed from real flight data, followed by evaluation on the benchmark and a physical UAV. This would test whether improvements in simulated search survive changes in visual appearance and the constraints of real sensing and movement.

Finally, a third direction is explaining the visual evidence behind an agent's decisions. My collaboration with the Max Planck Institute for Informatics explores interpretable vision foundation models that learn spatially grounded concepts through self-supervision. Extending these representations to embodied AI could help explain which observed features guide an agent's actions and make failures easier to diagnose and the model itself more trustworthy and reliable.

The contribution of this thesis is a set of methods for reducing the visual input needed to solve embodied visual tasks, together with a benchmark exposing the limits of current open-world search. The publications included in this thesis advance efficient observation selection and define concrete challenges for the next steps in research.

Acknowledgements

The research presented in this dissertation was carried out with the support of several funding bodies and computing infrastructures. Their contributions are acknowledged below, publication by publication, following the acknowledgements of the original papers.

AME (IJCAI 2023). This research was funded by National Science Centre, Poland (grants no 2020/39/B/ST6/01511, 2022/45/B/ST6/02817, and 2021/41/B/ST6/01370), Foundation for Polish Science (grant no POIR.04.04.00-00-14DE/18-00 carried out within the Team-Net program co-financed by the European Union under the European Regional Development Fund). We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2022/015753. The research was supported by a grant from the Faculty of Mathematics and Computer Science under the Strategic Programme Excellence Initiative at Jagiellonian University.

Beyond Grids (WACV 2025). This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237, and by National Science Centre, Poland (grant no. 2023/49/N/ST6/02465, 2022/47/B/ST6/03397, 2022/45/B/ST6/02817, and 2023/50/E/ST6/00469). Some experiments were performed on servers purchased with funds from a grant from the Priority Research Area (Artificial Intelligence Computing Center Core Facility) under the Strategic Programme Excellence Initiative at Jagiellonian University. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017483.

AdaGlimpse (ECCV 2024). This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. This research was funded by National Science Centre, Poland (grant no. 2023/49/N/ST6/02465, 2022/47/B/ST6/03397, 2022/45/B/ST6/02817, and 2023/50/E/ST6/00469). The research was supported by a grant from the Faculty of Mathematics and Computer Science under the Strategic Programme Excellence Initiative at Jagiellonian University. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016558.

FlySearch (NeurIPS 2025). This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. This research was funded by National Science Centre, Poland (grant

no. 2023/50/E/ST6/00469 and Sonata Bis grant no 2024/54/E/ST6/00388). The research was supported by a grant from the Faculty of Mathematics and Computer Science under the Strategic Programme Excellence Initiative at Jagiellonian University. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017483. Some experiments were performed on servers purchased with funds from the Priority Research Area (Artificial Intelligence Computing Center Core Facility) under the Strategic Programme Excellence Initiative at Jagiellonian University.

Other acknowledgements. My work was recognised with the START scholarship of the Foundation for Polish Science (grant no. 59.2026). For the preparation of this dissertation and the ongoing research described in Chapter 5, I gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2025/018688.

Bibliography

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. Technical Report MSR-TR-2024-12, Microsoft, August 2024. URL https://www.microsoft.com/en-us/research/publication/phi-3-technical-report-a-highly-capable-language-model-locally-on-your-phone/.

[2] Samira Abnar and Willem Zuidema. Quantifying Attention Flow in Transformers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4190–4197, 2020. https://doi.org/10.18653/v1/2020.acl-main.385.

[3] Elie Aljalbout, Jiaxu Xing, Angel Romero, Iretiayo Akinola, Caelan Reed Garrett, Eric Heiden, Abhishek Gupta, Tucker Hermans, Yashraj Narang, Dieter Fox, Davide Scaramuzza, and Fabio Ramos. The Reality Gap in Robotics: Challenges, Solutions, and Best Practices. *Annual Review of Control, Robotics, and Autonomous Systems*, 9(1):403–432, 2026. https://doi.org/10.1146/annurev-control-031924-100130.

[4] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, 1988. https://doi.org/10.1007/bf00133571.

[5] Ananthu Aniraj, Cassio F Dantas, Dino Ienco, and Diego Marcos. PDiscoFormer: Relaxing Part Discovery Constraints with Vision Transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 256–272, 2024. https://doi.org/10.1007/978-3-031-73013-9_15.

[6] Anthropic. Introducing Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet, Jun 2024.

[7] Isaac Asimov. *I, Robot*. Gnome Press, New York, 1950.

[8] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple Object Recognition with Visual Attention. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.7755.

[9] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL Technical Report. arXiv preprint arXiv:2511.21631, 2025. URL https://doi.org/10.48550/arXiv.2511.21631.

[10] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL Technical Report. arXiv preprint arXiv:2502.13923, 2025. URL https://doi.org/10.48550/arXiv.2502.13923.

[11] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988. https://doi.org/10.1109/5.5968.

[12] Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018. https://doi.org/10.1007/s10514-017-9615-3.

[13] Dana H. Ballard. Animate vision. *Artificial Intelligence*, 48(1):57–86, 1991. https://doi.org/10.1016/0004-3702(91)90080-4.

[14] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects. arXiv preprint arXiv:2006.13171, 2020. URL https://doi.org/10.48550/arXiv.2006.13171.

[15] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. FlexiViT: One Model for All Patch Sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14496–14506, 2023. https://doi.org/10.1109/cvpr52729.2023.01393.

[16] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. https://doi.org/10.15607/rss.2025.xxi.010.

[17] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. https://doi.org/10.1109/tro.2016.2624754.

[18] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134(1–2):57–83, 2002. https://doi.org/10.1016/s0004-3702(01)00129-1.

[19] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. https://doi.org/10.1109/tpami.1986.4767851.

[20] Karel Čapek. *R.U.R. (Rossum's Universal Robots)*. Aventinum, Prague, 1920. English translation, first performed 1921.

[21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 9630–9640, 2021. https://doi.org/10.1109/iccv48922.2021.00951.

[22] Yuning Chai. Patchwork: A Patch-Wise Attention Network for Efficient Object Detection and Segmentation in Video Streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3414–3423, 2019. https://doi.org/10.1109/iccv.2019.00351.

[23] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are We on the Right Way for Evaluating Large Vision-Language Models? In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 27056–27087, 2024. https://doi.org/10.52202/079017-0850.

[24] Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep Reinforcement Learning from Human Preferences. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 4299–4307, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

[25] Christine A Curcio, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson. Human photoreceptor topography. *Journal of Comparative Neurology*, 292(4):497–523, 1990. https://doi.org/10.1002/cne.902920402.

[26] Martin Davis and Hilary Putnam. A Computing Procedure for Quantification Theory. *Journal of the ACM*, 7(3):201–215, 1960. https://doi.org/10.1145/321033.321034.

[27] Leonardo De Moura and Nikolaj Bjørner. Z3: An Efficient SMT Solver. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 337–340, 2008. https://doi.org/10.1007/978-3-540-78800-3_24.

[28] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n' Pack: NaViT, a Vision Transformer for any Aspect Ratio and Resolution. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 2252–2274, 2023. https://doi.org/10.52202/075280-0106.

[29] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. https://doi.org/10.1109/cvpr.2009.5206848.

[30] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[31] Hubert L. Dreyfus. *What Computers Still Can't Do: A Critique of Artificial Reason*. MIT Press, Cambridge, MA, 1992.

[32] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A Survey of Embodied AI: From Simulators to Research Tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022. https://doi.org/10.1109/tetci.2022.3141105.

[33] Gamaleldin Elsayed, Simon Kornblith, and Quoc V. Le. Saccader: Improving Accuracy of Hard Attention Models for Vision. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 700–712, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8dd48d6a2e2cad213179a3992c0be53c-Abstract.html.

[34] Epic Games. City Sample. An overview of the features used to create the City Sample learning example. https://dev.epicgames.com/documentation/en-us/unreal-engine/city-sample-project-unreal-engine-demonstration, Nov 2024.

[35] Epic Games. Electric Dreams environment in Unreal Engine – Unreal Engine 5.5 documentation. https://dev.epicgames.com/documentation/en-us/unreal-engine/electric-dreams-environment-in-unreal-engine, Nov 2024.

[36] Epic Games. Unreal Engine 5.5. https://unrealengine.com, Nov 2024.

[37] Brett R. Fajen and William H. Warren. Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2): 343–362, 2003. https://doi.org/10.1037/0096-1523.29.2.343.

[38] Daniel J Felleman and David C Van Essen. Distributed Hierarchical Processing in the Primate Cerebral Cortex. *Cerebral Cortex*, 1(1):1–47, 1991. https://doi.org/10.1093/cercor/1.1.1.

[39] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. CoWs on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23171–23181, 2023. https://doi.org/10.1109/CVPR52729.2023.02219.

[40] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, et al. Gemini: A Family of Highly Capable Multimodal Models. arXiv preprint arXiv:2312.11805, 2023. URL https://doi.org/10.48550/arXiv.2312.11805.

[41] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16081–16091, 2022. https://doi.org/10.1109/cvpr52688.2022.01563.

[42] Google. Introducing Gemini 2.0: Our new AI model for the agentic era. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/, Dec 2024.

[43] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6325–6334, 2017. https://doi.org/10.1109/CVPR.2017.670.

[44] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Eric Wang. Vision-and-Language Navigation: A Survey of Tasks, Methods, and Future Directions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, pages 7606–7623, 2022. https://doi.org/10.18653/v1/2022.acl-long.524.

[45] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1861–1870, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

[46] Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in Cognitive Sciences*, 9(4):188–194, 2005. https://doi.org/10.1016/j.tics.2005.02.009.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. https://doi.org/10.1109/cvpr.2016.90.

[48] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022. https://doi.org/10.1109/cvpr52688.2022.01553.

[49] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[50] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based Deep Multiple Instance Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2127–2136, 2018. URL https://proceedings.mlr.press/v80/ilse18a.html.

[51] Dinesh Jayaraman and Kristen Grauman. Learning to Look Around: Intelligently Exploring Unseen Environments for Unknown Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1238–1247, 2018. https://doi.org/10.1109/CVPR.2018.00135.

[52] Abhishek Jha, Soroush Seifi, and Tinne Tuytelaars. SimGlim: Simplifying glimpse based active visual reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 269–278, 2023. https://doi.org/10.1109/wacv56688.2023.00035.

[53] Turhan Can Kargin, Wojciech Jasiński, Adam Pardyl, Bartosz Zieliński, and Marcin Przewięźlikowski. SpaRRTa: A Synthetic Benchmark for Evaluating Spatial Intelligence in Visual Foundation Models. *International Journal of Computer Vision*, 2026. URL https://arxiv.org/abs/2601.11729. Accepted for publication.

[54] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An Interactive 3D Environment for Visual AI. arXiv preprint arXiv:1712.05474, 2017. URL https://doi.org/10.48550/arXiv.1712.05474.

[55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 1106–1114, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[56] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023. https://doi.org/10.1145/3600006.3613165.

[57] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing Frontiers in Open Language Model Post-Training. In *Proceedings of the Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

[58] Michael F. Land and Dan-Eric Nilsson. *Animal Eyes*. Oxford University Press, Oxford, 2nd edition, 2012.

[59] Fritz Lang. Metropolis. Motion picture, 1927.

[60] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. https://doi.org/10.1109/5.726791.

[61] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. https://doi.org/10.1038/nature14539.

[62] Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. CityNav: A Large-Scale Dataset for Real-World Aerial Navigation. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 5912–5922, 2025. https://doi.org/10.1109/iccv51701.2025.00559.

[63] Xuanyu Lei, Zonghan Yang, Xinrui Chen, Peng Li, and Yang Liu. Scaffolding Coordinates to Promote Vision-Language Coordination in Large Multi-Modal Models. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pages 2886–2903, 2025. URL https://aclanthology.org/2025.coling-main.195/.

[64] Stanisław Lem. *Cyberiada*. Wydawnictwo Literackie, Kraków, 1965.

[65] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy Visual Task Transfer. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=zKv8qULV6n.

[66] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient Self-supervised Vision Transformers for Representation Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=fVu3o-YUGQK.

[67] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. LLaVA-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=oSQiao9GqB.

[68] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. https://doi.org/10.1007/978-3-319-10602-1_48.

[69] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 34892–34916, 2023. https://doi.org/10.52202/075280-1516.

[70] Jihao Liu, Boxiao Liu, Hang Zhou, Hongsheng Li, and Yu Liu. TokenMix: Rethinking Image Mixing for Data Augmentation in Vision Transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 455–471, 2022. https://doi.org/10.1007/978-3-031-19809-0_26.

[71] Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. AerialVLN: Vision-and-Language Navigation for UAVs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15338–15348, 2023. https://doi.org/10.1109/ICCV51070.2023.01411.

[72] Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Xixuan Song, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, et al. VisualAgentBench: Towards Large Multimodal Models as Visual

Foundation Agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=2snKOc7TVp.

[73] Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning Cyber Space With Physical World: A Comprehensive Survey on Embodied AI. *IEEE/ASME Transactions on Mechatronics*, 30(6):7253–7274, 2025. https://doi.org/10.1109/tmech.2025.3574943.

[74] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is Your Multi-modal Model an All-Around Player? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 216–233, 2024. https://doi.org/10.1007/978-3-031-72658-3_13.

[75] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. https://doi.org/10.1109/iccv48922.2021.00986.

[76] George Lucas. Star Wars. Motion picture, 1977.

[77] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. ZSON: Zero-Shot Object-Goal Navigation Using Multimodal Goal Embeddings. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 32340–32352, 2022. https://doi.org/10.52202/068431-2343.

[78] Adrienne Mayor. *Gods and Robots: Myths, Machines, and Ancient Dreams of Technology*. Princeton University Press, Princeton, NJ, 2018.

[79] John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. Technical report, Dartmouth College, 1955.

[80] Mistral. Pixtral Large. https://mistral.ai/news/pixtral-large/, Nov 2024.

[81] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent Models of Visual Attention. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 2204–2212, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/3e456b31302cf8210edd4029292a40ad-Abstract.html.

[82] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. https://doi.org/10.1038/nature14236.

[83] Hans Moravec. *Mind Children: The Future of Robot and Human Intelligence*. Harvard University Press, Cambridge, MA, 1988.

[84] Hans Peter Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Stanford University, 1980.

[85] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. PIVOT: Iterative Visual Prompting Elicits Actionable Knowledge for VLMs. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 37321–37341, 2024. URL https://proceedings.mlr.press/v235/nasiriany24a.html.

[86] Nils J. Nilsson. Shakey the Robot. Technical Note 323, SRI International, 1984.

[87] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. GPT-4 Technical Report. arXiv preprint arXiv:2303.08774, 2023. URL https://doi.org/10.48550/arXiv.2303.08774.

[88] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

[89] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training Language Models to Follow Instructions with Human Feedback. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 27730–27744, 2022. https://doi.org/10.52202/068431-2011.

[90] Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. BALROG: Benchmarking Agentic LLM and VLM Reasoning On Games. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=fp6t3F669F.

[91] Athanasios Papadopoulos, Pawel Korus, and Nasir Memon. Hard-Attention for Scalable Image Classification. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 14694–14707, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7b7916dd2de56297aa29cccb2bbf48d4-Abstract.html.

[92] Adam Pardyl, Dawid Rymarczyk, Zbisław Tabor, and Bartosz Zieliński. Automating Patient-Level Lung Cancer Diagnosis in Different Data Regimes. In *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, pages 13–24, 2022. https://doi.org/10.1007/978-981-99-1648-1_2.

[93] Adam Pardyl, Dawid Rymarczyk, Joanna Jaworek-Korjakowska, Dariusz Kucharski, Andrzej Brodzicki, Julia Lasek, Zofia Schneider, Iwona Kucybała, Andrzej Urbanik, Rafał Obuchowicz, et al. CompLung: Comprehensive Computer-Aided Diagnosis of Lung Cancer. In *Proceedings*

of the European Conference on Artificial Intelligence (ECAI), pages 1835–1842, 2023. https://doi.org/10.3233/FAIA230471.

[94] Adam Pardyl, Grzegorz Rypeść, Grzegorz Kurzejamski, Bartosz Zieliński, and Tomasz Trzciński. Active Visual Exploration Based on Attention-Map Entropy. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1303–1311, 2023. https://doi.org/10.24963/ijcai.2023/145.

[95] Adam Pardyl, Michał Wronka, Maciej Wołczyk, Kamil Adamczewski, Tomasz Trzciński, and Bartosz Zieliński. AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 112–129, 2024. https://doi.org/10.1007/978-3-031-72664-4_7.

[96] Adam Pardyl, Grzegorz Kurzejamski, Jan Olszewski, Tomasz Trzciński, and Bartosz Zieliński. Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8536–8545, 2025. https://doi.org/10.1109/WACV61041.2025.00827.

[97] Adam Pardyl, Dominik Matuszek, Mateusz Przebieracz, Marek Cygan, Bartosz Zieliński, and Maciej Wołczyk. FlySearch: Exploring how vision-language models explore. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 185979–186011, 2025. https://doi.org/10.52202/085713-5594.

[98] Andrew Parker. *In the Blink of an Eye: How Vision Sparked the Big Bang of Evolution*. Basic Books, New York, 2003.

[99] Yanyuan Qiao, Zheng Yu, Zijia Zhao, Sihan Chen, Mingzhen Sun, Longteng Guo, Qi Wu, and Jing Liu. VL-Mamba: Exploring State Space Models for Multimodal Learning. In *Proceedings of the 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, pages 102–113, 2024. URL https://proceedings.mlr.press/v262/qiao24a.html.

[100] Qwen Team. Qwen3.5-Omni Technical Report. arXiv preprint arXiv:2604.15804, 2026. URL https://doi.org/10.48550/arXiv.2604.15804.

[101] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[102] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 53728–53741, 2023. https://doi.org/10.52202/075280-2338.

[103] Santhosh K. Ramakrishnan and Kristen Grauman. Sidekick Policy Learning for Active Visual Exploration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 424–442, 2018. https://doi.org/10.1007/978-3-030-01258-8_26.

[104] Santhosh K. Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. An Exploration of Embodied Visual Exploration. *International Journal of Computer Vision*, 129(5):1616–1649, 2021. https://doi.org/10.1007/s11263-021-01437-z.

[105] Samrudhdhi B. Rangrej and James J. Clark. Visual Attention in Imaginative Agents. arXiv preprint arXiv:2104.00177, 2021. URL https://doi.org/10.48550/arXiv.2104.00177.

[106] Samrudhdhi B Rangrej, Chetan L Srinidhi, and James J Clark. Consistency driven Sequential Transformers Attention Model for Partially Observable Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2508–2517, 2022. https://doi.org/10.1109/cvpr52688.2022.00255.

[107] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to Zoom: A Saliency-Based Sampling Layer for Neural Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. https://doi.org/10.1007/978-3-030-01240-3_4.

[108] Jessica Riskin. The Defecating Duck, or, the Ambiguous Origins of Artificial Life. *Critical Inquiry*, 29(4):599–633, 2003. https://doi.org/10.1086/377722.

[109] Lawrence G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[110] Giulio Sandini and Giorgio Metta. Retina-Like Sensors: Motivations, Technology and Applications. In *Sensors and Sensing in Biology and Engineering*, pages 251–262. Springer, 2003. https://doi.org/10.1007/978-3-7091-6025-1_18.

[111] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9338–9346, 2019. https://doi.org/10.1109/iccv.2019.00943.

[112] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347, 2017. URL https://doi.org/10.48550/arXiv.1707.06347.

[113] Soroush Seifi and Tinne Tuytelaars. Where to Look Next: Unsupervised Active Visual Exploration on 360° Input. arXiv preprint arXiv:1909.10304, 2019. URL https://doi.org/10.48550/arXiv.1909.10304.

[114] Soroush Seifi and Tinne Tuytelaars. Attend and Segment: Attention Guided Active Semantic Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2020. https://doi.org/10.1007/978-3-030-58595-2_19.

[115] Soroush Seifi, Abhishek Jha, and Tinne Tuytelaars. Glimpse-Attend-and-Explore: Self-Attention for Active Visual Exploration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16117–16126, 2021. https://doi.org/10.1109/ICCV48922.2021.01583.

[116] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*, pages 621–635, 2018. https://doi.org/10.1007/978-3-319-67361-5_40.

[117] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv preprint arXiv:2402.03300, 2024. URL https://doi.org/10.48550/arXiv.2402.03300.

[118] Agnieszka Sroka-Oleksiak, Adam Pardyl, Dawid Rymarczyk, Aldona Olechowska-Jarząb, Katarzyna Biegun-Drożdż, Dorota Ochońska, Michał Wronka, Adriana Borowa, Tomasz Gosiewski, Miłosz Adamczyk, et al. AI-Driven Rapid Identification of Bacterial and Fungal Pathogens in Blood Smears of Septic Patients. *Computers in Biology and Medicine*, 199:111328, 2025. https://doi.org/10.1016/j.compbiomed.2025.111328.

[119] Łukasz Struski, Adam Pardyl, Jacek Tabor, and Bartosz Zieliński. ProPML: Probability Partial Multi-label Learning. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–8, 2023. https://doi.org/10.1109/DSAA60987.2023.10302620.

[120] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[121] Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 516–533, 2022. https://doi.org/10.1007/978-3-031-20053-3_30.

[122] Alan M. Turing. Computing Machinery and Intelligence. *Mind*, 59(236):433–460, 1950. https://doi.org/10.1093/mind/lix.236.433.

[123] Burak Uzkent and Stefano Ermon. Learning When and Where to Zoom With Deep Reinforcement Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12342–12351, 2020. https://doi.org/10.1109/cvpr42600.2020.01236.

[124] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[125] Adelheid Voskuhl. *Androids in the Enlightenment: Mechanics, Artisans, and Cultures of the Self*. University of Chicago Press, Chicago, 2013.

[126] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv preprint arXiv:2409.12191, 2024. URL https://doi.org/10.48550/arXiv.2409.12191.

[127] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the Reasoning Ability of Multimodal Large Language Models via Mixed Preference Optimization. arXiv preprint arXiv:2411.10442, 2024. URL https://doi.org/10.48550/arXiv.2411.10442.

[128] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 548–558, 2021. https://doi.org/10.1109/iccv48922.2021.00061.

[129] Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang. Glance and Focus: a Dynamic Approach to Reducing Spatial Redundancy in Image Classification. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 2432–2444, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/1963bd5135521d623f6c29e6b1174975-Abstract.html.

[130] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-Of-Thought Prompting Elicits Reasoning in Large Language Models. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 24824–24837, 2022. https://doi.org/10.52202/068431-1800.

[131] Patrick Wenzel, Rui Wang, Nan Yang, Qing Cheng, Qadeer Khan, Lukas von Stumberg, Niclas Zeller, and Daniel Cremers. 4Seasons: A Cross-Season Dataset for Multi-Weather SLAM in Autonomous Driving. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, pages 404–417, 2020. https://doi.org/10.1007/978-3-030-71278-5_29.

[132] Fred M. Wilcox. Forbidden Planet. Motion picture, 1956.

[133] Ronald J Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8(3–4):229–256, 1992. https://doi.org/10.1023/a:1022672621406.

[134] Kai Wittenmayer, Sukrut Rao, Amin Parchami-Araghi, Bernt Schiele, and Jonas Fischer. CFM: Language-aligned Concept Foundation Model for Vision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026. URL https://eccv.ecva.net/virtual/2026/spotlight/6025.

[135] Chao-Yuan Wu, Ross Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krahenbuhl. A Multigrid Method for Efficiently Training Video Models. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 150–159, 2020. https://doi.org/10.1109/cvpr42600.2020.00023.

[136] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-World Perception for Embodied Agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9068–9079, 2018. https://doi.org/10.1109/cvpr.2018.00945.

[137] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2695–2702, 2012. https://doi.org/10.1109/cvpr.2012.6247991.

[138] Quanting Xie, Tianyi Zhang, Kedi Xu, Matthew Johnson-Roberson, and Yonatan Bisk. Reasoning about the Unseen for Efficient Outdoor Object Navigation. arXiv preprint arXiv:2309.10103, 2023. URL https://doi.org/10.48550/arXiv.2309.10103.

[139] Alfred L. Yarbus. *Eye Movements and Vision*. Plenum Press, New York, 1967.

[140] Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. LLaDA-V: Large Language Diffusion Models with Visual Instruction Tuning. arXiv preprint arXiv:2505.16933, 2025. URL https://doi.org/10.48550/arXiv.2505.16933.

[141] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. MMMU-Pro: A More Robust Multi-discipline Multimodal Understanding Benchmark. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, pages 15134–15186, 2025. https://doi.org/10.18653/v1/2025.acl-long.736.

[142] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, 2017. https://doi.org/10.1109/cvpr.2017.544.

Appendix A

Full texts of the publications

This chapter contains the publications described in the main thesis text, including written appendices. Any additional supplementary materials (such as interactive content) are provided via links in the corresponding sections of the publications.

[I] Adam Pardyl, Grzegorz Rypeść, Grzegorz Kurzejamski, Bartosz Zieliński, and Tomasz Trzciński. Active Visual Exploration Based on Attention-Map Entropy. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1303–1311, 2023. https://doi.org/10.24963/ijcai.2023/145.

[II] Adam Pardyl, Grzegorz Kurzejamski, Jan Olszewski, Tomasz Trzciński, and Bartosz Zieliński. Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8536–8545, 2025. https://doi.org/10.1109/WACV61041.2025.00827.

[III] Adam Pardyl, Michał Wronka, Maciej Wołczyk, Kamil Adamczewski, Tomasz Trzciński, and Bartosz Zieliński. AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 112–129, 2024. https://doi.org/10.1007/978-3-031-72664-4_7.

[IV] Adam Pardyl, Dominik Matuszek, Mateusz Przebieracz, Marek Cygan, Bartosz Zieliński, and Maciej Wołczyk. FlySearch: Exploring how vision-language models explore. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 185979–186011, 2025. https://doi.org/10.52202/085713-5594.

Active Visual Exploration Based on Attention-Map Entropy

Adam Pardyl[1,2,3] , **Grzegorz Rypeść**[1,4] , **Grzegorz Kurzejamski**[1] ,
Bartosz Zieliński[1,2,6] , **Tomasz Trzciński**[1,2,4,5]
[1]IDEAS NCBR
[2]Jagiellonian University, Faculty of Mathematics and Computer Science
[3]Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4]Warsaw University of Technology
[5]Tooploox
[6]Ardigen
{adam.pardyl, grzegorz.rypesc, grzegorz.kurzejamski, bartosz.zielinski, tomasz.trzcinski}@ideas-ncbr.pl

Abstract

Active visual exploration addresses the issue of limited sensor capabilities in real-world scenarios, where successive observations are actively chosen based on the environment. To tackle this problem, we introduce a new technique called Attention-Map Entropy (AME). It leverages the internal uncertainty of the transformer-based model to determine the most informative observations. In contrast to existing solutions, it does not require additional loss components, which simplifies the training. Through experiments, which also mimic retina-like sensors, we show that such simplified training significantly improves the performance of reconstruction, segmentation and classification on publicly available datasets.

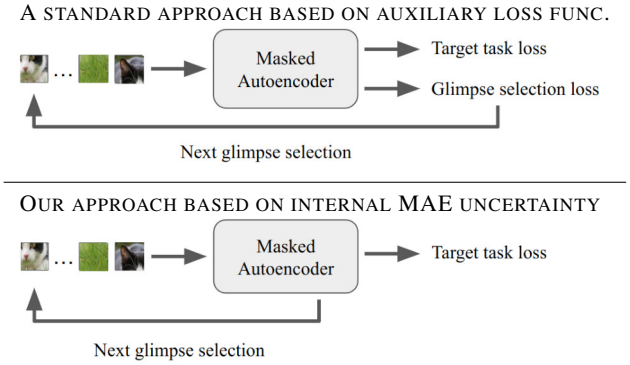


Figure 1: **Attention-Map Entropy (AME):** Our approach chooses the most informative observations by reusing the internal uncertainty coded in the attention maps. In contrast to existing methods, it does not require any auxiliary loss functions dedicated to active exploration. Therefore, the training concentrates on the target task loss, not on an auxiliary loss, which improves overall performance.

1 Introduction

Image-related tasks, such as image reconstruction, segmentation, or scene understanding, usually rely on the assumption of complete and unhindered access to input data [Girdhar *et al.*, 2022]. However, in many real-world applications, this assumption is not valid. For example, in the case of an embodied agent, such as a robot, its sensors have often limited registration capabilities, *e.g.* restricted field of view, while the environment is constantly changing [Wenzel *et al.*, 2021]. Additionally, the high computational cost of processing large amounts of data and the need for continuous movement further complicates the task of acquiring complete information about the environment. To reason about the whole environment, the agent needs to sample new observations in the most efficient way possible.[1]

Previous works in the field of embodied visual exploration have leveraged various sampling strategies to improve results on localization and mapping tasks [Ramakrishnan *et al.*, 2021]. Many of these methods use reinforcement learning [Ramakrishnan and Grauman, 2018] and multiple custom uncertainty measures [Ramakrishnan *et al.*, 2021], but they can be complex and difficult to train. To overcome this issue, [Seifi and Tuytelaars, 2019] introduced a streamlined architecture based on convolutional networks. They also simplified the setup for selecting image patches within a frame, referred to as glimpses, from 360° images in reconstruction tasks. The idea was further extended for segmentation task [Seifi and Tuytelaars, 2020]. In both scenarios, the choice of where to sample next was determined by using separate heads that output probability maps indicating the location of the next glimpse. In [Seifi *et al.*, 2021], they proposed a method of using attention pooling layers in a task-driven architecture to generate probability maps. However, this complex solution consisted of five separate training streams, each with its own optimization goals and a contrastive learning approach.

Furthermore, these existing methods involve separate training for the main task and the sampling decision strategy, which increases computational overhead of the entire system. [Jha *et al.*, 2023] addresses this limitation by using

[1]Supplementary material, including the source code, can be found at: https://github.com/apardyl/AME

a transformer-based auto-encoder with a glimpse selection neural head. Nevertheless, all previous works rely on a dual approach to the active exploration problem. One target function is optimized to improve the quality of the main task, while another one is aimed at training the sampling decision strategy. This introduces unnecessary complexity of the model, increases the number of neural network components and yields more memory-heavy solutions.

In this paper, we address these complexity issues by introducing a transformer-based active vision method that leverages internal model uncertainty for next glimpse selection. Inspired by [Aloimonos *et al.*, 1988], we train our method to explore a scene by gathering partial observations about it sequentially without incorporating any separate sampling decision module. Contrary to the recent solution [Jha *et al.*, 2023], we use attention values trained inherently within the transformer-based decoder to generate decision-making priors, which provides superior performance over both random sampling and state-of-the-art solutions.

Our strategy for glimpse selection is based on the entropy of the transformer's attention maps and thus needs suitable architecture for the main task. Similarly to [Jha *et al.*, 2023], we base our network on masked autoencoder (MAE [He *et al.*, 2022]) to efficiently deal with the problem of partially available input.

We evaluate our approach using common benchmarks and relevant competing methods for reconstruction, classification, and segmentation tasks. Furthermore, we test our solution using both full-resolution patches, and retina-like glimpses [Sandini and Metta, 2003], publishing for the first time the empirical results for the latter data when coupled with transformer-based architectures.

We can summarize our contributions as follows:

- We present a new approach to glimpse selection in active visual exploration that leverages the internal transformer uncertainty stored in attention maps, eliminating the need for additional loss components.
- We validate our method with multiple patch configurations, including retina-like glimpses mimicking retina-like sensors.
- We thoroughly evaluate our solution on various visual tasks, demonstrating its superiority over the existing methods and various selection alternatives.

2 Related Works

Missing data. Our tasks are strictly connected with a missing data problem addressed by various research works. Some of them, like [Śmieja *et al.*, 2018], concentrate on fully connected networks where the missing data can be estimated with input data distribution. Other approaches concentrate on visual inpainting using adversarial and reconstruction loss [Pathak *et al.*, 2016]. [Li *et al.*, 2022] address the inpainting problem by using transformer architecture with a local masking scheme in attention layers, resulting in computation savings. [Naseer *et al.*, 2021] analyze well-known CNNs and claims they reach as low as 20% accuracy in ImageNet evaluation after masking only 20% of the image. Adequate progress in this field was made in Masked Auto-encoders (MAE) [He *et al.*, 2022] where authors use arbitrarily chosen masked parts of the input data to enrich visual features and regularize the training. Our method builds upon the MAE architecture, proven to achieve significant results in partially-masked image reconstruction.

Active visual exploration. Many works discuss simultaneous localization and mapping (SLAM) challenges in the context of active exploration [Ramakrishnan *et al.*, 2021]. [Rangrej and Clark, 2021] use the variational auto-encoder to hallucinate multiple hypotheses of the unseen parts of the image and infer uncertainty scores, which are further used to select the next glimpse parameters. Authors of [Seifi *et al.*, 2021; Seifi and Tuytelaars, 2020; Seifi and Tuytelaars, 2019] presented various solutions for active visual exploration, notably using CNN-based attention maps for glimpse selection and contrastive learning for global feature maps training. [Jha *et al.*, 2023] introduces MAE-based [He *et al.*, 2022] autoencoder with additional glimpse decision neural network to solve image reconstruction tasks. Results of these works have been reported in reconstruction, segmentation, and classification tasks, giving suitable baselines for our research. Our method solves the active visual exploration problem, but in contrast to the existing method, it uses auto-encoder internal, task-driven uncertainty to select successive glimpses.

Glimpse Selection. As opposed to solutions imposed by SLAM tasks, many researchers present glimpse selection problems in other contexts. For example, [Chai, 2019] adopts a deep reinforcement learning approach based on attention mechanisms to choose the best patch from consecutive video frames, yielding significant computational savings. [Jayaraman and Grauman, 2017] use uncertainty measures with reinforcement learning to estimate the following observations' parameters for a single 3D object analysis setup. [Rangrej *et al.*, 2022] introduces classification transformer architecture with an actor-critic module with an uncertainty estimator used for patch sampling. In our work, we actively search for successive glimpses defining the next patch sample using the entropy of the transformer attention maps. In opposition to [Rangrej *et al.*, 2022] or [Jha *et al.*, 2023], we do not need an additional neural network to make a glimpse decision. We also use no additional losses except those related to the target task.

3 Method

The fundamental idea of our approach is to leverage the internal uncertainty of the transformer-based autoencoder to select the most informative glimpses to be explored. Therefore, instead of using an additional loss function for glimpse selection, we propose a new method that obtains successive glimpses by analyzing the entropy of attention maps. This way, we can select the patch that has the highest entropy and confuses our model the most, which improves the convergence of our method and reduces its complexity compared to the existing methods.

As presented in Figure 2, our approach is based on a masked autoencoder with T run-times. In the beginning, we run it for one glimpse, and at step $t < T$, there are t known

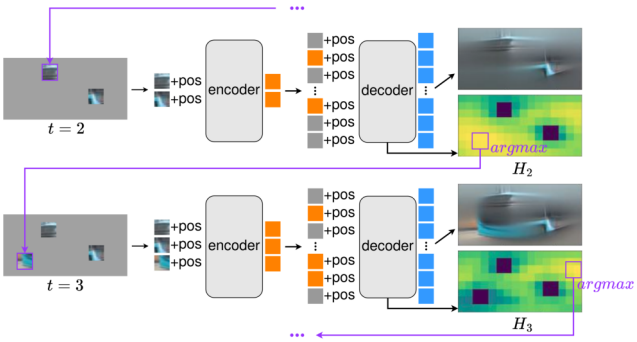


Figure 2: **Architecture for reconstruction:** The agent observed two patches of the image, which are processed by the encoder to produce their feature representations (orange rectangles). These outputs are combined with the masked patches (shown as gray rectangles) and passed through the decoder. The decoder reconstructs the missing image patches. Additionally, our method generates the entropy map for one of the decoder's multi-head self-attention layers and uses it to select the location of the third glimpse. The process repeats till we reach the assumed number of glimpses.

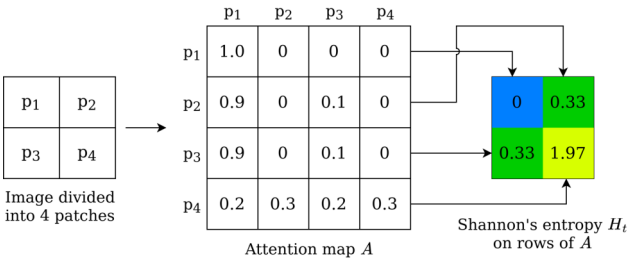


Figure 3: **Entropy map:** To explain the idea of the entropy map based on attention in the transformer layer, let us consider an image divided into four patches (2×2) on the left. Its attention map will be a 4×4 matrix, where each row represents the attention weights used to calculate the output in the next transformer layer for a corresponding patch. Calculating Shannon's entropy for each row will result in a 2×2 entropy map. The patch with the highest entropy value is selected as the next glimpse.

glimpses. To select the $t+1$ glimpse, we pass t glimpses selected so far through the masked autoencoder and calculate entropy for its attention maps. The masked patch with the highest entropy is chosen as the next glimpse. The network produces the final output for the target task (*e.g.* a reconstructed image for a reconstruction task) for $t = T$. This selection process is described in Section 3.2. Moreover, in Section 3.1 and Section 3.3, we describe the transformer-based autoencoder and the adjustments of its architecture to various target tasks.

3.1 Transformer-based Masked Autoencoder

Our approach employs a transformer-based Masked Autoencoder (MAE) network, as outlined in [He *et al.*, 2022], which includes an encoder and a decoder.

The encoder is a modified version of the Vision Transformer (ViT) [Dosovitskiy *et al.*, 2020]. As an input at a step t, it obtains a sequence of patches $p_1, p_2, ..., p_t : \forall_i p_i \in \mathbb{R}^{P^2 \times C}$ observed so far, where P is the patch size, and C is the number of input image channels. The patches are projected into E-dimensional embeddings using a linear layer and summed with positional embeddings. Then, they are processed through transformer blocks together with a learnable CLS token, resulting in t latent embeddings of dimension D. Because the number of visible patches may vary between images in a batch, we employ transformer blocks that allow pad tokens similar to those proposed in [Vaswani *et al.*, 2017].

The decoder is a transformer with a linear head that operates on t latent embeddings generated by the encoder and the coded positions of the $\frac{HW}{P^2} - t$ remaining patches, where H and W represent the height and width of the input image. The concatenated embeddings of dimension D are positionally encoded and processed through transformer blocks. The last part of the decoder is a linear layer with dimensions varies depending on the task (see Section 3.3).

3.2 Glimpse Selection

To explore the internal uncertainty of transformer-based models, we analyze the entropy of the attention map [Vaswani *et al.*, 2017], as presented in Figure 3. For this purpose, we start with calculating attention map $A \in \mathbb{R}^{t \times t}$ for patches $p_1, p_2, ..., p_t$ obtained so far as

$$A = \text{softmax}(KK^T/\sqrt{d_k}), \tag{1}$$

where $K \in \mathbb{R}^{t \times D}$ is a matrix that packs together all embeddings of patches from one of the decoder's layers. The successive rows of A contain weights used to calculate a weighted average of the patches outputs. Therefore, to obtain the output for patch i, we sum the values of all patches weighted with $A[i]$ (i-th row of A).

However, we can observe that the entropy of A rows significantly differs, and it can be used to select an adequate candidate for the next glimpse. To explain how, let us consider patches 2 and 4 from Figure 3 with $A[2] = [0.9, 0, 0.1, 0]$ and $A[4] = [0.2, 0.3, 0.3, 0.2]$. While the 2nd patch output is obtained based mostly on the value of the 1st patch, the 4th patch output is obtained based on all other patches. Hence, the model is confident about the 2nd patch but is uncertain about patch 4. Therefore, the latter is a better candidate for the next glimpse, as it needs to be explored.

To formalize this idea, let us also recall that transformers use multi-head attention, which computes multiple attention maps, each for embeddings obtained with different linear projections. Therefore, assuming that $A_{h,t}$ denotes the attention map of hth head in step t, the entropy map for a given multi-head self-attention layer is calculated as

$$H_t = \sum_h H(A_{h,t}[i]), \tag{2}$$

where H is the Shannon entropy. Then, we replace entropy values for known patches to 0 to omit them in the further selection process, and we choose the patch with the maximal value of H_t as the next glimpse

$$i^* = \text{argmax}_{i=1,...,d_{model}} H_t[i]. \tag{3}$$

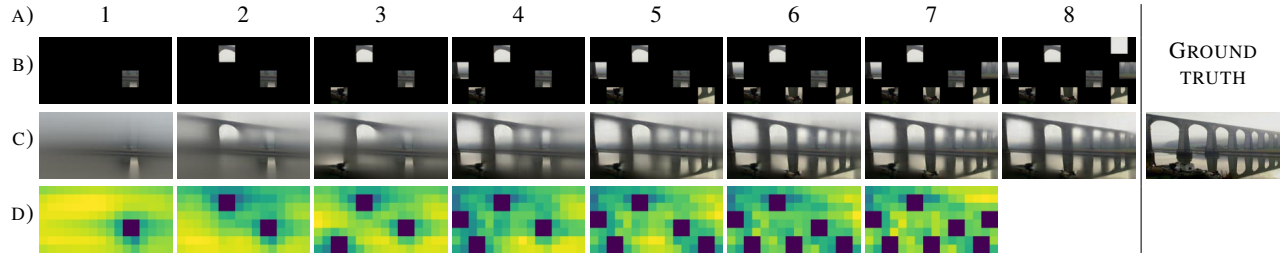


Figure 4: **Glimpse-based reconstruction step-by-step:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero). The algorithm explores the image in places where the reconstruction result is blurry.

3.3 Differences Caused by Final Task

For the reconstruction and segmentation tasks, the last layer of the decoder returns a representation of dimension $C' \times P^2$ for each processed patch, where C' is the number of input image channels for reconstruction and the number of segmented classes for segmentation. In this case, the output for the CLS token is omitted. For the classification task, we add an auxiliary classification head, which takes the CLS token from the encoder as input to classify the image.

4 Experimental Setup

4.1 General considerations

Architecture. In all our experiments, we use 24 transformer blocks in the encoder, with an embedding size of 1024 (the same parameters as the ViT-L architecture [Dosovitskiy *et al.*, 2020]), and 8 decoder blocks with an embedding size of 512. We use the AdamW optimization algorithm [Loshchilov and Hutter, 2018] with a weight decay value of 0 for reconstruction and 10^{-4} for other tasks. The learning rate is first linearly brought up to 10^{-4} for the first 10 training epochs, then decayed with the half-cycle cosine rate to 10^{-8} for the rest of the training. We train the model for 75 epochs with early stopping. We augment the training data with random scaling, cropping, and horizontal flipping. Then we resize the image to the required size. Unless otherwise specified, we initialize the model with MAE [He *et al.*, 2022] weights trained on the ImageNet-1K [Deng *et al.*, 2009] dataset. For 256×128 images, we also fine-tune those MAE weights on the target resolution using the MS COCO dataset [Lin *et al.*, 2014] using the original MAE implementation. For glimpse selection, we extract the attention map from the last decoder block. The first glimpse is selected by the entropy mechanism run with mask tokens only (zero known patches).

Datasets. We evaluate our method on 3 different vision datasets. The largest one, MS COCO dataset (Common Objects in Context; 2014 split version) [Lin *et al.*, 2014], consists of 83K train images and 41K validation images. The second one, ADE20K [Zhou *et al.*, 2017] dataset, consists of 26K training and 2k validation images, containing 150 classes for the semantic segmentation task. The smallest one is SUN360 [Song *et al.*, 2015] dataset with approximately 8K 360° panoramic images, unevenly split between 26 classes for the multi-class classification task. As the last dataset does not have a predetermined train-test split, we use a $9:1$ train-test split based on an index provided by authors of [Seifi *et al.*, 2021].

Glimpse regimes and baselines. We perform the experiments in two different glimpse regimes associated with two sets of state-of-the-art baselines.

First, we compare our method against fully convolutional approaches: Attend and segment [Seifi and Tuytelaars, 2020] and Glimpse, attend and explore [Seifi *et al.*, 2021]. Both methods use three-level retina-like glimpses, simulating the use of retina-like sensors introduced in [Sandini and Metta, 2003]. In this setup, an image is resized to 256×128 pixels, and the algorithm extracts 8 retinal glimpses of size 48x48 pixels. This corresponds to extracting 18.75% of pixels (accounting for retinal down-scaling) and 56.26% of the image area explored. As glimpses consist of multiple ViT patches, during the glimpse selection process, we average the glimpse selection map from Equation 2 over the size of a glimpse.

Secondly, we compare it to a transformer-based method called SimGlim [Jha *et al.*, 2023]. In this scenario, an image is resized to 224×224 pixels, and we extract 37 regular (non-retinal) glimpses of size 16x16 pixels (one ViT patch [Dosovitskiy *et al.*, 2020]). As a result, 18.75% of pixels are extracted, covering 18.75% of the image area.

The results of baseline methods (including visualizations) are copied from the original papers because we could not reproduce them due to incomplete or missing code.

We also conducted ablation experiments using two naive sampling methods. The first method is random sampling, which has the potential for overlap of glimpses. The second method samples glimpses from a fixed, non-overlapping checkerboard pattern [He *et al.*, 2022].

4.2 Tasks

Reconstruction. The reconstruction task involves recreating the ground-truth image with a limited number of glimpses. We evaluate our method on the MS COCO, ADE20K, and SUN360 datasets. The reconstruction quality is measured using the Root Mean Squared Error (RMSE), also used as a loss function.

Classification. We evaluate our model in the multi-class classification task on the SUN360 dataset using a limited number of glimpses. We use accuracy as a metric and cross entropy as a loss function. We evaluate our model in two

METHOD	SUN360	ADE20K	MS COCO	IMAGE RES.	GLIMPSE REGIME	PIXEL %	AREA %
ATTSEG	37.6	36.6	41.8	128×256	8×48^2 (RETINAL)	18.75	56.25
GLATEX	33.8	41.9	40.3	128×256	8×48^2 (RETINAL)	18.75	56.25
OURS (RETINAL)	**23.6**	**23.8**	**25.2**	128×256	8×48^2 (RETINAL)	18.75	56.25
OURS (NON-RETINAL)	37.9	40.7	43.2	128×256	8×16^2 (NON RET.)	6.25	6.25
OURS (NON-RETINAL)	29.8	30.8	32.5	128×256	8×32^2 (NON RET.)	25.00	25.00
OURS (NON-RETINAL)	**20,1**	**20.6**	**22.1**	128×256	8×48^2 (NON RET.)	56.25	56.25
SIMGLIM (DETACHED)	26.2	27.2	29.8	224×224	37×16^2 (NON RET.)	18.75	18.75
SIMGLIM (END-TO-END)	28.0	28.8	31.3	224×224	37×16^2 (NON RET.)	18.75	18.75
OURS (NON-RETINAL)	**23.4**	**26.2**	**28.6**	224×224	37×16^2 (NON RET.)	18.75	18.75

Table 1: **Reconstruction results:** Comparison of our model in reconstruction task against AttSeg [Seifi and Tuytelaars, 2020], GlAtEx [Seifi *et al.*, 2021] and SimGlim [Jha *et al.*, 2023] on SUN360, ADE20K and MS COCO datasets. The metric used is a root mean square error (RMSE; lower is better). For each experiment, we provide a training and evaluation regime defined by a number of glimpses of a specific resolution. Pixel % and area % denote respectively: the percentage of image pixels known to the model and the percentage of image area seen by the model. Differences in both measures occur when dealing with retina-like glimpses, which have lower pixel counts by design. Our method outperforms competitive solutions in all configurations.

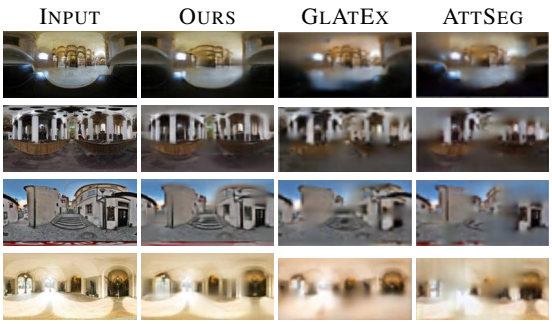


Figure 5: **Reconstruction quality for SUN360:** Reconstruction results of our method compared with AttSeg [Seifi and Tuytelaars, 2020] and GlAtEx [Seifi *et al.*, 2021] on the SUN360 dataset. Reconstructions done with our method are visibly better than the competition.

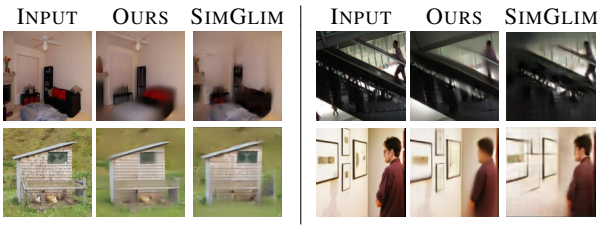


Figure 6: **Reconstruction quality for ADE20K:** Figure shows difference in reconstruction quality against SimGlim [Jha *et al.*, 2023] on the ADE20K dataset. SimGlim reconstructs a single object slightly better, but our method recovers more objects in the scene.

training regimes: a) training the entire model for classification (*train-all*) and b) training only the classification head-on features generated by a model trained on the reconstruction task (*head-only*). The classification head consists of linear layers: one in the *train-all* scenario or two separated by a GELU activation layer for the *head-only* scenario. For the *train-all* scenario, we do a weighted sum of the classification loss (cross-entropy) and a loss of the decoder (required for the glimpse selection mechanism).

Segmentation. The goal of the segmentation task is to classify each pixel of the input image. Our method is tested on the ADE20K dataset. The mean Pixel Accuracy (mPA; averaged over classes), Pixel Accuracy (PA; averaged over all pixels), and Intersection over Union (IoU) are used as a metric, and the pixel-wise cross-entropy is used as a loss function. We initialize the transformer weights in two ways: using MAE or SETR (trained on ADE20k) weights as the source of encoder weights. In both cases, the decoder is initialized randomly.

5 Results

To illustrate how the glimpse selection process works step by step, we present it in Figure 4 on a sample image from the ADE20K dataset. As can be seen, with each glimpse, not only the glimpse neighborhood but the entire output image becomes clearer because the transformer gains a better understanding of the scene.

5.1 Reconstruction

In our study, we demonstrate superior performance in the retina-like glimpse setup for all datasets examined. Supporting results are presented in the top part of Table 1. Additionally, as shown in Figure 5, the reconstructions produced by our method are of higher quality, with visibly increased clarity and sharpness.

We also conduct tests without retina degradation of patches and show that using non-retinal glimpses improved results by over 10% of all scores. Moreover, our tests with smaller non-retinal patch sizes show that we still achieved better RMSE scores using less than half the area. In this experiment, we use eight patches in a 32x32 resolution, using 25% of the area compared to 56.25% in baseline, as seen in the middle part of Table 1.

When comparing our method to the transformer-based SimGlim [Jha *et al.*, 2023], we find that our solution further improves reconstruction scores across all of the databases.

METHOD	ACCURACY (%)
ATTSEG	52.6
GLATEX (FULL)	56.4
GLATEX (NO DECODER)	67.2
OURS (HEAD-ONLY)	70.1
OURS (TRAIN-ALL)	**75.7**

Table 2: **Classification results:** Comparison of our model's classification performance against AttSeg [Seifi and Tuytelaars, 2020] and GlAtEx [Seifi *et al.*, 2021] on the SUN360 dataset. The metric used is accuracy (higher is better). Our method outperforms competitive methods in both Train-all and Head-only training options.

METHOD	MPA(%)	PA(%)	IOU(%)
ATTSEG	-	47.9	-
GLATEX	-	52.4	-
OURS (MAE-WEIGHTS)	32.2	**70.27**	24.4
OURS (SETR-WEIGHTS)	**35.6**	69.5	**27.6**

Table 3: **Segmentation results:** Comparison of our model against AttSeg [Seifi and Tuytelaars, 2020] and GlAtEx [Seifi *et al.*, 2021]. The metric used is mean Pixel-wise Accuracy (mPA, higher is better), Pixel-Accuracy (PA, higher is better), and Intersection over Union (IoU, higher is better). Our solution outperforms competitive methods on all metrics.

This is shown in the bottom part of Table 1. It is noteworthy that the authors of [Jha *et al.*, 2023] also used MAE [He *et al.*, 2022] architecture for the encoder and decoder. Therefore, improvement is mostly achieved from the glimpse selection method we propose rather than the underlying architecture.

In the qualitative comparison shown in Figure 6, we can observe that while SimGlim produces more detailed reconstructions of individual objects, our method can discover and reconstruct more objects in the image overall, indicating better exploration capabilities. Moreover, authors of [Jha *et al.*, 2023] analyzed attention-based solutions for glimpse selection and concluded that it was not performing better than random sampling. However, in our study, we observe that our more sophisticated way of using this information outperforms random approaches. We provide further discussion on these ablation studies in Section 5.5.

5.2 Classification

Our method showed improved performance compared to the solutions presented in [Seifi *et al.*, 2021] and [Seifi and Tuytelaars, 2020] in both training modes (train-all and head-only), as presented in Table 2. These results align with the improvements observed in the reconstruction task. The use of transformer-based architecture in the underlying MAE [He *et al.*, 2022] is likely a significant contributing factor to these improvements results, as we achieved better results even with auto-encoders weights frozen during training (head-only).

5.3 Segmentation

Consistently, our method obtains better results in the segmentation task than compared, state of the art methods. As presented in Table 3, our approach achieves higher PA scores than those reported in [Seifi and Tuytelaars, 2020] and [Seifi *et al.*, 2021]. Note, that the authors of both baseline methods present their results as mean pixel accuracy (mPA - averaged over classes), while in fact they report pixel accuracy (PA - averaged over all pixels). We confirmed this finding with the authors directly. The qualitative results presented in Figure 7 further validate the superior performance.

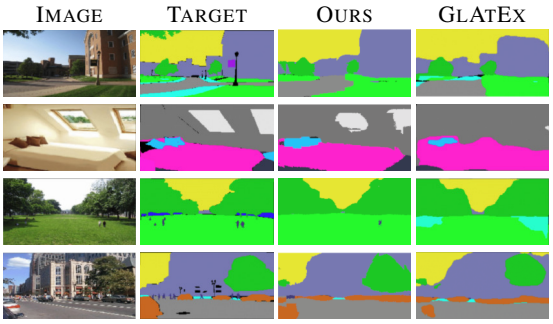


Figure 7: **Segmentation quality for ADE20K:** Semantic segmentation results of our method compared with GlAtEx [Seifi *et al.*, 2021] on the ADE20k dataset. Qualitatively, segmentation maps produced by our method are at least as good as those of the competition.

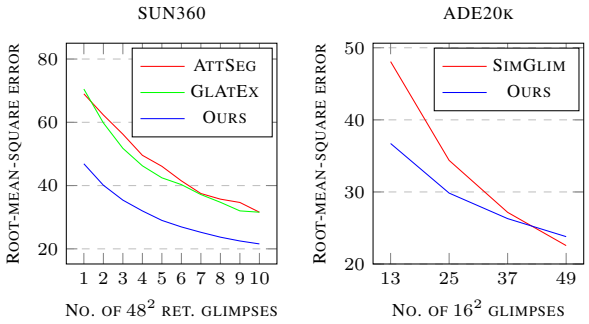


Figure 8: **Comparative reconstruction results in regards to glimpse selection step**: Figures present the effect of the number of glimpses on the reconstruction task. On the left, we compare our model against AttSeg [Seifi and Tuytelaars, 2020] and GlAtEx [Seifi *et al.*, 2021] on the SUN360 dataset in 48^2 retinal glimpse scenario. On the right, we compare our model against SimGlim [Jha *et al.*, 2023] on ADE20K dataset in 16^2 non-retinal glimpse task. Our solution outperforms competitive solutions when only a small amount of the image is available.

5.4 Glimpse Analysis

Number of glimpses. The relationship between model performance in the reconstruction task and the number of allowed glimpses is illustrated in Figure 8. Our model consistently outperforms all convolutional baselines. In the case of MAE-based SimGlim [Jha *et al.*, 2023], our solution performs better until around 20% of patches are visited. This can be attributed to the fact that our solution optimizes the general reconstruction task, whereas the latter focuses on specific types of visited data when there is enough visual information already acquired.

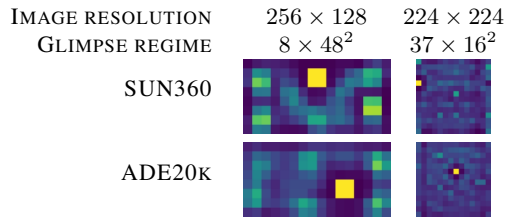


Figure 9: **Average glimpse image:** Mean glimpse map, averaged over all glimpse positions in all test images for reconstruction task on SUN360 and ADE20K. We can see that the first glimpse is consistently selected in the same place. Dataset and training regimes determine its best starting position.

SELECTION	GLIMPSE REGIME	SUN360	MS COCO
RANDOM	8×32^2	32.8	38.3
CHECKER	8×32^2	32.0	32.6
ATTENTION	8×32^2	**29.8**	**32.5**
RANDOM	8×16^2	39.8	47.3
CHECKER	8×16^2	39.3	44.1
ATTENTION	8×16^2	**37.9**	**43.2**

Table 4: **AME-based glimpse selection versus naive sampling - reconstruction:** Comparison of glimpse selection strategies for reconstruction task on SUN360 and MS COCO databases. The metric used is a root mean square error (RMSE; lower is better). We used regular (non-retinal) glimpses for this ablation. Our method performs better than random position sampling and random sampling from a checkerboard-like pattern.

Average glimpse selection. The average glimpse location masks are shown in Figure 9. For the 8×48^2 glimpse setup, it can be observed that the glimpse selector prefers locations in the image corners and in the center. It rarely selects locations on the left and right edge for the SUN360 dataset, which may be because the dataset consists of 360° panoramic images. Selection masks for the 37×16^2 setup are much more evenly distributed, with only the first glimpse location being distinctive.

5.5 Glimpse Selection Strategies

Our attention-based glimpse selection process outperforms random glimpse selection and a fixed checkerboard-like glimpse pattern, as demonstrated in Table 4 and 5. This supports our claim that our selection strategy is superior to naive strategies and contradicts the discussions presented in [Jha *et al.*, 2023].

Level of attention map. In Figure 10, we demonstrate the impact of the choice of decoder layer as the attention map source on the model performance. The results show that the further in the decoder the attention maps are sourced from, the better the overall performance.

6 Conclusions

This paper presents a new approach to active visual exploration that simplifies the existing methods by leveraging internal model uncertainty in glimpse selection process. By utilizing the entropy of the transformer's attention maps, we eliminate the need for any sampling decision module or separate network head focused on decision strategy. Furthermore, our approach does not require separate training for the main task and the sampling decision strategy, reducing the complexity of the overall system. Our approach is based on the masked auto-encoder architecture, which allows for the efficient processing of partial observations. The results demonstrate that the attention values trained inherently from the transformer-based decoder can be used to generate decision-making priors that improve the performance of scene understanding tasks. An exhaustive empirical comparison against the competing architectures shows that our solution performs better when limited data is available.

SELECTION	GLIMPSE REGIME	TRAIN ALL	HEAD-ONLY
RANDOM	8×32^2	72.4	66.2
CHECKER	8×32^2	70.1	64.0
ATTENTION	8×32^2	**73.4**	**66.9**
RANDOM	8×16^2	61.9	53.7
CHECKER	8×16^2	57.8	**56.2**
ATTENTION	8×16^2	**63.4**	55.9

Table 5: **AME-based glimpse selection versus naive sampling - classification:** Comparison of glimpse selection strategies in classification task for SUN360 database. The metric used is accuracy (higher is better). We used regular (non-retinal) glimpses for this ablation and tested full-model and head-only training strategies. Our method performs better than random position sampling and random sampling from a checkerboard-like pattern.

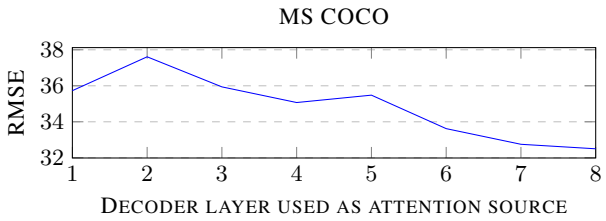


Figure 10: **RMSE results versus attention map source:** Comparison of attention-based glimpse selection results in regards to decoder layer used as attention map source. The figure shows that the best results can be achieved using the last layer of the decoder. The dataset used is MS COCO. The metric is a root mean square error (RMSE; lower is better). The image resolution is 256×128 in an 8×32^2 non-retinal glimpse regime.

Acknowledgments

This research was funded by National Science Centre, Poland (grants no 2020/39/B/ST6/01511, 2022/45/B/ST6/02817, and 2021/41/B/ST6/01370), Foundation for Polish Science (grant no POIR.04.04.00-00-14DE/18-00 carried out within the Team-Net program co-financed by the European Union under the European Regional Development Fund). We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2022/015753. The research was sup-

ported by a grant from the Faculty of Mathematics and Computer Science under the Strategic Programme Excellence Initiative at Jagiellonian University.

References

[Aloimonos *et al.*, 1988] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International journal of computer vision*, 1(4):333–356, 1988.

[Chai, 2019] Yuning Chai. Patchwork: A patch-wise attention network for efficient object detection and segmentation in video streams. *CoRR*, abs/1904.01784, 2019.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[Girdhar *et al.*, 2022] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2022.

[He *et al.*, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[Jayaraman and Grauman, 2017] Dinesh Jayaraman and Kristen Grauman. Learning to look around. *CoRR*, abs/1709.00507, 2017.

[Jha *et al.*, 2023] Abhishek Jha, Soroush Seifi, and Tinne Tuytelaars. Simglim: Simplifying glimpse based active visual reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 269–278, January 2023.

[Li *et al.*, 2022] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[Loshchilov and Hutter, 2018] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[Naseer *et al.*, 2021] Muzammal Naseer, Kanchana Ranasinghe, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *CoRR*, abs/2105.10497, 2021.

[Pathak *et al.*, 2016] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[Ramakrishnan and Grauman, 2018] Santhosh K. Ramakrishnan and Kristen Grauman. Sidekick policy learning for active visual exploration. *CoRR*, abs/1807.11010, 2018.

[Ramakrishnan *et al.*, 2021] Santhosh K. Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. An exploration of embodied visual exploration. *Int. J. Comput. Vis.*, 129:1616–1649, 2021.

[Rangrej and Clark, 2021] Samrudhdhi B. Rangrej and James J. Clark. Visual attention in imaginative agents. *CoRR*, abs/2104.00177, 2021.

[Rangrej *et al.*, 2022] Samrudhdhi B Rangrej, Chetan L Srinidhi, and James J Clark. Consistency driven sequential transformers attention model for partially observable scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2518–2527, 2022.

[Sandini and Metta, 2003] Giulio Sandini and Giorgio Metta. Retina-like sensors: motivations, technology and applications. In *Sensors and sensing in biology and engineering*, pages 251–262. Springer, 2003.

[Seifi and Tuytelaars, 2019] Soroush Seifi and Tinne Tuytelaars. Where to look next: Unsupervised active visual exploration on 360° input. *CoRR*, abs/1909.10304, 2019.

[Seifi and Tuytelaars, 2020] Soroush Seifi and Tinne Tuytelaars. Attend and segment: Attention guided active semantic segmentation. *CoRR*, abs/2007.11548, 2020.

[Seifi *et al.*, 2021] Soroush Seifi, Abhishek Jha, and Tinne Tuytelaars. Glimpse-attend-and-explore: Self-attention for active visual exploration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16137–16146, 2021.

[Śmieja *et al.*, 2018] Marek Śmieja, Łukasz Struski, Jacek Tabor, Bartosz Zieliński, and Przemysław Spurek. Processing of missing data by neural networks. *Advances in neural information processing systems*, 31, 2018.

[Song *et al.*, 2015] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wenzel *et al.*, 2021] Patrick Wenzel, Rui Wang, Nan Yang, Qing Cheng, Qadeer Khan, Lukas von Stumberg, Niclas Zeller, and Daniel Cremers. 4seasons: A cross-season dataset for multi-weather slam in autonomous driving. In Zeynep Akata, Andreas Geiger, and Torsten Sattler, editors, *Pattern Recognition*, pages 404–417, Cham, 2021. Springer International Publishing.

[Zhou *et al.*, 2017] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

Active Visual Exploration Based on Attention-Map Entropy – Supplementary Materials

Adam Pardyl[1,2,3] , **Grzegorz Rypeść**[1,4] , **Grzegorz Kurzejamski**[1] ,
Bartosz Zieliński[1,2,6] , **Tomasz Trzciński**[1,4,5]
[1]IDEAS NCBR
[2]Jagiellonian University, Faculty of Mathematics and Computer Science
[3]Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4]Warsaw University of Technology
[5]Tooploox
[6]Ardigen
{firstname.lastname}@ideas-ncbr.pl

In this Supplementary Materials, we present additional information regarding performed experiments, results, and visualizations for our Attention-Map Entropy active visual exploration method.

1 Details on models used for experiments

We implemented the Attention-Map Entropy algorithm using Python (version 3.9) and PyTorch (version 1.13) library. In this section we explain the structure, hyperparameters, and training regime of AME which we use in Section 4 of the paper.

1.1 Structure

For each task (reconstruction, segmentation, and classification), we use the same structure of the encoder. It consists of 24 standard transformer blocks, which are initialized with weights of the MAE ViT-Large model that was pre-trained on the ImageNet 1K dataset. Each encoder block consists of 16 self-attention heads. The input to the encoder are patches of size 16x16 which are later transformed into embedding vectors of size 1024. The feed forward network in each block consists of 4096 hidden neurons. As the number of input patches may vary between images in batch, we use padding tokens to even out the sequence length. We explicitly assign zero attention scores to padding tokens to mask them from computation, which is a standard approach used in NLP transformers. The difference in the architecture between tasks lies in the decoder part of the architecture.

For reconstruction, we use a decoder consisting of 8 transformer blocks followed by a linear head, initialized with MAE weights. The size of embedding in the decoder is 512, each block uses 16 self-attention heads. The feed forward network in each block has a size of 2048.

Similarly, for segmentation, we utilize 8 transformer blocks followed by the linear head as the decoder; however, we randomly initialize all parameters.

For classification, we attach an auxiliary classification head to the encoder output. The head takes as input the *cls* token embedding of size 1024, as returned by the encoder. It consists of linear layers, one for the *train-all* scenario and two for the *head-only* regime, as described in the Experimental Setup section of the main paper. For two linear layer classification head we set the latent dimension size to 256, and use a GELU activation between linear layer.

1.2 Training

During training we utilize AdamW optimizer and set the initial learning rate to 0.0001. We use the half-cycle cosine learning rate scheduling and set the number of epochs to 100. We set the weight decay factor to 0.0001. We choose the best model based on values of metrics on validation dataset.

2 Additional visualizations

In this section, we provide additional visualizations for both the reconstruction and segmentation tasks.

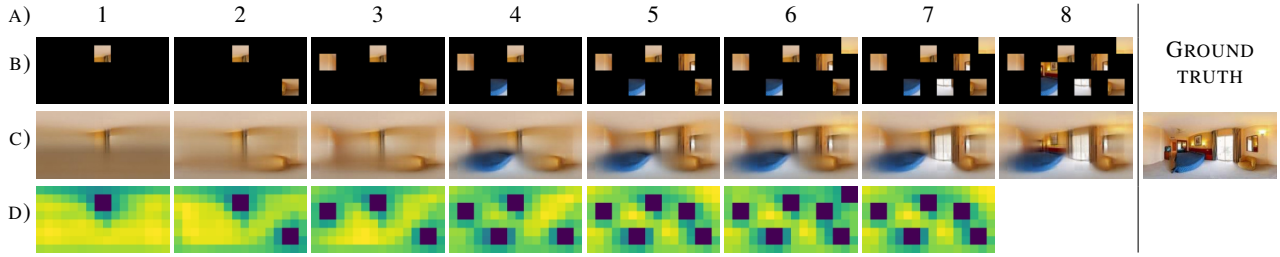


Figure 1: **Glimpse-based reconstruction step-by-step on SUN360:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

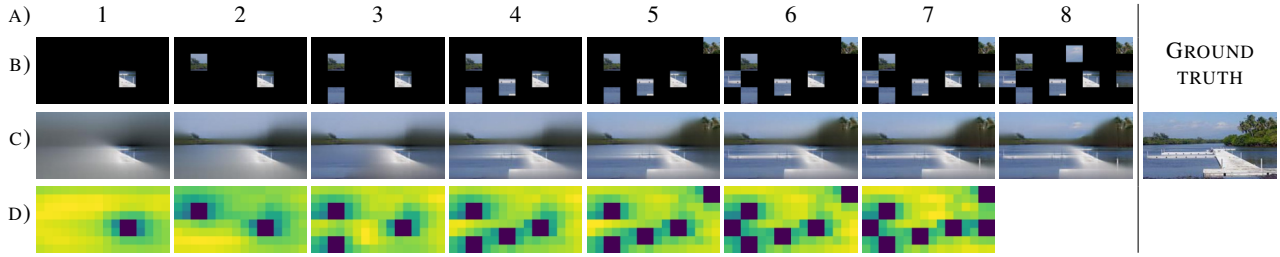


Figure 2: **Glimpse-based reconstruction step-by-step on ADE20K:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

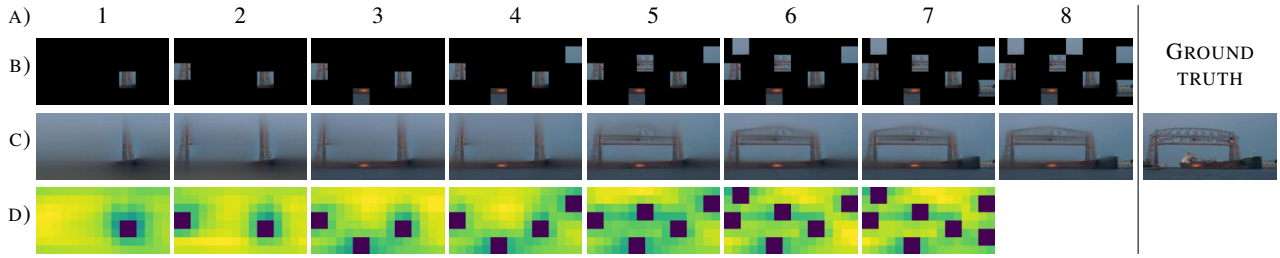


Figure 3: **Glimpse-based reconstruction step-by-step on ADE20K:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

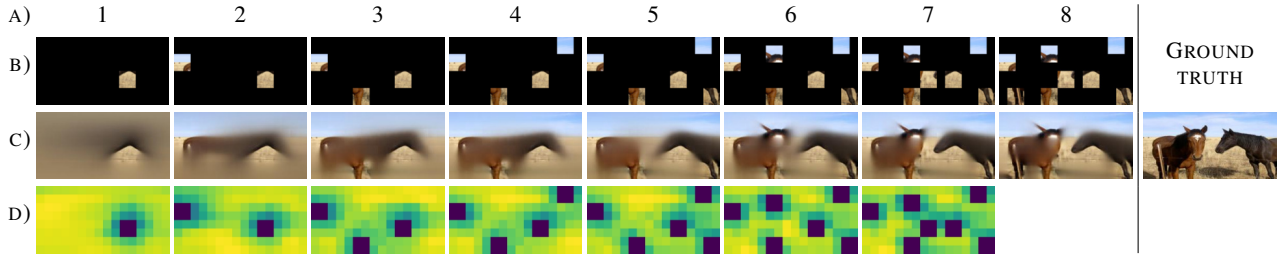


Figure 4: **Glimpse-based reconstruction step-by-step on MS COCO:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

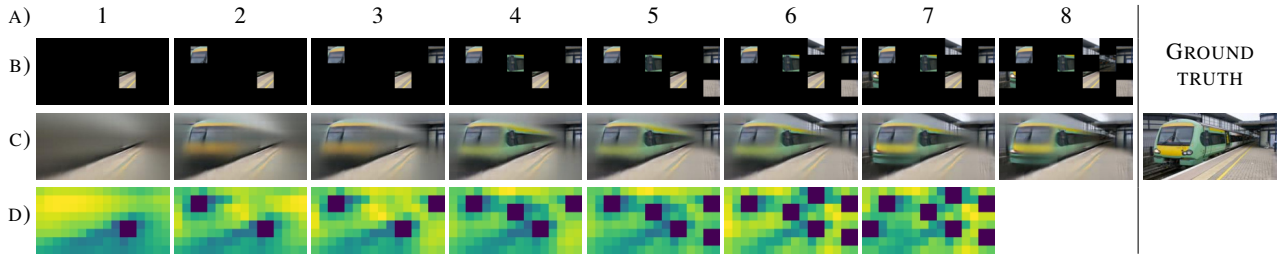


Figure 5: **Glimpse-based reconstruction step-by-step on MS COCO:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

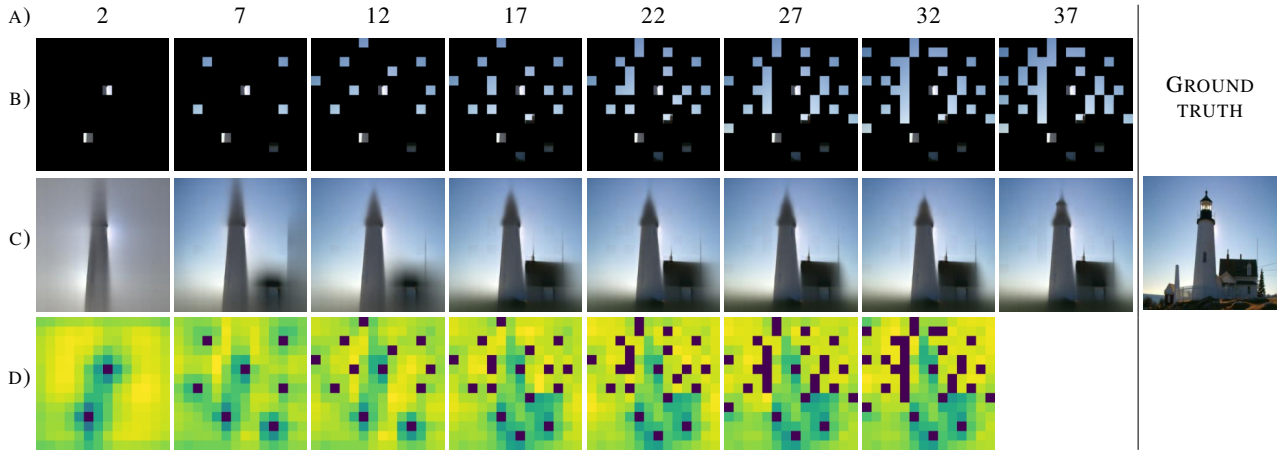


Figure 6: **Glimpse-based reconstruction step-by-step on ADE20K:** The figure shows a glimpse selection process based on AME for 37×16^2 glimpses for a sample 224×224 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

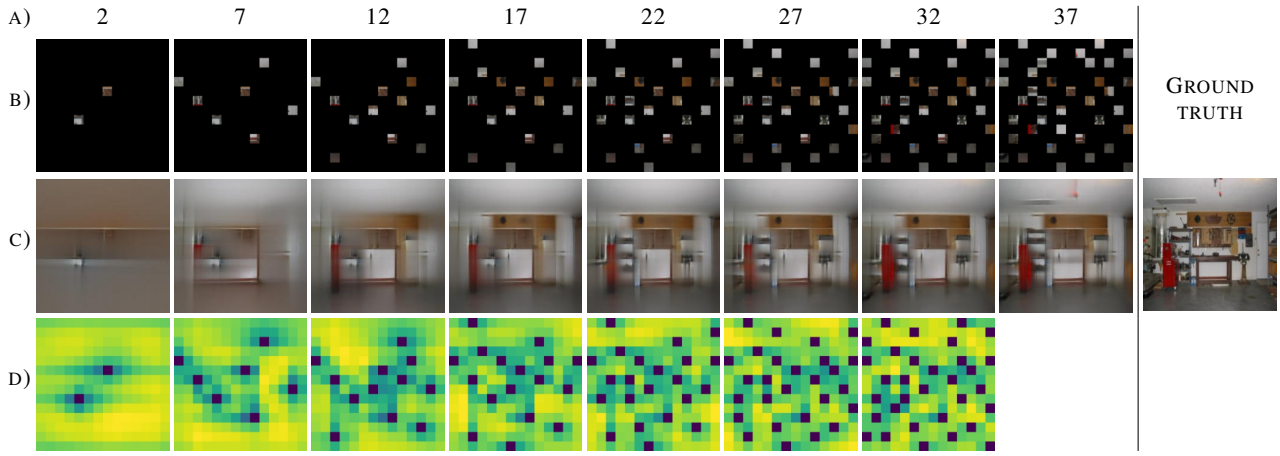


Figure 7: **Glimpse-based reconstruction step-by-step on ADE20K:** The figure shows a glimpse selection process based on AME for 37×16^2 glimpses for a sample 224×224 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

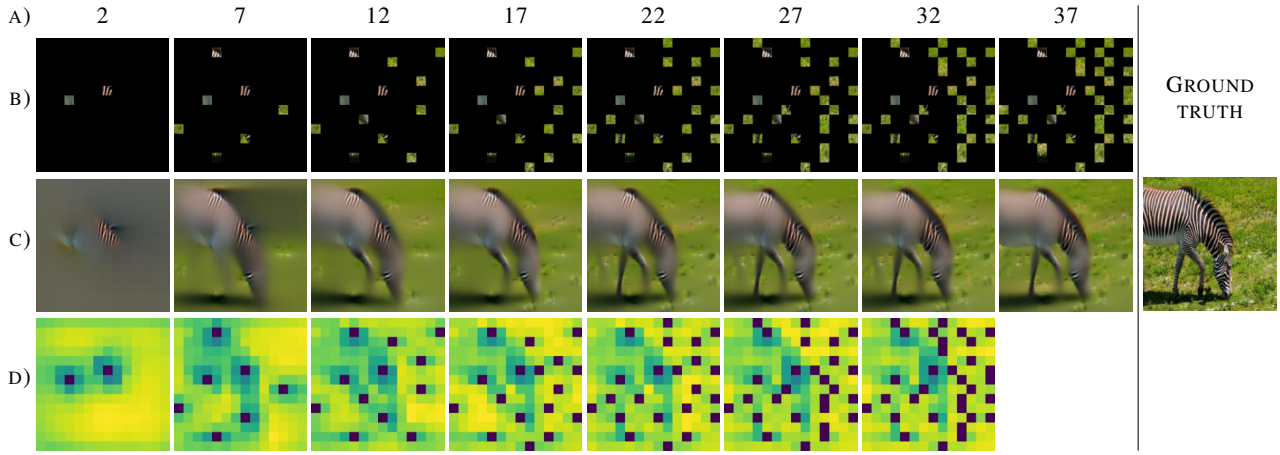


Figure 8: **Glimpse-based reconstruction step-by-step on MS COCO:** The figure shows a glimpse selection process based on AME for 37×16^2 glimpses for a sample 224×224 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

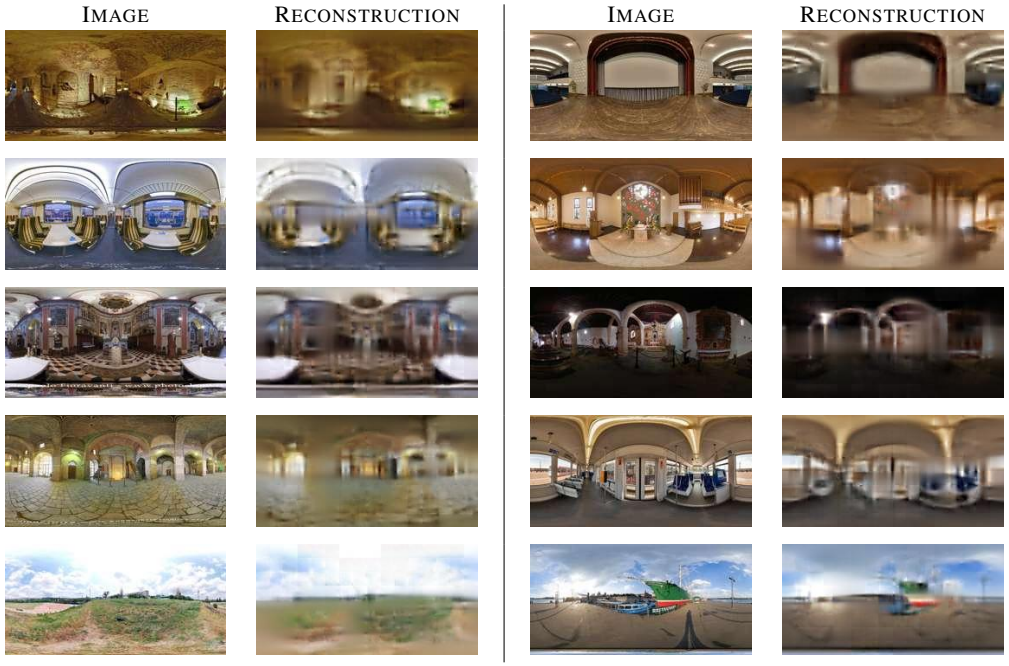


Figure 9: **Image reconstruction on SUN360:** Figure shows the qualitative results of the image reconstruction task. Image size is 256×128 and 8×48^2 retinal glimpses are used.

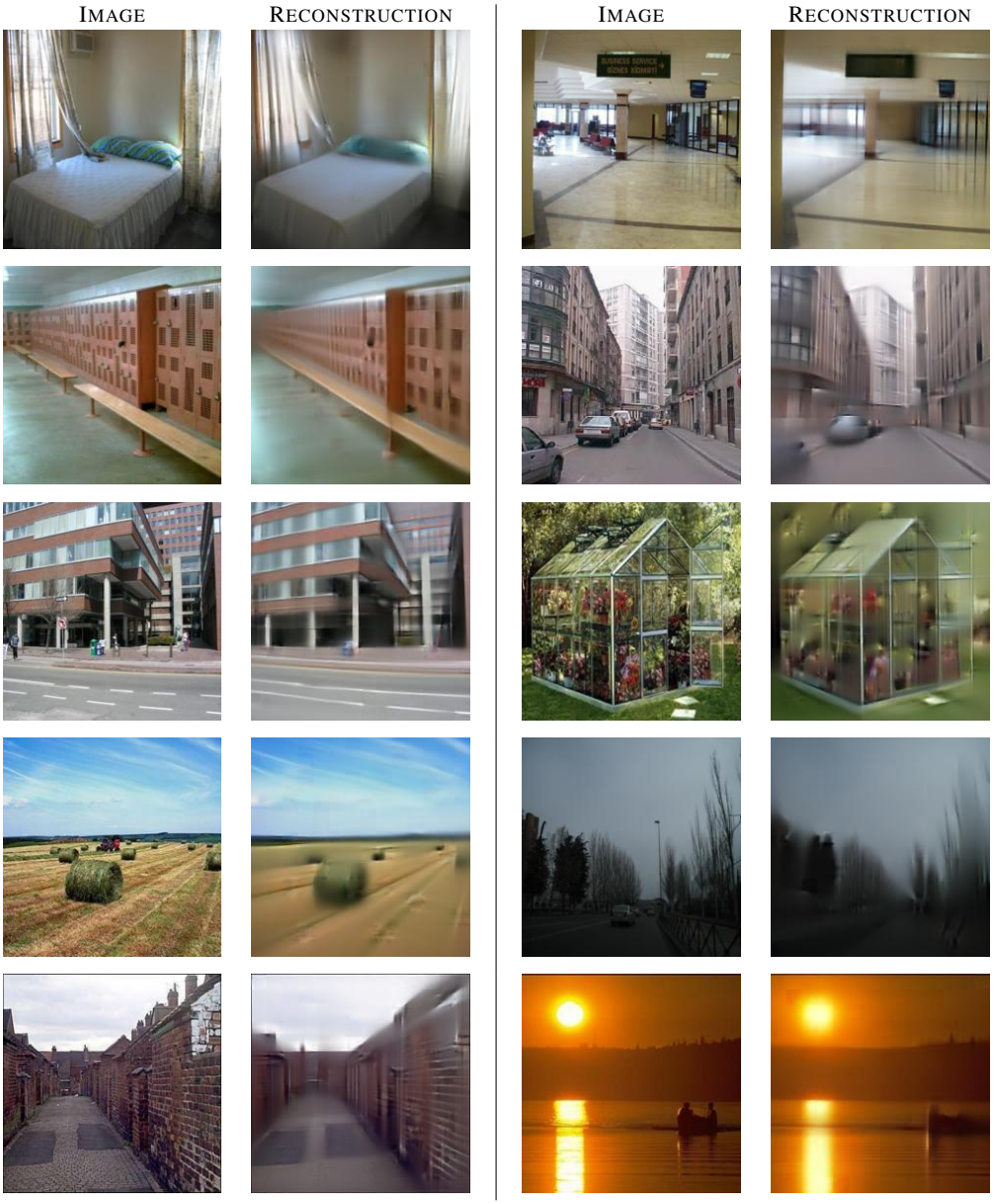


Figure 10: **Image reconstruction on ADE20K:** Figure shows the qualitative results of the image reconstruction task. Image size is 224×224 and 37×16^2 glimpses are used.

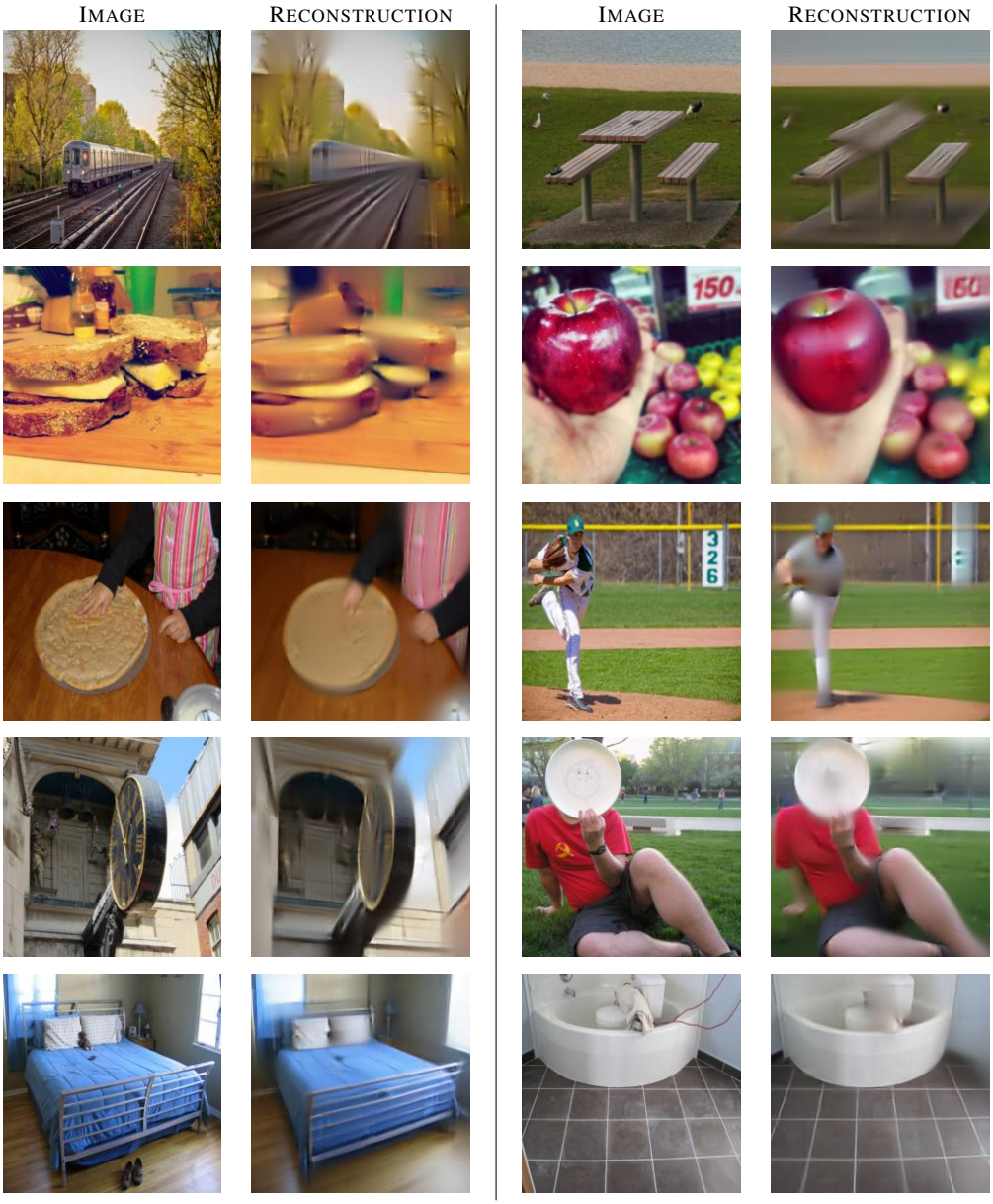


Figure 11: **Image reconstruction on MS COCO:** Figure shows the qualitative results of the image reconstruction task. Image size is 224×224 and 37×16^2 glimpses are used.

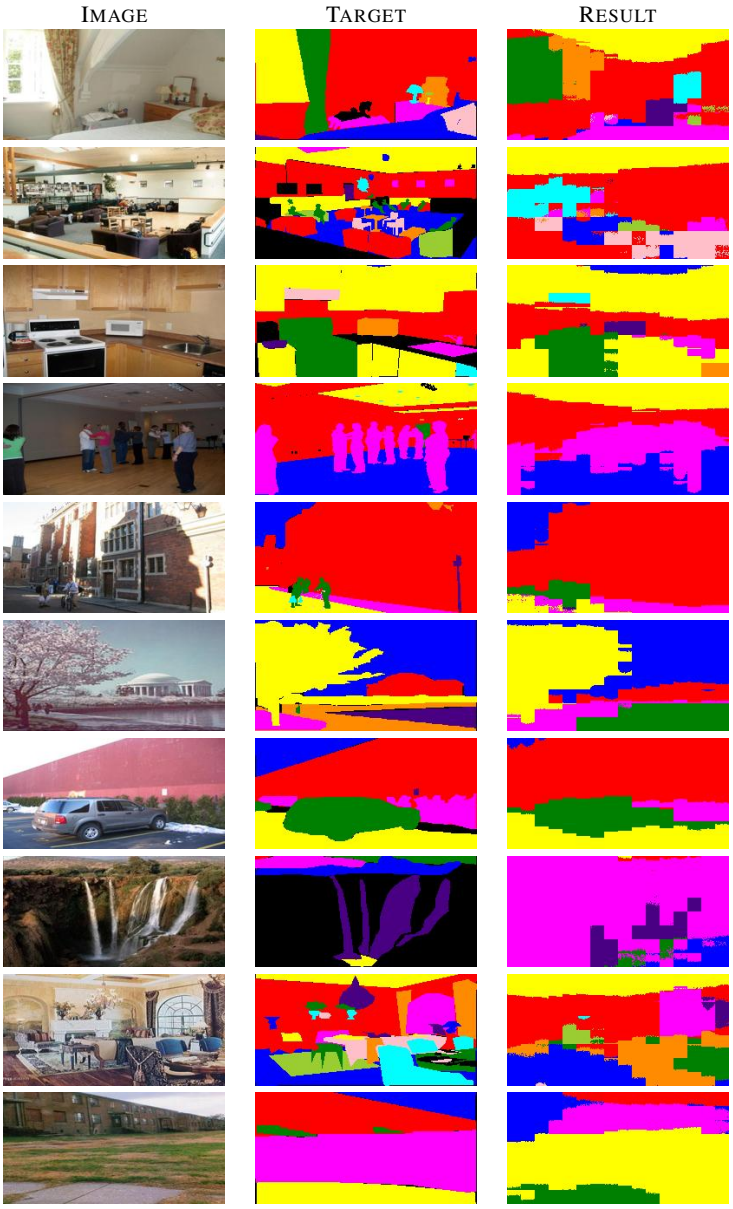


Figure 12: **Image segmentation on ADE20K:** Figure shows the qualitative results of the image segmentation task. Image size is 256×128 and 8×48^2 retinal glimpses are used.

Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers

Adam Pardyl[1,2,3] Grzegorz Kurzejamski[1] Jan Olszewski[1,4]
Tomasz Trzciński[1,5,6] Bartosz Zieliński[1,2]
[1]IDEAS NCBR
[2]Jagiellonian University, Faculty of Mathematics and Computer Science
[3] Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4] University of Warsaw [5] Warsaw University of Technology [6] Tooploox

Abstract

Vision transformers have excelled in various computer vision tasks but mostly rely on rigid input sampling using a fixed-size grid of patches. It limits their applicability in real-world problems, such as active visual exploration, where patches have various scales and positions. Our paper addresses this limitation by formalizing the concept of input elasticity for vision transformers and introducing an evaluation protocol for measuring this elasticity. Moreover, we propose modifications to the transformer architecture and training regime, which increase its elasticity. Through extensive experimentation, we spotlight opportunities and challenges associated with such architecture.

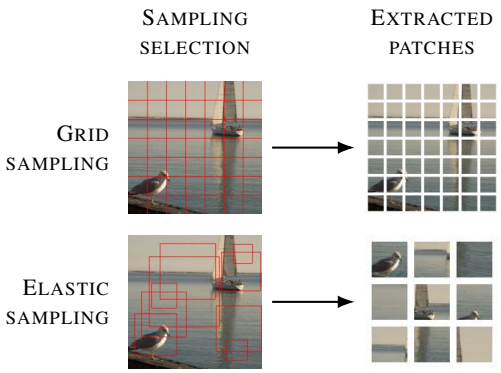


Figure 1. **Grid vs. elastic sampling:** patch sampling with arbitrary patch positions and scales creates new possibilities for more effective and efficient vision transformers.

1. Introduction

Vision Transformers (ViT) [8] achieve state-of-the-art results in many computer vision applications [4, 13, 33]. They cut the image into a regular grid of non-overlapping patches and pass them as tokens to attention modules that exchange information between them. Many variations of the algorithm were created [5, 21, 25, 32], including cross modal [1] and long-term memory [31] solutions.

Nevertheless, almost all transformer-based methods assume that input tokens form a regular grid of 16×16 pixel patches, limiting their applicability in real-life problems. For example, in Active Visual Exploration (AVE) [20], in which an agent has to navigate an environment by actively choosing successive observations while having a restricted field of view. AVE is commonly encountered in robotic applications, where an agent must localize dangers as quickly as possible based on incomplete images and observations of various scales and positions.

In this paper, we will consider two research questions. First, we ask if the standard ViT architectures are resilient to input perturbations that commonly occur in the vision tasks of embodied AI, such as AVE. For this purpose, we introduce an evaluation protocol that measures three necessary types of input elasticity a good model should showcase: *scale elasticity*, *position elasticity*, and *missing data elasticity*. We use this protocol to measure the input elasticity of common ViT architectures. Then, we move on to the second question: How can ViT input elasticity be increased by modifying model architecture and a training policy? We propose architectural and training modifications, named *ElasticViT*[1], to increase the elasticity.

We hope this work will draw the attention of the machine learning community to the important topic of input sampling strategies. It is crucial because, according to recent studies [2], accommodating alternative patch resolutions can significantly impact the algorithm's real-life performance. We can summarize our contributions as follows:

- We formalize the notion of input elasticity for vision transformers and provide a comprehensive evaluation protocol to assess it.

[1]Our code is available at: https://github.com/apardyl/beyondgrids

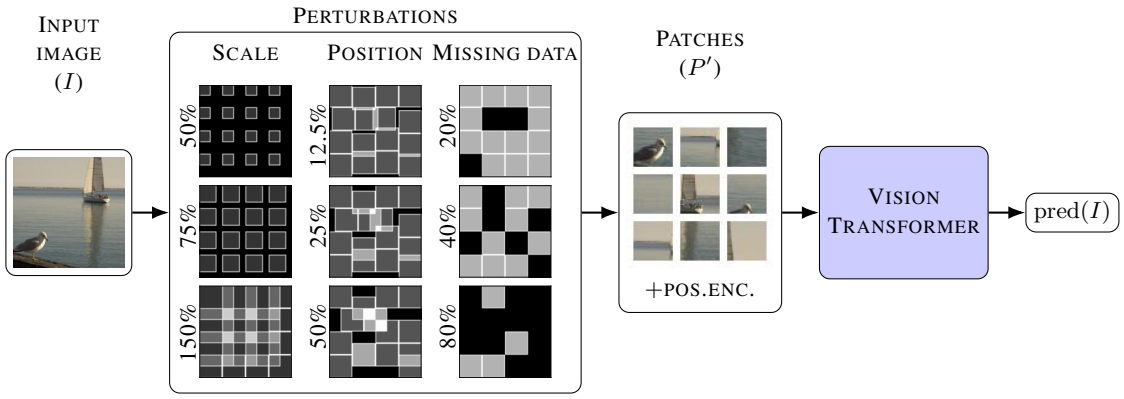


Figure 2. **Evaluation protocol:** To analyze scale, position, and missing data elasticity we introduce three types of perturbations that are applied separately to each patch from the sampling grid.

- We propose modifications to the vision transformer architecture and training policy, including a novel *PatchMix* augmentation, that increase elasticity.
- We show that elastic sampling strategies can boost transformers' performance, especially for significantly limited input data.

2. Related Work

Vision Transformers (ViTs). Vision transformer introduced in [8] is a versatile method due to its state-of-the-art performance obtained for many visual tasks [26]. Swin [19] modifies the attention mechanism to increase the locality of information exchange between tokens. AViT [5] demonstrates how transformer neural networks can be dynamically scaled. Pyramid ViT [28] implements a multi-scale feature extraction model using ViT, inspired by previous work on convolutional neural networks. Vision Longformer [36] uses a token-based input format that enables seamless addition of global context tokens and control of the computation complexity. Other positional encoding methods for vision transformers were studied in [6, 12, 22].

ViT sampling strategies. Many works explore the possibility of using different grid resolutions as ViT inputs. Some perform different grid scale sampling during training [14, 30], and others introduce position and patch encoding rescaling tricks [2]. They usually improve ViT's accuracy across grids with varying resolutions and constant patches or when using native resolution grid sampling with a variable number of patches in a batch [7].

ViTs applications. Initially, vision transformers were used in classical computer vision tasks, like object detection [4] and semantic segmentation [13]. However, as the field matures, the models are being deployed in real-world scenarios [15], where it is often necessary to understand input to the model as a collection of images captured from multiple views, each contributing partial information about a scene rather than a single image. Active visual exploration [11, 20] is one possible setup of such a real-world scenario. While performing the tasks, robots or drones often capture images that cover only part of the scene and rarely come in grid-aligned format with consistent scale. Therefore, it is crucial to provide higher input elasticity. Some of them were already considered, like ViT resiliency to missing data [10] has already been conducted [18, 25, 32], but their cumulative effect on performance is unknown.

3. Evaluation protocol

In this section, we first define three types of model elasticity corresponding to configurations that occur in embodied robotics data, and then we propose the evaluation protocol, which we implement by applying perturbations to the input sampling procedure. We use the protocol to rank the models by their overall elasticity.

3.1. Elasticities

We define model *elasticity* as resilience to particular forms of input data configurations that can occur in real-life applications.

Scale elasticity is defined as resilience to the relative scale change across image tokens processed by the model. Standard ViT algorithm uses fixed patch size and normalizes the size of each input image, while recent works allow for the use of native image resolution [7] or different sampling grid sizes [2]. We generalize the concept of scale elasticity to include varying scales of patches sampled from a single image. We measure it by sampling each patch individually and resampling it to the input token's standard 16×16 size. Note that this implementation might introduce *missing data* to the input as shown in Fig. 2.

Positional elasticity constitutes the resilience to positional change in input patches. Vanilla ViT is always trained

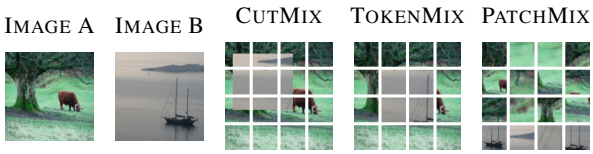


Figure 3. **PatchMix:** We introduce PatchMix, an adaptation of CutMix and TokenMix to elasticity oriented training regime. PatchMix takes full advantage of the ElasticViT position and scale encoding, mixing randomly sampled patches.

with a rigid grid sampling even though, except for standard ViT positional encoding, the architecture does not require this. We measure the positional elasticity by sampling image patches at the dense pixel-level resolution in contrast to the usual coarse grid-level resolution. Note that this sampling procedure might introduce *missing data* as in Fig. 2.

Missing data elasticity represents the resilience to missing image parts at the input. Regular ViT assumes complete knowledge of the image, while many derivative works mask or remove a portion of input tokens [10, 21]. In this work, we extend the concept of missing data to take into account situations where the tokens create an overlap of their receptive fields and do not cover the whole image area at the same time. Such overlapping might occur after patch scales or position perturbations, as shown in Fig. 2. However, to measure the individual impact of missing data on model performance, independent of position and scale change, we drop a subset of grid-aligned patches from the input.

3.2. Protocol

The evaluation pipeline aims to assess the model elasticity by analyzing its resistance to changes in input sampling strategies. For this purpose, we create a set of patches based on the input image and then process this set through three perturbation functions corresponding to three considered elasticities. We use the perturbed set as a transformer input and report its performance. We provide conclusions on model elasticity by analyzing the performance obtained for different types of perturbations.

To strictly define the evaluation pipeline, let us consider image I for which we generate set P of patches $p = (x, y, s)$, where x and y denote the top-left corner's coordinates, and s represents the relative scale (i.e. for a native patch of resolution $r \times r$, we sample a patch $r \cdot s \times r \cdot s$ and rescale it bilinearly to size $r \times r$). Initially, the coordinates x and y are from the regular grid and $s = 1$. However, in the next step, we perturb them with three functions (presented in Fig. 2) corresponding to the considered elasticities:

- $\mathrm{E}_{\mathrm{scale}(s_1,s_2)}(P)$ - introduces the scale perturbations, sampling the s parameter of every patch $p \in P$ independently and uniformly from range $[s_1, s_2]$.
- $\mathrm{E}_{\mathrm{pos}(q)}(P)$ - applies positional perturbation, modifying x and y parameters of each patch $p \in P$, independently moving them by offsets sampled uniformly from range $[-r \cdot q, r \cdot q]$, where r is the size of the patch.
- $\mathrm{E}_{\mathrm{miss}(d)}(P)$ - adds missing data perturbations, dropping out d patches from P randomly with equal probability.

Mathematically, this process can be described as follows:

$$P' = \mathrm{E}_{\mathrm{miss}(d)} \circ \mathrm{E}_{\mathrm{pos}(q)} \circ \mathrm{E}_{\mathrm{scale}(s_1,s_2)}(P).$$

A disturbed set of patches P' is used as an input of the transformer to test its elasticity. Elasticity is high if the predictions for P' are as good as those obtained for P.

4. Elastic ViT

In this section, we introduce a modification of the vision transformer architecture [8] for elastic input.

4.1. Position and scale encoding

Standard ViT [8] implementations utilize learnable positional embeddings. This is feasible because the grid sampling limits the number of possible patch positions. Unfortunately, such embeddings are not attainable with variable position and scale sampling. Each patch can be sampled at an arbitrary pixel position and with an arbitrary scale in elastic sampling. Therefore, the number of possible positions to be encoded is significantly larger. Consequently, we used a four-dimensional modification of the sine-cosine positional encoding of the original transformer model [27]. Let us recall the one-dimensional version of the sine-cosine encoding (l is the length of the embedding and i the i-th element of it):

$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin\left(\mathrm{pos}/10000^{2i/l}\right)$$
$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos\left(\mathrm{pos}/10000^{2i/l}\right).$$

To encode the position of a patch $p = (x, y, s)$, we separately encode the pixel coordinates of the patch upper-left and lower-right corner $(x, y, x + r \cdot s, y + r \cdot s)$ and concatenate resultant embedding vectors.

Finally, we modify the input of a ViT to accept a set of cropped patches and a list of patch coordinates instead of a full image. The coordinates are used to generate positional encoding embeddings for each patch, allowing for continuous positional space.

4.2. Augmented training

We propose a training regime modification for greater elasticity. As our baseline, we use the augmentation regime introduced in [26], as it allows training the model on the

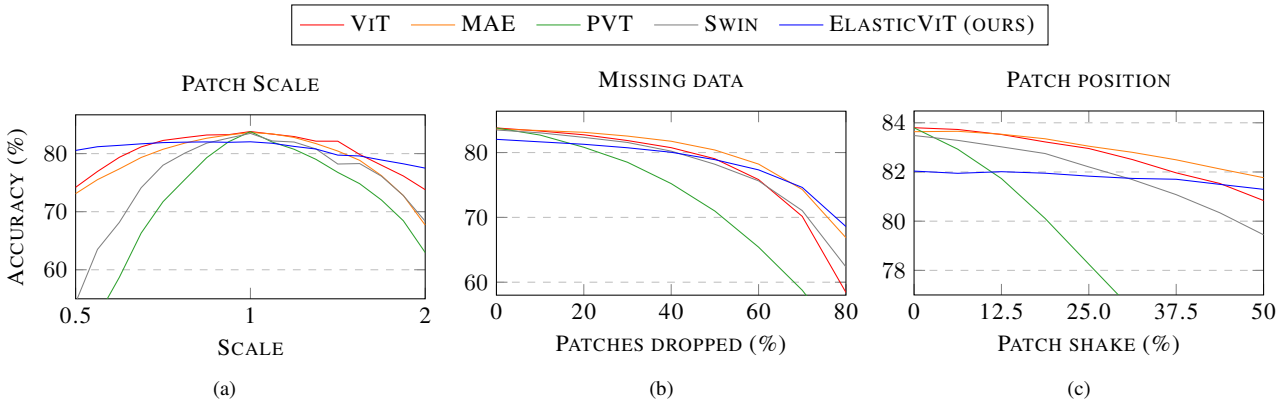


Figure 4. **Isolated perturbations:** **(a)** The impact of changing the patch scale on accuracy. ElasticViT can extract information from patches of various scales, being superior to other models in situations when information is lost either because patches do not cover the whole image or because the patches are of low resolution. **(b)** The effect of random patch dropout on the accuracy. ElasticViT is more resilient to significant patch dropout than the other models and even outperforms the image reconstruction model (MAE) at the end of the measured spectrum. **(c)** The effect of applying positional perturbation on the accuracy. ElasticViT can naturally interpret a whole range of possible position values. Being almost immune to the movement of sampled patches it eventually outperforms all but MAE for large perturbations. Note that the X-axis represents the percentage of patch movement relative to the patch size

ImageNet-1k [23] dataset without using any additional pre-training. We denote a ViT model trained with our modified regime as *ElasticViT* in the following sections. A standard model trained without introduced elasticity is denoted as *ViT* for comparison.

First, we introduce the elasticity functions into the augmentation pipeline. During training we use $\mathrm{E}_{\mathrm{scale}(1/3,1)}$ (random patch sampling size in range 16 to 48), $\mathrm{E}_{\mathrm{miss}(0)}$ (no patch dropout), $\mathrm{E}_{\mathrm{pos}(\infty)}$ (unrestricted random patch positions). The native patch resolution (i.e., the resolutions all patches are rescaled to after sampling) is set to 16×16, and the native (scale = 1) image size is 224×224. We use 448×448 image resolution in the augmentation pipeline to accommodate for variable scale when required.

Second, we observe that the CutMix [34] augmentation used in [26] is not optimal with the input elasticity we introduce. As our sampling strategy may change the proportions of images mixed by CutMix due to patch overlap, we must recalculate the mixed labels after applying elasticity functions to match the actual proportions of mixed images. Therefore, we replace it with PatchMix, a custom but comparable augmentation dedicated to ElasticViT. Similarly to TokenMix [17], it mixes patches after sampling. However, contrary to TokenMix it is not dependant on standard grid sampling, and can take full advantage of the models ability to process arbitrary sampled patches (see Fig. 3 and supplementary materials). Moreover, we modify the MixUp [35] augmentation, performing it after applying perturbations to patch sampling and ensuring that both sequences of patches to be mixed have the same patch scales element-wise.

4.3. Experimental setup

We conduct experiments on a variety of state-of-the-art models with comparable parameter counts, differing in architecture or training policy. Namely, we evaluate our ElasticViT, ViT from DeiT III [26], and MAE [10], all based on ViT-B architecture, and further Swin-B [19], and PVT-v2-B5 [29]. All models were trained on ImageNet-1k [23].

As the tested models significantly differ in architecture, we provide the implementation details of the missing data and position perturbations. The scale perturbation is implemented by patch resampling, independent of the model.

Because ElasticViT, MAE, and DeiT III model build on original [8] ViT architecture, we implement the action of $\mathrm{E}_{\mathrm{miss}}$ by removing the tokens from input entirely. Contrary to Swin and PVT, where attention mechanisms and positional encodings are more involved. For those models, we implement the action of $\mathrm{E}_{\mathrm{miss}}$ as zero-value masking of input tokens.

In all experiments, the magnitude of patch displacement is less than half of the patch size. Therefore, because Swin and PVT use relative positional encoding and $\mathrm{E}_{\mathrm{miss}}$ is implemented via patch masking, we implement the action of $\mathrm{E}_{\mathrm{pos}}$ by stitching sampled and masked patches into an image of the original size. Each patch is aligned to the nearest corresponding position in the grid. ViT from DeiT III and MAE use learned discrete-valued absolute positional encoding. Therefore, we encode the sampled patch position with an embedding corresponding to the position nearest to the patch. ElasticViT accepts continuous-valued positions, therefore we provide the model with true patch coordinates.

The experiments were performed on ImageNet-1k classification task. We report our results with both *classification accuracy* and the *number of tokens* used to achieve given results. All training experiments were run using 4×NVIDIA A100 GPUs. ElasticViT was trained for 800 epochs using the same training parameters as in [26]. For other models we used checkpoints provided by their authors.

To analyses transfer learning capabilities, we fine-tuned the above models for the MS COCO 2014 [16], Pascal VOC 2007 [9] and ColonCancer [24] datasets for 25 epochs. Only the final fully-connected classification head was trained using standard grid sampling as in standard ViT. The training regime from [26] was used, excluding the utilization of MixUp and CutMix augmentations. We report the mean class average precision scores.

5. Research questions

Our goal was to investigate the resilience of transformers to input perturbations. For this purpose, we conducted extensive experiments, divided into six groups. The first three correspond to individual perturbations. The fourth group relates to their combinations. The fifth focuses on fundamental sampling strategies. In the sixth group, we examine a training trade-off between elasticity and base accuracy. Finally, in the last group we look into transferability of resilience to other datasets. All of them are described in the following subsections.

5.1. Scale elasticity

We maintain a consistent grid layout while introducing scale changes to each patch. The number of patches remains unchanged, but the perturbation inherently introduces overlapping or missing data to the input. This process is visually represented in Fig. 2 as *Scale Elasticity*. The outcomes are illustrated in Fig. 4a.

Is ViT robust with respect to scale? All baseline models significantly decrease in accuracy when patch scale changes. Notably, the best-performing baseline model is the original supervised ViT while the worst are PVT and Swin.

Does applying randomized patch sampling in training improve scale elasticity at inference? Ours ElasticViT despite having lower accuracy than other models at the base point (82.04% vs. 83.80% of ViTs), due to high resilience to input scale variations, maintains its performance even at the ends of the measured spectrum and eventually outperforms all the other models.

5.2. Missing data elasticity

Missing data can be simulated by adding perturbations to the input token set so part of the image is not represented by any of the cut patches. The patch scale experiment (Fig. 4a) introduced missing data aspects due to changing the scale of the patches, but it was a byproduct of other types of perturbations. We use a patch dropout scheme to isolate only the missing data aspect and call it the *missing data* experiment. The perturbation is depicted in Fig 2. The results can be seen in Fig. 4b.

How does input dropout affect ViT performance? Introducing dropout results in a consistent decline in the accuracy of all models, being increasingly noticeable as the percentage of dropped tokens increases. The best-performing baseline model is MAE, which is expected as it was trained on an image reconstruction task. The PVT significantly stands out from the other models, even from Swin, for which an identical dropout scheme was implemented and which has sparse attention too.

Do elastic perturbations during training enhance performance when dealing with missing data? ElasticViT exhibits a comparable behavior to the baselines when it comes to missing data, displaying a decline in performance as the token count decreases. Nevertheless, the rate of accuracy loss is gentler. Notably, ElasticViT surpasses MAE performance after the removal of 75% of tokens. This is significant because in the training the ElasticViT always accepted the full token count, contrary to MAE, which was trained to reconstruct the image from the 25% of input tokens.

5.3. Positional elasticity

Elastic vision transformers should not be limited to accepting only rigid sets of patch positions. In *patch position* experiment, we introduce a concept called "patch shake" – a randomized shift in the position of the original patches from a grid by a specified percentage of the patch size. ElasticViT's positional embeddings readily adapt to these positional changes. However, for the other models, we had to modify their positional embeddings to make this experiment viable. We encode each patch with embedding corresponding to the grid layout position being nearest to the patch. The patch shake operation is illustrated in Fig. 2. The outcomes can be observed in Fig. 4c.

Is ViT elastic to positional perturbations? Patch shake results in a minor accuracy reduction for all the models except PVT. It's important to note that patch shake can inherently introduce missing data and overlap perturbations. Therefore, this experiment doesn't isolate which specific perturbation has the most substantial impact on performance. Nonetheless, it's worth considering that the classification task might be inherently position agnostic, as ImageNet classification could be efficiently accomplished by treating patches as "bags of words" without explicit positional context.

Does training with randomized sampling improve positional elasticity? Like the missing data experiment (Sec. 5.2), ElasticViT exhibits greater resilience to patch

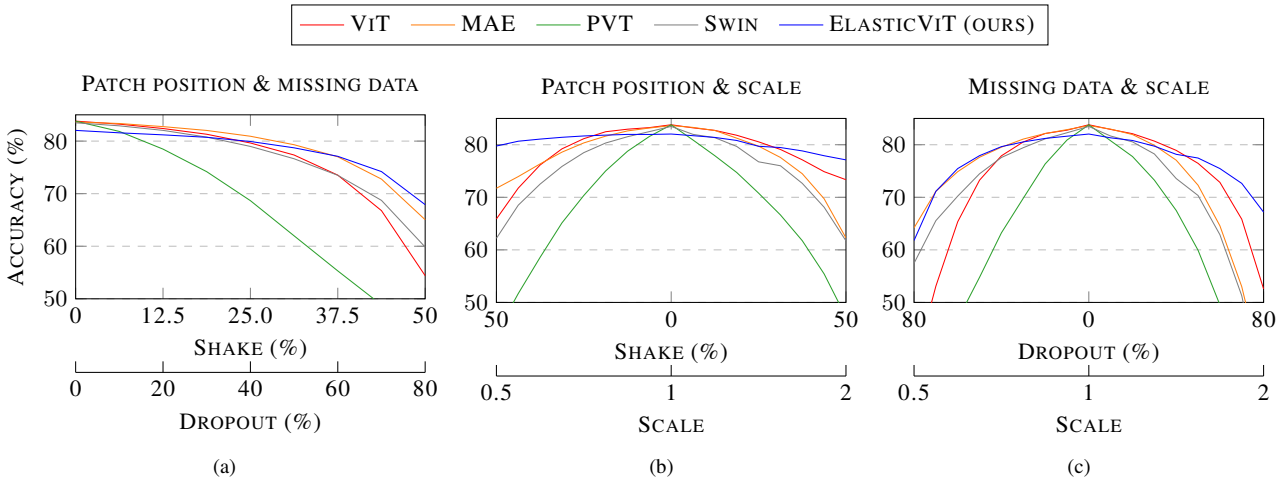


Figure 5. **Combining perturbations:** The results of mixing multiple types of perturbations in one experiment. We observe that the performance of baseline models degrades quickly as perturbations add up. At the same time, the accuracy of ElasticViT remains stable for much longer, allowing more elastic input.

Figure 6. **Patch sampling strategies:** We introduce two sampling strategies to show the benefits of elastic sampling. CENTRAL employ human retina-like higher density sampling in the center, EDGE samples based on weights provided by an edge detector (the more edges the denser sampling).

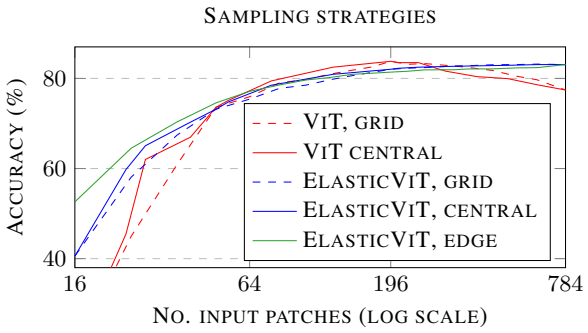


Figure 7. **Central and adaptive patch sampling:** Results of applying two patch sampling strategies (CENTRAL and EDGE) compared to standard grid sampling at inference. We observe that by leveraging input elasticity we can gain a significant improvement in accuracy when the number of sampled patches is limited.

shake than the baselines. ElasticViT outperforms all models but MAE when the patch positions are shaken by approximately 40% of the patch size. Notably, the accuracy of ElasticViT remained almost constant with respect to the increase of patch shake, while the second-best model in this regard (MAE) lost 2 p.p. of accuracy.

5.4. Combining perturbations

We conduct a series of tests by combining the previously introduced input perturbations to assess their collective impact on performance. The scaling ranges for perturbation parameters are the same as in the earlier experiments. The objective is to determine whether ElasticViT's resilience will continue to outperform that of the original ViT in more complex scenarios.

How does positional elasticity combine with missing data elasticity? In *patch position & missing data* experiment, we introduce a combination of dropout and shake perturbations, and the results are depicted in Fig. 5a. Notably, the detrimental effect on ViT's performance closely aligns with the summed impact of shake and dropout when performed independently. This lets ElasticViT surpass MAE at the 60% dropout threshold, instead of 75% as in the *missing data* 4b experiment.

What is the interaction between positional elasticity and scale data elasticity? In *patch position & patch scale* experiment, we combine the zoom perturbation with the shake perturbation. The results are shown in Fig. 5b. The ratio of performance decline between ViT and ElasticViT remained consistent with that reported in the previous experiment for patches twice as large. However, as patches

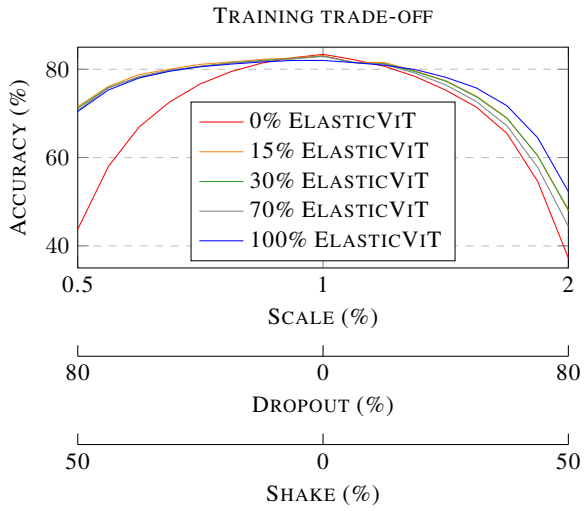


Figure 8. **Training for elasticity:** Evaluation of training regimes with different proportions of the grid and ElasticViT augmented training (see Sec. 4) in training batches. For validation we apply all three perturbations. Note that only a small number of ElasticViT augmented training batches ($< 15\%$) is required to improve elasticity significantly.

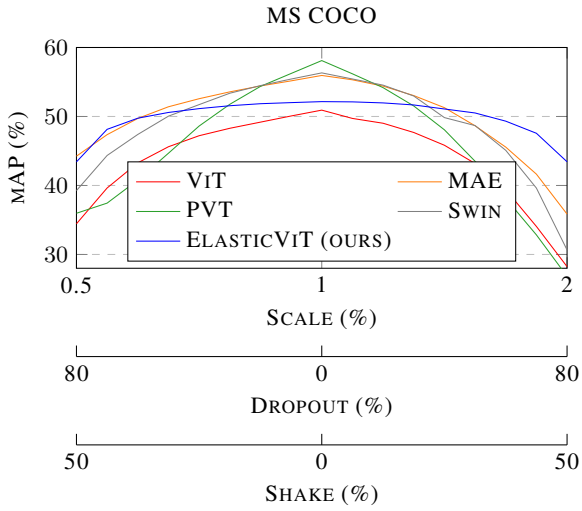


Figure 9. **Transfer learning:** Results of fine tuning the last layers of models for multi-label classification of the MS COCO 14 dataset. All models were trained with standard grid sampling. We observe, that for high perturbation rate ElasticViT outperforms baselines.

became smaller, the performance gap between both algorithms widened. In the most altered scenario, ElasticViT outperforms ViT with a nearly 20 percentage point lead.

How does scale elasticity combine with missing data elasticity? In *missing data & patch scale* experiment, we combine dropout with patch scale change, and the results are illustrated in Fig. 5c. The outcomes resemble those of the grid zoom experiment (Sec. 5.1) but exhibit a more pronounced negative impact of the perturbations. ElasticViT begins to outperform ViT at approximately the midway point of the perturbation intensity scale.

5.5. Patch sampling strategies

The elastic features demonstrated by ElasticViT create an opportunity to apply a multi-scale approach to partitioning images into patches of different scales in a non-uniform manner. Given the results obtained in the grid experiments, there's a possibility that this approach can lead to performance improvement without sacrificing accuracy compared to standard grid sampling (referred to as *GRID*).

First, we test a hypothesis that, for the ImageNet, the central portion of an image might carry more significance than its periphery. To explore this, we iteratively divide the patches closest to the center of the image into four smaller patches. We refer to this algorithm as *CENTRAL*, see Fig. 6. The rationale is that smaller patches offer a higher sampling resolution and introduce more data into the input, potentially leading to improved accuracy. The CENTRAL algorithm is designed so that the neutral point corresponds to a token count of 196, which aligns with the original ViT grid of 14×14 and a single scale. Lower values in this context result in larger patches near the image periphery, while higher values position smaller patches toward the center.

Next, we extend on the previous experiment, analyzing if an adaptive patch sampling strategy can improve the results, especially in low patch count scenarios. We create a toy adaptive sampling algorithm, which we refer to as *EDGE*, based on the Canny edge detector [3]. The method starts by computing a bitmap of edges with a large σ value for the detector. The map is then divided into 112×112 patches, which are then sorted by their sums over the bitmap. Then, the algorithm iteratively divides a patch with the highest sum larger than the minimal size of 8×8, into 4 new patches, which are added to the list, preserving the order. The algorithm is repeated until the number of patches reaches the target. Visualization of EDGE is presented in Fig. 6.

Does using higher resolution of the grid in the center improve accuracy? Our findings for the CENTRAL algorithm are presented in Fig. 7. In the case of the vanilla ViT model, we observed issues with resilience to scale changes. Depending on the non-standard scale content of the input dataset, ViT results can be either negatively or positively influenced by the CENTRAL algorithm. Conversely, when considering ElasticViT, the differences are significantly reduced, and there is a higher likelihood that CENTRAL sampling leads to improved model performance.

Can adaptive patch size sampling perform better? The results of this experiment are presented in Fig. 7 for the ElasticViT model. We observe, that EDGE performs significantly better than both GRID and CENTRAL algorithms in low patch count scenarios (less than 64 patches). When limited to 25 patches, it achieves over 6% improvement compared to ElasticViT and 20% over standard ViT, increasing to 12% and 32% (accordingly) when only 16 patches can be sampled. Consequently, we claim that even very simple adaptive patch sampling methods can reduce the number of tokens needed to achieve targeted performance.

5.6. Trade-offs

This section explores a training trade-off for elastic sampling. Further experiments on trade-off in inference are provided in supplementary materials.

What is the tradeoff between accuracy and elastic training? A consistent pattern emerges in all our experiments comparing ViT and ElasticViT: ViT generally exhibits higher accuracy than ElasticViT when evaluated under the original, unperturbed grid sampling scenario. However, as we introduce perturbations into the evaluation, ElasticViT can surpass ViT. Earlier we performed experiments that juxtapose classical single-scale grid training against a fully randomized sampling method of ElasticViT, representing two extremes in the spectrum.

Fig. 8 illustrates the results of training with mixed simple grid sampling (standard ViT training) and a fully randomized setup (ElasticViT training regime, see 4). The evaluation was conducted by applying all three perturbation variants at the same time.

The findings reveal that for simple grid evaluation, elastic training has a relatively minor negative impact on accuracy. However, in the case of randomized evaluation that requires elasticity, even with only 15% of the training data containing randomized patches, there is a substantial gain of more than ten percentage points in accuracy. This implies that in practical applications, fine-tuning the network can be accomplished using just a fraction of perturbed input data while having minimal repercussions on baseline (simple grid) accuracy. Yet, this approach offers increased resilience to perturbations, demonstrating the practical utility of elastic inputs in the training process.

5.7. Transfer Learning

We evaluate elasticity of models pre-trained on the ImageNet-1k dataset and fine-tuned to the target tasks.

Does elasticity capabilities transfer to other datasets? An important question that remains to be answered is whether the elasticity capabilities observed on the ImageNet-1k dataset transfer to other datasets and tasks. We analyze this problem on the matter of multi-label classification of the MS COCO 2014 dataset as described in Sec. 4.3. We perform evaluation applying all three perturbations (see Sec. 3) at the same time. Results of this experiment are presented in Fig. 9, PVT, Swin and MAE perform the best when no perturbation are introduced. However, ElasticViT exhibits the best consistency in results across all perturbation range. The standard ViT model performs the worst, even without any perturbations, indicating worse generalization capabilities. Results for VOC and ColonCancer are provided in supplementary materials.

6. Conclusions

This study examines Vision Transformers (ViT) and their adaptability to varying input sampling strategies required in real-world applications. We introduce an evaluation protocol to assess ViT's resistance to input perturbations, including scale, missing data, and positional changes.

Standard ViT models prove to be vulnerable to these perturbations, displaying a significant performance drop due to potential overfitting during training. In contrast, the modified ViT we proposed exhibits better resilience thanks to randomized training, maintaining or improving performance in various scenarios, see overall comparison in Fig. 8 of the supplementary material.

Our experiments also explore adaptive patch sampling using CENTRAL and EDGE algorithms, which ElasticViT benefits from, particularly in scenarios with fewer patches. Adaptive patch sampling efficiently reduces token requirements while preserving target performance.

Future work in this area should focus on refining the adaptive patch sampling strategies and further investigating the trade-offs between downscaling input tokens and patch dropout under computational constraints. Additionally, research should aim to apply these findings to real-world applications, enhancing transformer models' adaptability in practical contexts.

Acknowledgments

This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237, and by National Science Centre, Poland (grant no. 2023/49/N/ST6/02465, 2022/47/B/ST6/03397, 2022/45/B/ST6/02817, and 2023/50/E/ST6/00469). Some experiments were performed on servers purchased with funds from a grant from the Priority Research Area (Artificial Intelligence Computing Center Core Facility) under the Strategic Programme Excellence Initiative at Jagiellonian University. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017483.

References

[1] Hassan Akbari, Liangzhe Yuan, Rui Qian, et al. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *NeurIPS*, 34:24206–24221, 2021. 1

[2] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14496–14506, 2023. 1, 2

[3] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, pages 679–698, 1986. 7

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, et al. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 1, 2

[5] Wuyang Chen, Wei Huang, Xianzhi Du, et al. Auto-scaling vision transformers without training. In *ICLR*, 2022. 1, 2

[6] Xiangxiang Chu, Zhi Tian, Bo Zhang, et al. Conditional positional encodings for vision transformers. In *ICLR*, 2022. 2

[7] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. In *NeurIPS*, volume 36, 2024. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1, 2, 3, 4

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009. 5

[10] Kaiming He, Xinlei Chen, Saining Xie, et al. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 2, 3, 4

[11] Abhishek Jha, Soroush Seifi, and Tinne Tuytelaars. Simglim: Simplifying glimpse based active visual reconstruction. In *WACV*, pages 269–278, 2023. 2

[12] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, pages 5148–5157, 2021. 2

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. Segment anything. *arXiv:2304.02643*, 2023. 1, 2

[14] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, et al. Efficient self-supervised vision transformers for representation learning. In *ICLR*, 2021. 2

[15] Hanting Li, Mingzhe Sui, Feng Zhao, et al. Mvt: mask vision transformer for facial expression recognition in the wild. *arXiv:2106.04520*, 2021. 2

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 5

[17] Jihao Liu, Boxiao Liu, Hang Zhou, Hongsheng Li, and Yu Liu. Tokenmix: Rethinking image mixing for data augmentation in vision transformers. In *European Conference on Computer Vision*, pages 455–471. Springer, 2022. 4

[18] Yue Liu, Christos Matsoukas, Fredrik Strand, et al. Patchdropout: Economizing vision transformers using patch dropout. In *WACV*, pages 3942–3951, 2023. 2

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2, 4

[20] Adam Pardyl, Grzegorz Rypeść, Grzegorz Kurzejamski, et al. Active visual exploration based on attention-map entropy. In *IJCAI*, pages 1303–1311, 2023. 1, 2

[21] Yongming Rao, Wenliang Zhao, Benlin Liu, et al. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 1, 3

[22] Tomer Ronen, Omer Levy, and Avram Golbert. Vision transformers with mixed-resolution tokenization. In *CVPR*, pages 4613–4622, 2023. 2

[23] Olga Russakovsky, Jia Deng, Hao Su, et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 4

[24] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging*, 35(5):1196–1206, 2016. 5

[25] Yehui Tang, Kai Han, Yunhe Wang, et al. Patch slimming for efficient vision transformers. In *CVPR*, pages 12155–12164, 2022. 1, 2

[26] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, pages 516–533. Springer, 2022. 2, 3, 4, 5

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. *NeurIPS*, 30, 2017. 3

[28] Wenhai Wang, Enze Xie, Xiang Li, et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 2

[29] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022. 4

[30] Chao-Yuan Wu, Ross Girshick, Kaiming He, et al. A multigrid method for efficiently training video models. In *CVPR*, pages 153–162, 2020. 2

[31] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, et al. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *CVPR*, pages 13587–13597, 2022. 1

[32] Hongxu Yin, Arash Vahdat, Jose Alvarez, et al. AdaViT: Adaptive tokens for efficient vision transformer. In *arXiv:2112.07658*, 2021. 1, 2

[33] Jiahui Yu, Zirui Wang, Vijay Vasudevan, et al. Coca: Contrastive captioners are image-text foundation models. *arXiv:2205.01917*, 2022. 1

[34] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *CVPR*, pages 6023–6032, 2019. 4

[35] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 4

[36] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, et al. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *ICCV*, pages 2998–3008, 2021. 2

Supplementary material for
Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers

Adam Pardyl[1,2,3] Grzegorz Kurzejamski[1] Jan Olszewski[1,4]
Tomasz Trzciński[1,5,6] Bartosz Zieliński[1,2]
[1]IDEAS NCBR
[2]Jagiellonian University, Faculty of Mathematics and Computer Science
[3] Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4] University of Warsaw [5] Warsaw University of Technology [6] Tooploox

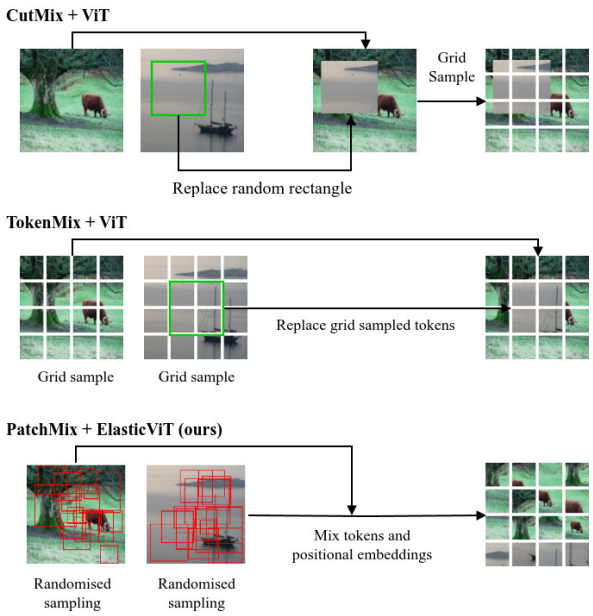


Figure 1. **PatchMix:** Standard CutMix poses an issue in ElasticViT training regime, as proportions of mixed targets may change in randomized sampling. TokenMix replaces patches after sampling, incidentally eliminating this issue, but nevertheless relies on grid sampling. Our PatchMix takes full advantage of the ElasticViT position and scale encoding, and enables mixing of randomly sampled patches of different scales.

1. Additional experiments

1.1. PatchMix ablation

To evaluate the effectiveness of PatchMix (see Fig. 1 for visualization) we perform an ablation study, training the model with the regime presented in Sec. 4 of the main paper, but without the PatchMix augmentation applied. The results are presented in Tab. 1. The PatchMix augmentation improves the accuracy of ElasticViT by over 1.5% on ImageNet-1k.

ElasticViT	Accuracy
without PatchMix	80.67%
with PatchMix	82.04%

Table 1. **PatchMix ablation study:** The effectiveness of the PatchMix augmentation evaluated on the Imagnet-1k dataset. The test was performed without any introduced perturbations. We observe that PachMix provides over 1.5% gain in accuracy.

1.2. Grid density

Continuing on scale elasticity experiments presented in Sec. 5.1, we decided to investigate the resistance of ViT architectures to change of grid density. In real-world scenarios, scale changes are quite common. However, when conducting synthetic experiments with ViT, the typical approach is to maintain a consistent input image resolution and grid layout density.

In this setup, we adjust the grid density, thereby altering the number of patches while using the same input image. This modification process is illustrated in Fig. 2 as *Grid density* and compared to the *Grid zoom* perturbation shown in Sec. 5.1. The outcomes for standard ViT, MAE and ElasticViT are presented in Fig. 3. PVT and Swin models were omitted in this evaluation, as changing the grid density in those models is not trivial due their internal structure. We observe, that all models perform similarly well when decreasing the density of the grid, while only ElasticViT can utilize denser sampling to its advantage.

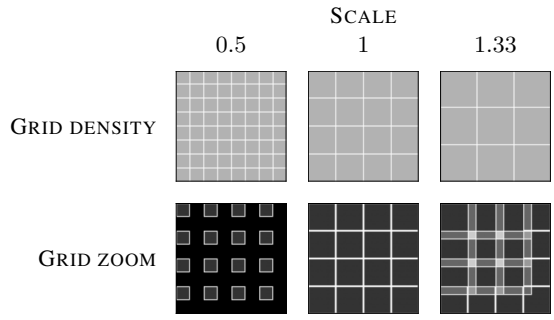


Figure 2. **Scale elasticities:** We use two perturbation scenarios, that change the size of a patch in a grid. The first (grid density) changes the number of patches. The second (grid zoom) keeps same number of patches but changes theirs size.

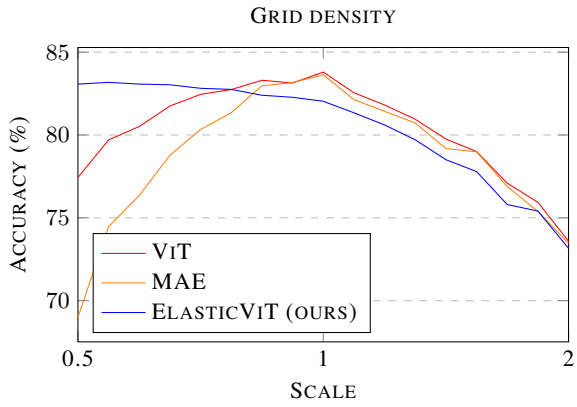


Figure 3. **Grid density**: The impact of changing the grid density (see Fig. 2) on accuracy. ElasticViT can utilize extra information from denser grid sampling (0.5), outperforming the original ViT.

1.3. Is it better to down-scale input or dropout patches?

The results from our previous experiments naturally lead to an important question: When faced with computational constraints, is it more beneficial to remove an entire patch from the input or to rescale neighboring patches in order to maintain complete image coverage, even at the cost of changing the token scale? To investigate this, we conduct experiments where we iteratively select a random 2×2 patch block from the uniform 14×14 patch grid and modify it in two ways. The first is to replace a 2×2 block of patches with a single larger patch, preserving the same coverage. In the second, we remove three out of four patches within the block. In both cases, these operations reduced the input token set by three, either through a dropout operation or by changing the scale of the patches.

The results of these experiments are depicted in Fig. 4. Surprisingly, ElasticViT exhibits almost no discernible difference in performance between the dropout and rescale operations, with dropout slightly outperforming rescaling at the extremes. Again, This would suggest a potential overfit with results better when having only 25% of the image covered by patches than having 100% coverage but with two times lower sampling resolution. In contrast, for the original ViT model, there is a visible distinction in performance between the two methods of limiting token counts, with the rescaling option proving to be a much more effective solution at the extremes.

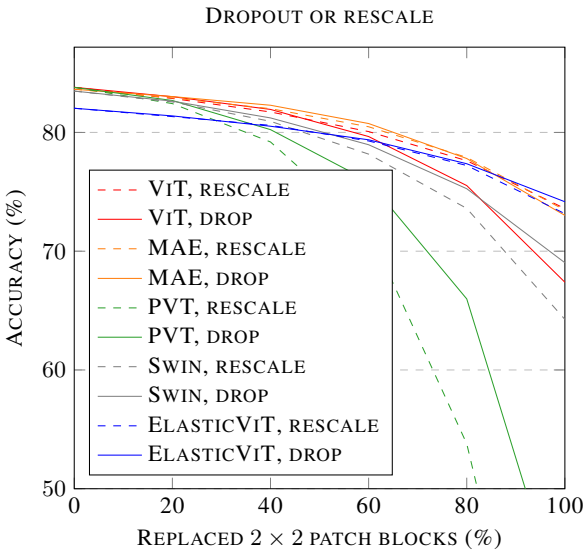


Figure 4. **Reducing number of patches:** Evaluation of the trade-off between lower resolution sampling and patch dropout. The X-axis represents the number of 2×2 patch blocks either replaced by a single lower-resolution patch or kept with only one patch while dropping the rest. We observe that the chosen strategy only affects ViT significantly at a high number of replacements, whereas ElasticViT remains unaffected due to its greater elasticity in handling missing data.

1.4. Transfer learning (continued)

1.4.1 Pascal VOC dataset

In this section we show additional results for transfer learning elasticity, evaluated on the Pascal VOC dataset. We follow the same training and evaluation setup as in Sec. 5.7 of the main paper. Results of the experiments are presented in Fig. 5. We observer, that standard ViT performs the best for the native resolution, but is outperformed by ElasticViT when elasticity is introduced. Both PVT and Swin performs slightly worse, and surprisingly, the self-supervised pre-trained MAE performs the worst, failing to achieve event half of the mean average precision score of ViT and ElasticViT.

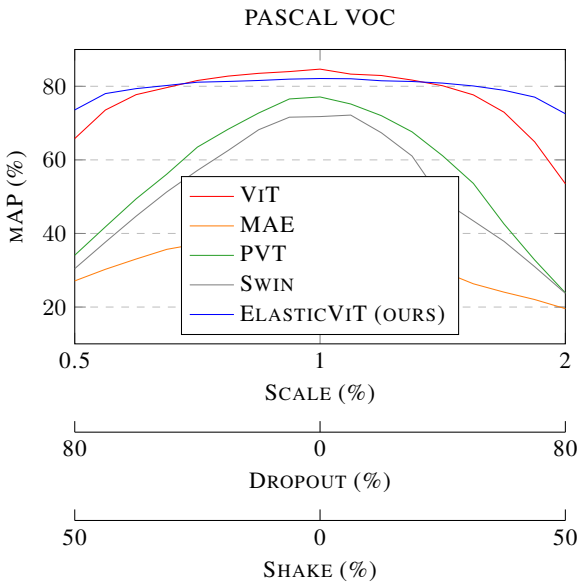


Figure 5. **PASCAL VOC transfer learning**: Results of fine tuning the last layers of models for multi-label classification of the PASCAL VOC 2007 dataset. All models were trained with standard grid sampling. We observe, that for high perturbation rate ElasticViT outperforms baselines. Note the surprisingly poor performance of the self-supervised pre-trained MAE.

Model	Sampling type	Accuracy
ViT	GRID	82%
ElasticViT	EDGE	87%

Table 2. **ColonCancer transfer learning:** Results of fine-tuning the last layers of models for binary classification of the ColonCancer dataset. The standard ViT model was run with grid sampling, while our ElasticViT model utilized EDGE sampling as described in the main paper. We observe that our model outperforms the standard ViT, benefiting from variable scale sampling.

1.4.2 ColonCancer dataset

We further test transfer learning properties of our model, evaluating it on the ColonCancer dataset. The dataset consists of histopathological images to be classified as malignant or benign. As previously, we train only the final linear layer of the model. For standard ViT we apply regular grid sampling, our ElasticViT uses EDGE sampling as described in the main paper. Results are presented in Tab. 2, showing superior performance of ElasticViT. We attribute this superior performance to EDGE sampling, which extracts more patches in regions containing cell nuclei, enabling ElasticViT to create a better representation of the data.

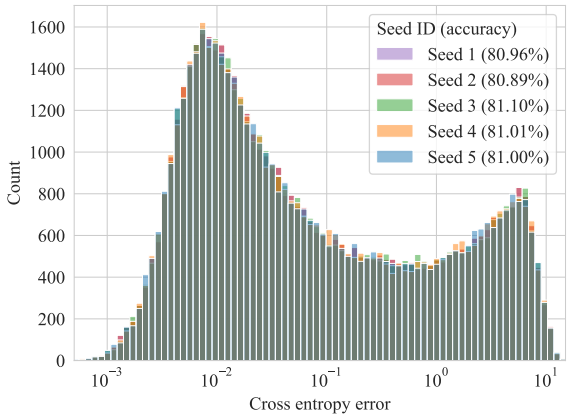


Figure 6. **Elastic sampling impact on performance across multiple runs:** The histogram represents the distribution of classification error for five inference runs with different random seeds used for elastic sampling. We observe that the differences between runs are insignificant. Note that the accuracy scores of particular runs are provided in the plot legend.

1.5. Stability of elastic evaluation

As our elasticity benchmark introduces randomness, we run the evaluation pipeline on the same trained model multiple times to check for any fluctuations in performance across different random seeds. The differences in performance are insignificant over multiple runs of elastic sampling, as shown in the histogram of performance standard deviations in Fig. 6. This consistency can be attributed to the large size of the ImageNet-1k validation set.

1.6. Patch redundancy

Elastic sampling allows for patch overlap, which provides redundant information to the model. In this experiment, we explored vision transformer capability to deal with redundant information. The model was provided with a standard full grid of 16×16 patches, that covered the entire image. Then, a number of redundant, randomly sampled patches with scales between 0.5 and 2 was added to the input. The results are presented in Fig. 7. We observe, that for standard ViT and MAE models those redundant patches essentially constitute noise, which reduces the overall performance. Our ElasticViT is capable of using the extra information to slightly increase the accuracy.

1.7. Overall performance comparison

To assess the overall elasticity of the compared methods, in Fig. 8 we present a critical difference diagram, aggregating results from Fig. 4 and Fig. 5 of the main paper.

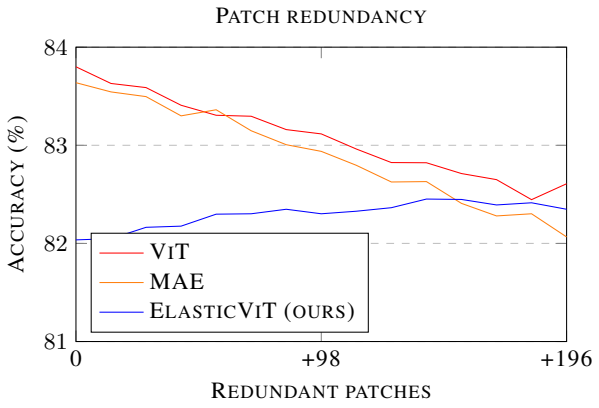


Figure 7. **Patch redundancy**: The impact of adding randomly sampled patches in addition of a standard sampling grid. We observe, that standard ViT and MAE lose performance, as those randomly sampled patches as treated as noise. In contrast, ElasticViT can utilize the extra information to improve the result.

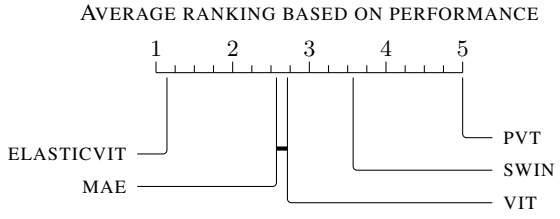


Figure 8. **Critical difference diagram**: Average performance ranking of ElasticViT and baseline methods for ImageNet with high perturbation (most extreme settings). ElasticViT significantly outperforms remaining approaches, while more structured methods (PVT and Swin) perform significantly worse than simple ViT. Moreover, the difference between ViT and MAE is insignificant. Notice that this diagram was generated for rankings generated based on results from Fig. 4 and Fig. 5 of the main paper.

2. Pipeline visualizations

In Fig. 9 we present visualization of our elasticity pipeline output. For clarity, patches are separated with red borders and patch overlap is highlighted in bright colors.

3. Theoretical analysis of input sampling strategies and their impact on positional embedding

3.1. Definition recall

To strictly define the evaluation pipeline, let us consider image I for which we generate set P of patches $p = (x, y, s)$, where x and y denote the top-left corner's coordinates, and s represents the relative scale (i.e. we sample a patch $r \cdot s \times r \cdot s$ and rescale it bilinearly to size $r \times r$). Initially, the coordinates x and y are from the regular grid and $s = 1$. However, in the next step, we perturb them with three functions corresponding to the considered elasticities:

- $\mathrm{E}_{\mathrm{scale}(s_1,s_2)}(P)$ - introduces the scale perturbations, sampling the s parameter of every patch $p \in P$ independently and uniformly from range $[s_1, s_2]$.
- $\mathrm{E}_{\mathrm{miss}(d)}(P)$ - adds missing data perturbations, dropping out d patches from P randomly with equal probability.
- $\mathrm{E}_{\mathrm{pos}(q)}(P)$ - applies positional perturbation, modifying x and y parameters of each patch $p \in P$, independently moving them by offsets sampled uniformly from range $[-r \cdot q, r \cdot q]$, where r is the size of the patch.

The patches are described by their upper left (x, y) and lower right corner $(x + rs, y + rs)$. Then each coordinate $\mathrm{pos} \in \{x, y, x + rs, y + rs\}$ is encoded by the sinusoidal positional encoding:

$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin\left(\frac{\mathrm{pos}}{10000^{2i/l}}\right)$$
$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos\left(\frac{\mathrm{pos}}{10000^{2i/l}}\right).$$

which are concatenated into a single vector.

3.2. Perturbations influence on positional embedding

The application of $\mathrm{E}_{\mathrm{miss}}$ does not affect the positional encoding of a patch. However, if a patch was not dropped by the $\mathrm{E}_{\mathrm{miss}}$ perturbation, then the other two perturbations can modify its position embedding.

After application of $\mathrm{E}_{\mathrm{pos}}$, the values of x and y get updated to $x+\Delta x$, $y+\Delta y$, while the lower right corner $x+rs$ and $y+rs$ get updated to values $x+rs+\Delta x$ and $y+rs+\Delta y$. The offset values Δx and Δy do not depend on x nor y. Thus, for x, we obtain the following positional embedding

$$\mathrm{PE}_{(x+\Delta x,2i)} = \sin\left(\frac{x}{10000^{2i/l}} + \frac{\Delta x}{10000^{2i/l}}\right)$$
$$\mathrm{PE}_{(x+\Delta x,2i+1)} = \cos\left(\frac{x}{10000^{2i/l}} + \frac{\Delta x}{10000^{2i/l}}\right),$$

which holds analogously for the other three coordinates $\{y, x + rs, y + rs\}$.

When it comes to $\mathrm{E}_{\mathrm{scale}}$, it modifies the s value to $s + \Delta s$ therefore $\mathrm{pos} \in \{x, y\}$ remains unchanged, while both coordinates $\mathrm{pos} \in \{x + rs, y + rs\}$ are offset by the value $\Delta = r\Delta s$. The formula is analogous to the $\mathrm{E}_{\mathrm{pos}}$ case.

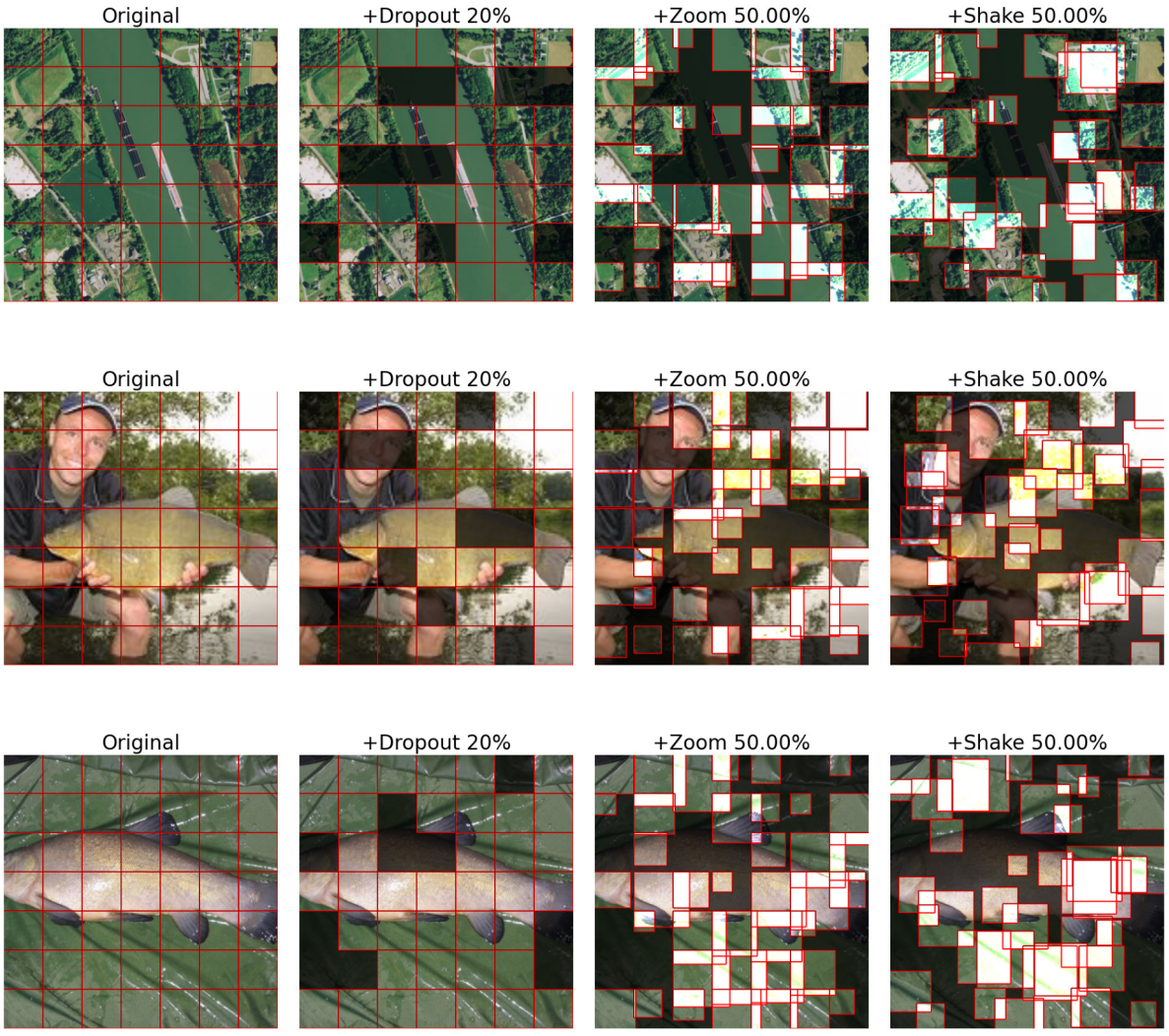


Figure 9. **Elasticity pipeline visualization:** In this figure, we present visualizations of successive input data perturbations. For clarity, patch borders are marked in red, and patch overlap is highlighted.

AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale

Adam Pardyl[1,2,3], Michał Wronka[2], Maciej Wołczyk[1], Kamil Adamczewski[1], Tomasz Trzciński[1,4,5], and Bartosz Zieliński[1,2]

[1] IDEAS NCBR
{adam.pardyl, maciej.wolczyk, kamil.adamczewski, tomasz.trzcinski, bartosz.zielinski}@ideas-ncbr.pl
[2] Jagiellonian University, Faculty of Mathematics and Computer Science
michal.wronka@student.uj.edu.pl
[3] Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4] Warsaw University of Technology
[5] Tooploox

Abstract. Active Visual Exploration (AVE) is a task that involves dynamically selecting observations (glimpses), which is critical to facilitate comprehension and navigation within an environment. While modern AVE methods have demonstrated impressive performance, they are constrained to fixed-scale glimpses from rigid grids. In contrast, existing mobile platforms equipped with optical zoom capabilities can capture glimpses of arbitrary positions and scales. To address this gap between software and hardware capabilities, we introduce AdaGlimpse. It uses Soft Actor-Critic, a reinforcement learning algorithm tailored for exploration tasks, to select glimpses of arbitrary position and scale. This approach enables our model to rapidly establish a general awareness of the environment before zooming in for detailed analysis. Experimental results demonstrate that AdaGlimpse surpasses previous methods across various visual tasks while maintaining greater applicability in realistic AVE scenarios.

Keywords: Active visual exploration · Vision transformers · Reinforcement learning

1 Introduction

Common machine learning solutions for computer vision tasks, such as classification, segmentation, or scene understanding, usually presume access to complete input data [13]. However, this assumption does not apply to embodied agents functioning in the real world. Agents such as robots and UAVs face constraints on their data-gathering capabilities, such as a restricted field of view and limited operational time, caused by dynamically changing environments [48]. Moreover, capturing and analyzing high-resolution images of the entire visible area is inefficient, as not every part of an image contains the same amount of details.

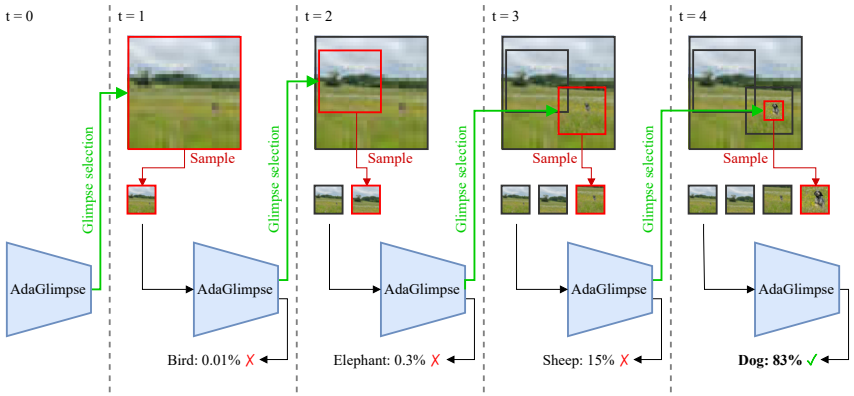


Fig. 1: Adaptive Glimpse (AdaGlimpse): Our approach selects and processes glimpses of arbitrary position and scale, fully exploiting the capabilities of modern hardware. In this example, AdaGlimpse selects a low-resolution glimpse of the whole environment. Based on this glimpse, it predicts a bird with probability 0.01, too low to make the final decision. Instead, it selects the second glimpse by zooming in to the upper left corner. The process repeats four times until the probability of the predicted class is higher than a specified threshold.

Active Visual Exploration (AVE) addresses the challenge of how an agent should select visual information from its environment to achieve a particular objective. Instead of systematically sampling and analyzing the entire available environment at the highest resolution, an agent dynamically chooses the location for sampling subsequent observations, informed by insights from prior exploration steps [33]. This process of selecting visual samples, often referred to as *glimpses*, is inspired by the natural way humans explore their surroundings by instinctively moving their heads and eyes [17].

Current research in active visual exploration can be categorized into two groups. The first group of approaches divides the image into a regular grid of fixed-sized glimpses from which the model tries to pick the most informative ones [32, 35, 39, 41]. The second group starts by capturing a low-resolution image of the entire environment, and then it again selects glimpses from regular grids [12, 30, 46]. Relying solely on regular grids fails to fully exploit the capabilities of modern hardware, which can provide a glimpse of any position and scale. For example, in a pan-tilt-zoom camera, we can achieve it by using optical zoom [11], while in a UAV, we can alter its altitude [22].

In this paper, we overcome current limitations by introducing **AdaGlimpse** (Adaptive Glimpse)[6], an active visual exploration method that selects glimpses of arbitrary scale and position, significantly reducing the number of observations required to understand the environment. Drawing inspiration from [31], we build our network on an input-elastic vision transformer. In each exploration step, our

[6] Source code is available at `https://github.com/apardyl/AdaGlimpse`

model predicts the optimal position and scale for the next glimpse as a value in a continuous space. Since the patch-sampling operation is not differentiable, we train the model using a reinforcement learning algorithm. In particular, we use the Soft Actor-Critic algorithm [16] since it excels in exploration tasks.

Through exhaustive experiments, we show that AdaGlimpse outperforms state-of-the-art methods on common benchmarks for reconstruction, classification, and segmentation tasks. As such, our method enables more effective utilization of embodied platform capabilities, leading to faster environmental awareness. Our contributions can be summarized as follows:

- We introduce a novel approach to Active Visual Exploration (AVE) that selects and processes glimpses of arbitrary position and scale.
- We present a task-agnostic architecture based on a visual transformer.
- We formulate AVE as a Markov Decision Process with a carefully designed observation space, and leverage the Soft Actor-Critic reinforcement learning algorithm that excels in exploration.

2 Related work

Missing data. The problem of missing data in context of images has been addressed in a variety of ways, such as inferring remaining information from the input distribution using a fully connected network [42], or more commonly by image reconstruction. In particular, MAT [25] performs inpainting using a transformer network with local attention masking, a solution that additionally reduces computation by processing only informative parts of the image. A similar principle can be found in ViT based Masked Autoencoder (MAE) [18] where the encoder network operates only on visible patches, while the decoder processes all patches, including the masked ones.

Region selection. Numerous methods exist for selecting the most informative regions from an image, including expectation maximization [36, 51], majority voting [2], wake-sleep algorithm [4], sampling from self-attention or certainty maps [32, 39, 41], and Bayesian optimal experiment design [34]; yet, recently the most predominant solution is using reinforcement learning algorithms, such as variants of the Policy Search [28, 29, 50] Deep Q-Learning [6, 7] or Actor-Critic [35].

Variable scale transformers. A number of studies have been conducted to overcome the constraint of Vision Transformers (ViTs) of working only with rigid grid of fixed size patches, be it by modifying grid scale sampling during training phase [24, 45, 49] or with position and patch encoding rescaling tricks [5]. Beyond Grids [31] interests us in particular, as it equips ViT with the ability to use any square present in an image as a patch, removing both grid and size limitation.

Active visual exploration. The SLAM (simultaneous localization and mapping) challenge is often described in the context of active exploration [33]. A popular approach seen in many models [3, 12, 30, 46] is to feed the model a low-resolution version of the image and use a variation of a policy gradient algorithm to choose parts of an image to focus on. Many notable works in domain of AVE are using CNN-based attention maps for glimpse selection [39–41, 47]. Simglim [21] introduces a MAE-based model with an additional glimpse decision neural network to solve image reconstruction tasks. AME [32] similarly uses MAE as a backbone, but makes decisions solely on the basis of attention maps without added loss or modules. STAM [35] uses a visual transformer and a one-step actor-critic for choosing glimpse locations in the classification task.

3 AdaGlimpse

The key idea of Adaptive Glimpse (AdaGlimpse) is to let the agent select both the scale and position of each successive observation (*glimpse*) from a continuous action space. This way, the agent can learn to decide whether, at a given step of the exploration process, it is preferable to sample a wider view field at a lower relative resolution, zoom in on detail to capture a small high-resolution glimpse, or choose a midway solution.

In this section, we start by formalizing the concept of adaptive glimpse sampling (see Sec. 3.1), and then we discuss the main two components of AdaGlimpse presented in Fig. 2. In Sec. 3.2, we describe a vision transformer encoder with variable scale sampling and a task-specific head. In Sec. 3.3, we present the reinforcement learning agent based on the Soft Actor-Critic (SAC) algorithm.

3.1 Adaptive glimpse sampling

Glimpses. Let X be an unobserved scene to explore. We assume X to be a rectangle within the Cartesian coordinate system. A *glimpse* G is a square region within X that can be observed by a camera, specified by the position of its top-left corner (x, y) and its size d (camera field of view). Furthermore, we define glimpse scale as: $z = (d - d_{\min})/(d_{\max} - d_{\min})$, where $d_{\max}$ and $d_{\min}$ are constants denoting the maximum and minimum field of view of an agent camera. Intuitively, a scale of 0 corresponds to the maximum camera zoom level and a scale of 1 to the widest view possible. Finally, let $C = (x, y, z)$ be the coordinates of glimpse G and constant $d_{\text{cam}} \times d_{\text{cam}}$ denote the sampling resolution of G, i.e., the resolution glimpses obtained from the camera sensor.

AVE process. Now, we can define the active visual exploration process as a sequence of glimpse selections. Let T be the maximum number of exploration steps. The process starts at time $t = 0$ with an empty sequence of observations. At time $t \in \{1, ..., T\}$, the camera captures a glimpse G_t at coordinates C_t proposed by the model, generating a patch of resolution $d_{\text{cam}} \times d_{\text{cam}}$. We simulate the process of glimpse capturing by cropping a patch from a large image representing

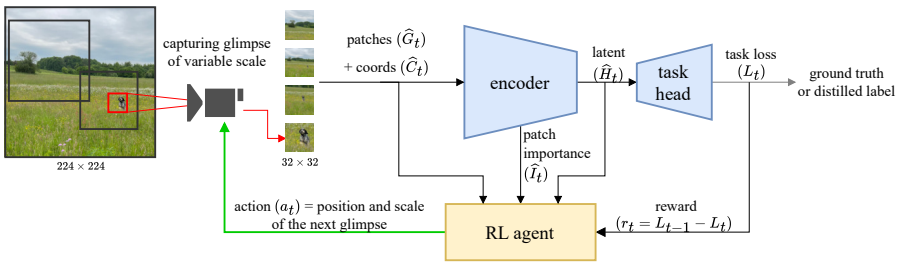


Fig. 2: Architecture: AdaGlimpse consists of two parts: a vision transformer-based encoder with a task-specific head (see Sec. 3.2) and a Soft Actor-Critic RL agent (see Sec. 3.3). At each exploration step, the RL agent selects the position and scale of the next glimpse based on the information about previous patches, their coordinates, importance, and latent representations.

the environment X and scaling it to $d_{\text{cam}} \times d_{\text{cam}}$. The model stores information about glimpses; therefore, at step t, it can access glimpses $G_1, G_2, ..., G_t$ and their corresponding coordinates. The exploration process is stopped when a set confidence level or a maximum number of glimpses is reached. As we will show in Sec. 3.3, this process can be formulated as a Markov Decision Process (MDP) to leverage RL methods [44].

3.2 Vision transformer with variable scale sampling

The architecture of our backbone network consists of two parts: an encoder based on a modified version of ViT [10] and a task-specific head (decoder). The goal is to perform the main task, e.g. classification or reconstruction, based on already observed glimpses while providing information for the RL agent network.

Glimpse encoder. At each step t, the ViT encoder is provided with a sequence of glimpses $G_1, G_2, ..., G_{t-1}$ and their coordinates $C_1, C_2, ..., C_{t-1}$. Depending on the d_{cam} resolution relative to the ViT native patch size d_{patch}, each glimpse G_i is divided with a standard sampling grid into a sequence of patches $G'_i = g'_{i,1}, ..., g'_{i,k}$ with coordinates $C'_i = c'_{i,1}, ..., c'_{i,k}$, where $k = (\lceil d_{\text{cam}}/d_{\text{patch}} \rceil)^2$. The resulting sequences of patches and coordinates from all previous glimpses are concatenated into $\widehat{G}_t$ and $\widehat{C}_t$, respectively.

The standard ViT positional embeddings assume that all patches are sampled from a regular grid. As our method relaxes this constraint, we apply ElasticViT [31] positional encoding calculated according to the patch coordinates. Finally, a trainable class token is appended to the sequence, which is then passed through ViT transformer blocks. The encoder outputs a sequence of latent tokens $\widehat{H}_t$, one per patch, and the class token. Furthermore, we estimate the importance of each input patch, calculating a transformer attention rollout [1], which results in a sequence $\widehat{I}_t$.

Task-specific decoder. The head of the classification model is a simple linear layer taking as an input the class token, as in standard ViT. However, for dense prediction tasks (e.g., reconstruction, segmentation), a MAE-like [18] transformer decoder is used. Then, the decoder receives a sequence of all tokens from the encoder and a full grid of mask tokens as input. The mask tokens consist of a positional embedding for each position in the decoder grid and a shared learnable query embedding, indicating that the token value is to be predicted. This is in contrast to MAE, which only uses mask tokens for unknown areas of the image. However, using tokens for all positions is essential in our case because with variable scale sampling we must predict the entire image rather than solely focus on the absent portions.

The decoder consists of a series of transformer blocks, generating output tokens projected through a linear layer for reconstruction or a progressive upscale module [52] of 4 convolutional layers and interpolations for segmentation. Finally, the output mask tokens are arranged according to a grid, and the remaining ones are discarded.

Training objectives. The entire backbone network is trained using a task-specific loss function. Optimization is performed only on the last exploration step $t = T$ after seeing all glimpses. The loss function for reconstruction is the root mean squared error (RMSE). For classification and segmentation, we use distilled soft targets computed by a teacher model and the Kullback-Leibler divergence as the loss function. The teacher model is a pre-trained ViT from [45] for classification and a DeepLabV3 [8] with a ResNet-101 backbone for segmentation. In both cases, the teacher model is provided with the entire scene X, as in STAM [35].

3.3 Soft Actor-Critic agent

We consider AVE as a Markov Decision Process (MDP), where at timestep t, the agent observing state s_t (information about previous glimpses) takes action a_t (coordinates of the next glimpse). It leads to state s_{t+1} (information about previous glimpses and the next glimpse) as well as the reward r_{t+1} (scalar estimating how much the glimpse helped in refining the prediction). Presenting AVE as MDP allows us to leverage reinforcement learning algorithms.

Preliminaries. The focal point of reinforcement learning is the policy π_θ, which chooses the next action based on the current state, i.e., $a_t \sim \pi_\theta(s_t)$. The goal is to find the policy π_θ that maximizes the value function, corresponding to the expected discounted sum of rewards: $V^{\pi_\theta}(s) := \mathbb{E}_{\pi_\theta}\left[\sum_{t=k}^{T} \gamma^t r_t | s_k = s\right]$, where $\gamma = 0.99$ is the discount factor. The expectation is taken under the policy π_θ, i.e., we use π_θ to take actions in the environment. As such, we aim to find $\theta^* = \arg\max_\theta V^{\pi_\theta}$. Additionally, in order to evaluate actions and facilitate the learning of the policy, we define the state-action value function

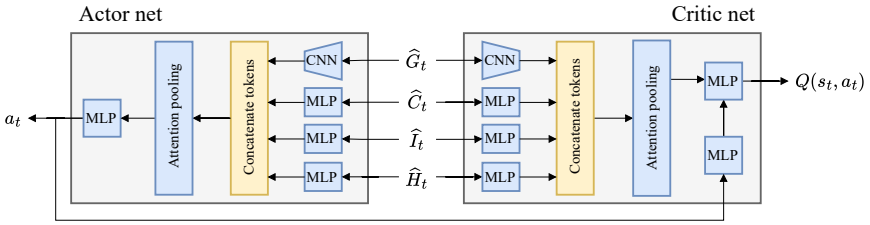


Fig. 3: RL agent: RL module of AdaGlimpse uses two networks: the actor and the critic. The actor predicts the action a_t (position and scale of the next glimpse) based on state $s_t = (\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$. The critic estimates the $Q(s_t, a_t)$, corresponding to the expected cumulative reward for taking this action.

$Q^{\pi_\theta}(s, a) := r(s, a) + \gamma \mathbb{E}_{s'} V^{\pi_\theta}(s')$, where $r(s, a)$ is the reward received in state s when executing action a, and state s' represents the next sampled state. Intuitively, it represents the expected return of our policy, that is, the value function of executing the action a in the first step and choosing the subsequent actions according to the policy π. Below, we define observations, actions, rewards, the maximum-entropy objective central to the Soft Actor-Critic algorithm, and the design of our actor and critic architectures.

State, action and reward. To create the *state* for the RL agent, we supplement the glimpses G_t with additional information to make the inference easier. In particular, the state s_t consists of sequences $(\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$, where (as defined in Sec. 3.1 and 3.2):

- $\widehat{G}_t$ are all patches of the previously sampled glimpses,
- $\widehat{C}_t$ are coordinates of these patches,
- $\widehat{I}_t$ are their importance (one importance value per patch),
- $\widehat{H}_t$ are latent representations of these patches (one token per patch).

Notice that the starting state s_0 is an empty sequence corresponding to the fact that we do not know anything about the environment.

AdaGlimpse proposes continuous-valued *actions* that describe the arbitrary position and scale of an image. Therefore, the action is a tuple $(x, y, z) \in [0, 1]^3$, where x, y represents the normalized coordinates of the top-left glimpse corner and z is its scale.

Finally, we define the *reward* as the difference between the loss in the successive timesteps, i.e. $r_t = L_{t-1} - L_t$.

Soft Actor-Critic. RL objective can be optimized with various approaches [44]. However, since exploration is crucial to solving AVE, we decided to use Soft Actor-Critic [16] (SAC) reinforcement learning algorithm. SAC operates in the maximum entropy framework, meaning that besides maximizing the expected

sum of rewards, it also takes into account the entropy of the action distribution. That is, the goal is to optimize $V^{\pi_\theta}(s) := \mathbb{E}_\pi \left[\sum_{t=k}^{T} \gamma^t r_t + \alpha \mathcal{H}(\pi(s_t)) | s_k = s\right]$, where $\mathcal{H}$ is the entropy, and α is used to weigh its importance. Higher α values encourage the RL algorithm to find more exploratory policies, resulting in more diverse actions.

Actor and Critic architectures. AdaGlimpse requires two networks: the actor that encodes the policy π and the critic that encodes the state-action value function Q. For this purpose, we build a custom-crafted architecture, see Fig. 3. In particular, we create separate token encoders for each part of the input s_t: a small convolutional network that processes patches in $\widehat{G}_t$, and a small MLP for each of the other inputs $\widehat{C}_t, \widehat{I}_t, \widehat{H}_t$. As a result, we obtain four embedding vectors for each patch. We concatenate and process them using an attention pooling [20] layer to combine information across patches. Finally, we use another MLP to obtain the action a_t for the actor and the value prediction $Q(s_t, a_t)$ for the critic. Despite the similarity in the actor and critic architectures, we do not share any parameters between them as it destabilizes the training process.

4 Experimental Setup

Architecture. In all our experiments we use an encoder of the same size as standard ViT-B [10], i.e., 12 transformer blocks and embedding size of 768. The decoder consists of 8 blocks with the embedding size of 512. For the RL networks the hidden dimension size is 256, the number of attention heads is 8 and MLPs have 3 layers. All networks use GELU activation functions [19].

Training. We adopt the AdamW optimization algorithm [27], setting the weight decay value to 10^{-4} and the initial learning rate to 10^{-5} for classification and 10^{-4} for other tasks. The learning rate is then decayed using a half-cycle cosine rate to 10^{-8} for the remainder of the training. The model is trained for 100 epochs with early stopping. During training, we alternate between optimizing the backbone and the RL agent each epoch, except for the first 30 epochs when we train the RL agent only. We augment the training data using the 3-Augment regime proposed in [45] extended with a random affine transform for the segmentation target. Additionally, we pre-train the model for 600 epochs with 196 random glimpses per image with sizes and positions sampled from a uniform distribution. In segmentation experiments we fine-tune a model trained for reconstruction to accommodate for the relatively small size of the dataset.

Datasets and metrics. We assess our method on several publicly available vision datasets. ImageNet-1k [9] is used for classification and reconstruction tasks. The performance of zero-shot reconstruction is evaluated on MS COCO 2014 [26], ADE20K [53] and SUN360 [43]. Semantic segmentation is evaluated

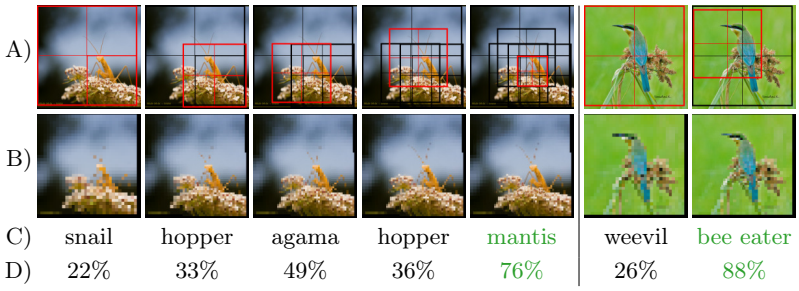


Fig. 4: Glimpse selection step-by-step: AdaGlimpse explores 224×224 images from ImageNet with 32×32 glimpses of variable scale, zooming in on objects of interest and stopping the process after reaching 75% predicted probability. The rows correspond to: A) glimpse locations, B) pixels visible to the model (interpolated from glimpses for preview), C) predicted label, D) prediction probability.

on the ADE20K dataset (the MIT scene parsing benchmark subset) [53]. Since the SUN360 dataset does not have a predetermined train-test split, we use a 9:1 train-test split according to an index provided by the authors of [39]. We report accuracy for classification tasks, root mean squared error (RMSE) for reconstruction, and pixel average precision (AP), class-mean average precision (mAP) and class-mean intersection over union score (mIoU) for segmentation.

Glimpse regimes We compare our model to baselines with different glimpse regimes, that can be categorized as follows: (a) simple - square glimpses with a fixed and constant resolution [21, 32, 35], (b) retinal - retina-like glimpses [38] with more pixels in the center than on the edges [32, 39, 41], (c) full+simple - one low-resolution glimpse of the entire scene followed by simple glimpses [3, 12, 30, 37, 46, 47], and (d) adaptive - our variable scale glimpse regime.

For easy comparison of different glimpse regimes, we provide a *pixel percentage* metric representing the percentage of image pixels known to the model, as defined in [32], which is calculated as the number of pixels captured in all glimpses divided by the number of pixels in the full scene image.

5 Results

In this section, we present an evaluation of AdaGlimpse compared to competitive methods, followed by an analysis of our approach. Both quantitative and qualitative results for all baseline methods were taken from [32] for reconstruction and segmentation, and [35] for classification. Further results are provided in the supplementary materials.

An overview of glimpse selection performed by our model is portrayed in Fig. 4. The visualization demonstrates the arbitrary glimpse position and scale capabilities of our method. AdaGlimpse detects objects of interest and zooms in on them to extract fine details.

Table 1: Reconstruction results: RMSE (lower is better) obtained by our model for reconstruction task against AttSeg [41], GlAtEx [39], SimGlim [21], and AME [32] on ImageNet-1k, SUN360, ADE20K and MS COCO datasets. Regardless of the number of glimpses, as well as their resolution and regime (see Sec. 4), our method outperforms competitive solutions. Note that Pixel % denotes the percentage of image pixels known to the model, † a reproduced result not published in the relevant paper, and * zero-shot performance.

Method	IMNET	SUN360	ADE20k	COCO	Image res.	Glimpses	Regime	Pixel %
AME	30.3†	29.8	30.8	32.5	128×256	8×32^2	simple	25.00
Ours	**14.5**	**11.1***	**14.0***	**14.5***	224×224	12×32^2	adaptive	24.49
AttSeg	–	37.6	36.6	41.8	128×256	8×48^2	retinal	18.75
GlAtEx	–	33.8	41.9	40.3	128×256	8×48^2	retinal	18.75
AME	–	23.6	23.8	25.2	128×256	8×48^2	retinal	18.75
SimGlim	–	26.2	27.2	29.8	224×224	37×16^2	simple	18.75
AME	–	23.4	26.2	28.6	224×224	37×16^2	simple	18.75
Ours	**14.7**	**11.1***	**14.2***	**14.7***	224×224	9×32^2	adaptive	18.36
AME	–	37.9	40.7	43.2	128×256	8×16^2	simple	6.25
Ours	**20.9**	**17.6***	**20.5***	**21.5***	224×224	12×16^2	adaptive	6.12
Ours	**20.7**	**17.2***	**20.7***	**21.4***	224×224	3×32^2	adaptive	6.12

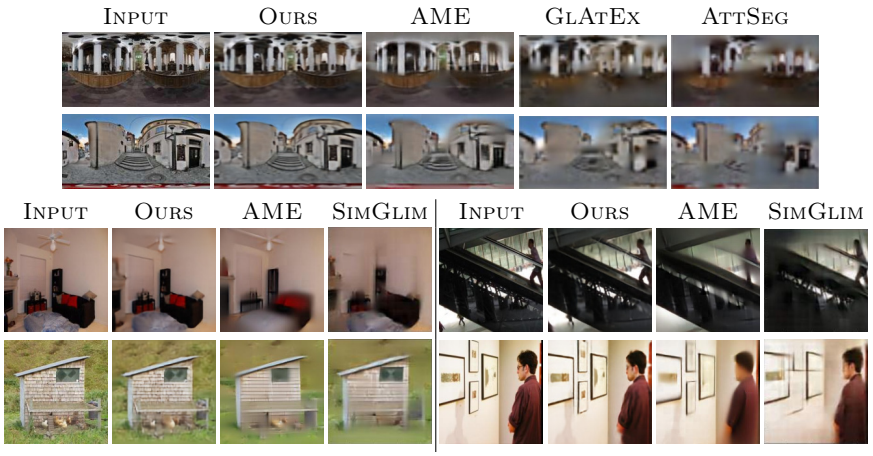


Fig. 5: Reconstruction quality for SUN360 (top) and ADE20K (bottom): Sample reconstructions of our method compared with AME [32], AttSeg [41], GlAtEx [39] and SimGlim [21] on the SUN360 and ADE20K datasets. Reconstructions done with our method are visibly more detailed and less blurry than those obtained by baseline methods. Notice that images for comparison were taken from the baseline publications (we did not select them).

Table 2: Classification results: Accuracy obtained by our model for classification task against DRAM [3], GFNet [47], Saccader [12], STN [37], TNet [30], PatchDrop [46] and STAM [35] on ImageNet-1k dataset. Our AdaGlimpse needs 40% less pixels to match the performance of the best baseline method. Note that Pixel % denotes the percentage of image pixels known to the model, while regimes are described in Sec. 4.

Method	Accuracy %	Glimpses	Regime	Pixel %
DRAM	67.50	8×77^2	full+simple	94.53
GFNet	75.93	5×96^2	full+simple	91.84
Saccader	70.31	6×77^2	full+simple	70.90
TNet	74.62	6×77^2	full+simple	70.90
STN	71.40	9×56^2	full+simple	56.25
PatchDrop	76.00	$\sim 8.9 \times 56^2$	full+simple+stopping	∼55.63
STAM	76.13	14×32^2	simple	28.57
Ours	**77.54**	14×32^2	adaptive	28.57
Ours	**76.30**	$\sim 8.3 \times 32^2$	adaptive+stopping	**∼16.94**

5.1 Tasks

Reconstruction. Reconstructing the entire scene from observed glimpses verifies comprehensive scene understanding. In Tab. 1 we group the results by pixel percentage (fraction of pixels of the original input image known to the model). Our approach outperforms existing methods by a large margin. In particular, with only 6% of the pixels seen, AdaGlimpse performs better than other methods that had over 18% of the image visible to them. Note that unlike baseline methods, we train our model only on ImageNet-1k without fine-tuning for each evaluated dataset. Qualitative outcomes in Fig. 5 showcase AdaGlimpse producing reconstructions with a higher level of detail, reproducing more objects compared to baseline methods. All models except AttSeg have backbone networks pre-trained on ImageNet-1k; AttSeg's pre-training details were not disclosed.

Classification. Results for multi-class classification on the ImageNet-1k dataset can be found in Tab. 2. AdaGlimpse outperforms all prior methods, achieving 77.54% (±0.18) accuracy compared to 76.13% of the best baseline method STAM [35]. With early exploration termination after reaching 85% probability of predicted class, it requires over 40% less pixels to match STAM accuracy. Visualizations of glimpse selection for classification are presented in Fig. 4. With an early stopping probability threshold of 75%, it is able to classify 224×224 images using only a few 32×32 glimpses of 4 patches each.

Segmentation. The goal of the semantic segmentation task is to classify each pixel of the full scene based on the captured glimpses. The numerical results are presented in Tab. 3. AdaGlimpse outperforms both AttSeg [41] and GlAtEx [39] by a large margin. It performs on par with AME [32] in terms of accuracy;

Table 3: Segmentation results: Comparison of our model against AttSeg [41], GlAtEx [39], and AME [32] on the ADE20K dataset. Our method performs on pair with AME, but requires 35% less pixels, while outperforming other competitive methods on all considered metrics: Pixel-wise Accuracy (mPA, higher is better), Pixel-Accuracy (PA, higher is better), and Intersection over Union (IoU, higher is better).

Method	PA %	mPA %	IoU %	Image res.	Glimpses	Regime	Pixel %
AttSeg	47.9	–	–	128×256	8×48^2	retinal	18.75
GlAtEx	52.4	–	–	128×256	8×48^2	retinal	18.75
Ours	**67.4**	**29.4**	**22.7**	224×224	4×48^2	adaptive	18.36
AME	70.3	32.2	24.4	128×256	8×48^2	simple	56.25
Ours	70.0	32.8	25.7	224×224	8×48^2	adaptive	**36.73**

Table 4: Importance of state components: The RL state consists of the sequence $(\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$ as described in Sec. 3.3. For this study, we replaced each element with its mean value (averaged over the entire dataset) to see how important it is for the model. As a result, we observe that the transformer latent is the most informative part of the state, followed by glimpse coordinates.

patches $\widehat{G}_t$	coordinates $\widehat{C}_t$	importance $\widehat{I}_t$	latent $\widehat{H}_t$	Accuracy %
✓	✓	✓	✓	77.54
✗	✓	✓	✓	76.99
✓	✗	✓	✓	68.25
✓	✓	✗	✓	77.36
✓	✓	✓	✗	61.82

however, it requires 35% less information to achieve this result. Consistently, visualizations in Sec. 5 confirm the quality of the produced segmentation maps.

5.2 Analysis

Percentage of image pixels. The relationship between the percentage of the full image pixels known to the model (pixel %, see Sec. 4) and performance is plotted in Fig. 7. AdaGlimpse requires fewer pixels to perform better than the baseline methods. In particular for reconstruction, with only 5% of pixels it produces superior results to those achieved by competitive approaches when provided with 25% of scene pixels.

Importance of RL state elements. The key component of AdaGlimpse reinforcement learning algorithm is the state, which consists of the sequence $(\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$ as described in Sec. 3.3. In Tab. 4, we present the ImageNet-1k classification performance when omitting and replacing each component with

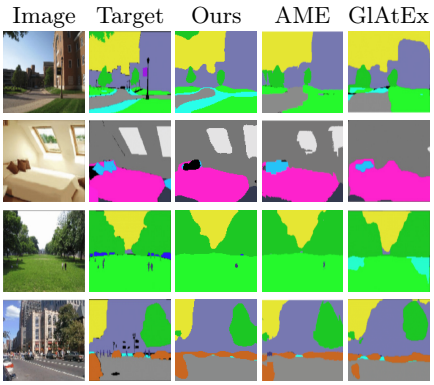


Fig. 6: Segmentation qualitative results. Sample semantic segmentation of our method compared with AME [32] and GlAtEx [39] on the ADE20k dataset. In terms of quality, the segmentation maps generated by our approach are at least comparable to those produced by competing methods.

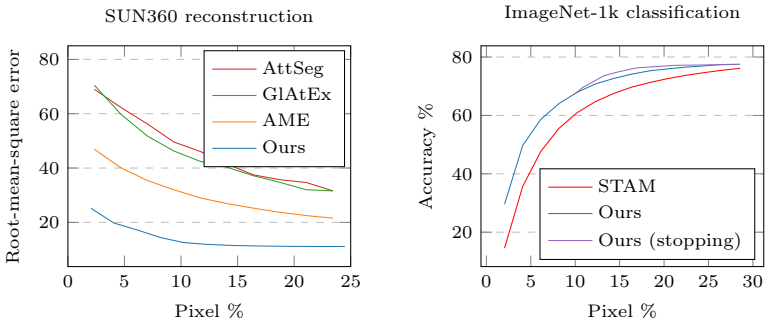


Fig. 7: Percentage of image pixels observed: Figures present the relationship between the amount of pixels observed by the model relative to the full scene resolution (pixel %), and its performance. AdaGlimpse outperforms competitive solutions, requiring significantly less information to achieve the same performance level.

its mean value during inference. The resulting lower accuracy proves the significance of each component. The transformer latent $\widehat{H}_t$ and glimpse positions $\widehat{C}_t$ are especially crucial for this task, highlighting both the importance of the glimpse location within the original scene and the benefit of using the processed input over the original image.

Glimpse location. In Fig. 8, we illustrate the average location of each subsequent glimpse, revealing a notable distinction between reconstruction and classification tasks explored by AdaGlimpse. In the reconstruction task, attention

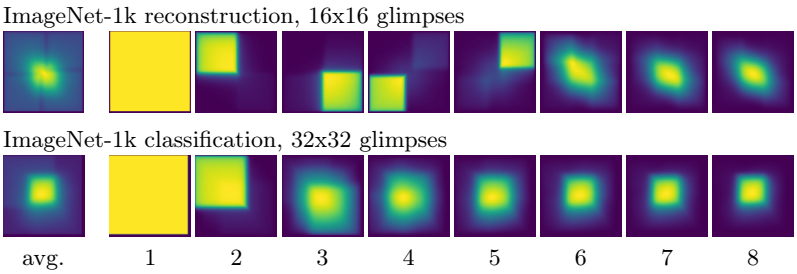


Fig. 8: Average glimpse image: Mean glimpse maps for models trained for reconstruction (top) and classification (bottom) averaged over all test images. On the left, an average map for all glimpses is presented, followed by maps for successive glimpses $t = 1, ..., 8$. One can observe that AdaGlimpse learns to select the entire image as the first glimpse for both tasks, but subsequent glimpse maps differ. Four successive glimpses in reconstruction concentrate on four parts of the image, while for classification, they mostly explore the center.

spans all image regions initially and later focuses on key elements. In contrast, in the classification task our model swiftly identifies crucial class-specific elements in the image center, directing attention those regions early on. Notably, both tasks uncover the well-known fact about ImageNet-1k, where key objects are concentrated around the image center [23].

6 Discussion and Conclusions

This paper presents AdaGlimpse, a novel approach to Active Visual Exploration, which enables the selection and processing of glimpses at arbitrary positions and scales. We formulate the glimpse selection problem as a Markov Decision Process with continuous action space and leverage the Soft Actor-Critic reinforcement learning algorithm, which specializes in exploration problems. Our task-agnostic architecture allows for a more efficient exploration and understanding of environments, significantly reducing the number of observations needed. AdaGlimpse can quickly analyze the scene with large low-resolution glimpses before zooming in on details for a closer inspection. Its success across multiple benchmarks suggests a broad applicability and potential for further development in embodied AI and robotics.

While excelling in exploration, AdaGlimpse is limited in performance by the underlying transformer architecture, which incurs quadratic computational cost relative to the number of sampled patches. A possible way to overcome this limitation is to replace it with a selective structured state-space model [14, 15]. Finally, although AdaGlimpse perform well on current benchmarks, they do not fully reflect the complexity of Active Visual Exploration, as they do not incorporate dynamic scenes that change over time. As such, further evaluations are required before real-life deployment.

Acknowledgments

This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. This research was funded by National Science Centre, Poland (grant no. 2023/49/N/ST6/02465, 2022/47/B/ST6/03397, 2022/45/B/ST6/02817, and 2023/50/E/ST6/00469). The research was supported by a grant from the Faculty of Mathematics and Computer Science under the Strategic Programme Excellence Initiative at Jagiellonian University. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016558.

References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. arXiv preprint arXiv:2005.00928 (2020)
2. Alexe, B., Heess, N., Teh, Y., Ferrari, V.: Searching for objects driven by context. Advances in Neural Information Processing Systems **25** (2012)
3. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. In: ICLR (2015)
4. Ba, J., Salakhutdinov, R.R., Grosse, R.B., Frey, B.J.: Learning wake-sleep recurrent attention models. Advances in Neural Information Processing Systems **28** (2015)
5. Beyer, L., Izmailov, P., Kolesnikov, A., et al.: Flexivit: One model for all patch sizes. arXiv:2212.08013 (2022)
6. Caicedo, J.C., Lazebnik, S.: Active object localization with deep reinforcement learning. In: Proceedings of the IEEE international conference on computer vision. pp. 2488–2496 (2015)
7. Chai, Y.: Patchwork: A patch-wise attention network for efficient object detection and segmentation in video streams. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3415–3424 (2019)
8. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. IEEE (2009)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
11. Double Robotics, Inc.: Double 3 - telepresence robot for the hybrid office. `https://www.doublerobotics.com/` (2024), accessed: 2024-02-24
12. Elsayed, G., Kornblith, S., Le, Q.V.: Saccader: Improving accuracy of hard attention models for vision. Advances in Neural Information Processing Systems **32** (2019)
13. Girdhar, R., Singh, M., Ravi, N., van der Maaten, L., Joulin, A., Misra, I.: Omnivore: A Single Model for Many Visual Modalities. In: ICCV (2022)
14. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
15. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021)

16. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. pp. 1861–1870. PMLR (2018)
17. Hayhoe, M., Ballard, D.: Eye movements in natural behavior. Trends in cognitive sciences **9**(4), 188–194 (2005)
18. He, K., Chen, X., Xie, S., et al.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 16000–16009 (2022)
19. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
20. Ilse, M., Tomczak, J., Welling, M.: Attention-based deep multiple instance learning. In: International conference on machine learning. pp. 2127–2136. PMLR (2018)
21. Jha, A., Seifi, S., Tuytelaars, T.: Simglim: Simplifying glimpse based active visual reconstruction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 269–278 (2023)
22. Krotenok, A.Y., Yu, A.S., Yu, V.A.: The change in the altitude of an unmanned aerial vehicle, depending on the height difference of the area taken. In: IOP Conference Series: Earth and Environmental Science. vol. 272, p. 022165. IOP Publishing (2019)
23. Kümmerer, M., Theis, L., Bethge, M.: Deep gaze I: boosting saliency prediction with feature maps trained on imagenet. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings (2015)
24. Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., Gao, J.: Efficient self-supervised vision transformers for representation learning. arXiv preprint arXiv:2106.09785 (2021)
25. Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., Jia, J.: Mat: Mask-aware transformer for large hole image inpainting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10758–10768 (2022)
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
27. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
28. Mathe, S., Pirinen, A., Sminchisescu, C.: Reinforcement learning for visual object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2894–2902 (2016)
29. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. Advances in neural information processing systems **27** (2014)
30. Papadopoulos, A., Korus, P., Memon, N.: Hard-attention for scalable image classification. Advances in Neural Information Processing Systems **34**, 14694–14707 (2021)
31. Pardyl, A., Kurzejamski, G., Olszewski, J., Trzciński, T., Zieliński, B.: Beyond grids: Exploring elastic input sampling for vision transformers. arXiv preprint arXiv:2309.13353 (2023)
32. Pardyl, A., Rypeść, G., Kurzejamski, G., Zieliński, B., Trzciński, T.: Active visual exploration based on attention-map entropy. In: Elkind, E. (ed.) Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. pp. 1303–1311 (8 2023), main Track
33. Ramakrishnan, S.K., Jayaraman, D., Grauman, K.: An exploration of embodied visual exploration. International Journal of Computer Vision **129**, 1616–1649 (2021)

34. Rangrej, S.B., Clark, J.J.: A probabilistic hard attention model for sequentially observed scenes. arXiv preprint arXiv:2111.07534 (2021)
35. Rangrej, S.B., Srinidhi, C.L., Clark, J.J.: Consistency driven sequential transformers attention model for partially observable scenes. In: CVRP. pp. 2518–2527 (2022)
36. Ranzato, M.: On learning where to look. arXiv preprint arXiv:1405.5488 (2014)
37. Recasens, A., Kellnhofer, P., Stent, S., Matusik, W., Torralba, A.: Learning to zoom: a saliency-based sampling layer for neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 51–66 (2018)
38. Sandini, G., Metta, G.: Retina-like sensors: motivations, technology and applications. In: Sensors and sensing in biology and engineering, pp. 251–262. Springer (2003)
39. Seifi, S., Jha, A., Tuytelaars, T.: Glimpse-attend-and-explore: Self-attention for active visual exploration. In: ICCV. pp. 16137–16146 (2021)
40. Seifi, S., Tuytelaars, T.: Where to look next: Unsupervised active visual exploration on 360° input. CoRR **abs/1909.10304** (2019), `http://arxiv.org/abs/1909.10304`
41. Seifi, S., Tuytelaars, T.: Attend and segment: Attention guided active semantic segmentation. In: European Conference on Computer Vision. pp. 305–321. Springer (2020)
42. Śmieja, M., Struski, Ł., Tabor, J., Zieliński, B., Spurek, P.: Processing of missing data by neural networks. Advances in neural information processing systems **31** (2018)
43. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 567–576 (2015)
44. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
45. Touvron, H., Cord, M., Jégou, H.: Deit iii: Revenge of the vit. In: European Conference on Computer Vision. pp. 516–533. Springer (2022)
46. Uzkent, B., Ermon, S.: Learning when and where to zoom with deep reinforcement learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12345–12354 (2020)
47. Wang, Y., Lv, K., Huang, R., Song, S., Yang, L., Huang, G.: Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. Advances in Neural Information Processing Systems **33**, 2432–2444 (2020)
48. Wenzel, P., Wang, R., Yang, N., Cheng, Q., Khan, Q., von Stumberg, L., Zeller, N., Cremers, D.: 4seasons: A cross-season dataset for multi-weather slam in autonomous driving. In: Akata, Z., Geiger, A., Sattler, T. (eds.) Pattern Recognition. pp. 404–417. Springer International Publishing, Cham (2021)
49. Wu, C.Y., Girshick, R., He, K., Feichtenhofer, C., Krahenbuhl, P.: A multigrid method for efficiently training video models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 153–162 (2020)
50. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057. PMLR (2015)
51. Yoo, D., Park, S., Lee, J.Y., Paek, A.S., So Kweon, I.: Attentionnet: Aggregating weak directions for accurate object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2659–2667 (2015)

52. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6881–6890 (2021)
53. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)

AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale – Supplementary Materials

Adam Pardyl[1,2,3], Michał Wronka[2], Maciej Wołczyk[1], Kamil Adamczewski[1], Tomasz Trzciński[1,4,5], and Bartosz Zieliński[1,2]

[1] IDEAS NCBR
{adam.pardyl, maciej.wolczyk, kamil.adamczewski, tomasz.trzcinski, bartosz.zielinski}@ideas-ncbr.pl
[2] Jagiellonian University, Faculty of Mathematics and Computer Science
michal.wronka@student.uj.edu.pl
[3] Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4] Warsaw University of Technology
[5] Tooploox

In this Supplementary Materials, we present additional experiments and visualizations for our AdaGlimpse active visual exploration method.

Table 1: Heuristic baselines: We compared our reinforcement learning-based method against two heuristic exploration policies for the reconstruction task on the ImageNet-1k dataset. Both heuristic methods use the same ElasticViT backbone as our approach and utilize 32^2 glimpses, consisting of four ViT patches. The first method, called BasicElasticAME, samples one low-resolution glimpse of the entire image, followed by small, high-resolution glimpses selected using the AME algorithm. The second method, called QuadTreeAME, also begins with a single low-resolution glimpse. It then iteratively selects the most uncertain patch using AME, sampling a new glimpse at its position to effectively acquire four new higher-resolution patches in place of the selected lower-resolution patch. For comparison, we also provide results for a standard, fixed-resolution AME baseline. We observed that our method outperforms these heuristic baselines.

Method	RMSE	Image res.	Glimpses	Regime	Pixel %
AME	30.3	128×256	8×32^2	simple	25.00
BasicElasticAME	25.95	224×224	12×32^2	adaptive (heuristic)	24.49
QuadTreeAME	23.2	224×224	12×32^2	adaptive (heuristic)	24.49
Ours	**14.5**	224×224	12×32^2	adaptive	24.49

Table 2: Model and training configuration: In this table we specify the model and training configuration details.

Parameter name	Value
Encoder	
ViT type	`base`
native patch size	16
transformer embed dim	768
transformer blocks	12
attention heads	12
mlp ratio	4
Decoder	
transformer embed dim	512
transformer blocks	8
attention heads	16
mlp ratio	4
RL agent	
hidden dim	256
action distribution	TanhNormal
sac target entropy value	-3
sac initial alpha value	1
Training	
training epochs	100
backbone pre-training epochs	600
backbone lr	10^{-5}
rl agent lr	$5 * 10^{-4}$
backbone weight decay rate	10^{-4}
rl agent weight decay rate	10^{-2}
lr scheduler type	one cycle
lr warmup epochs	10
minimum lr	10^{-8}
initial random action batches	10000
initial frozen backbone epochs	10
rl loss function	L2
rl batch size	256
backbone batch size	128
replay buffer size	10000

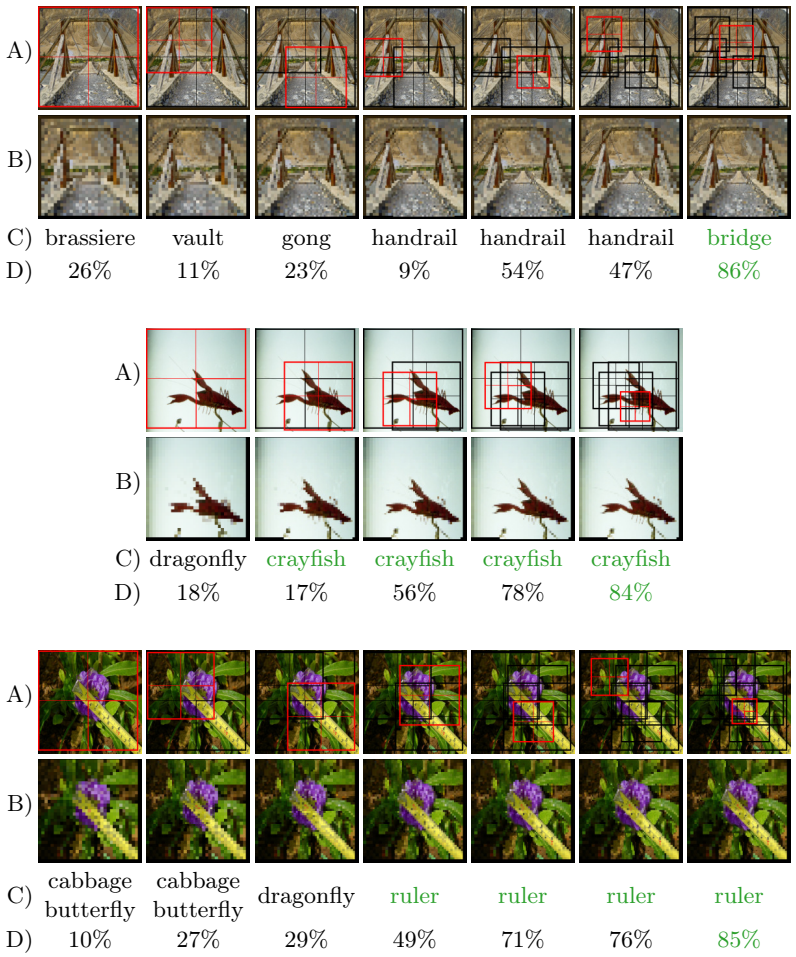


Fig. 1: Image classification on ImageNet-1k step-by-step: AdaGlimpse explores 224×224 images from ImageNet with 32×32 glimpses of variable scale, zooming in on objects of interest and stopping the process after reaching 75% predicted probability. The rows correspond to: A) glimpse locations, B) pixels visible to the model (interpolated from glimpses for preview), C) predicted label, D) prediction probability.

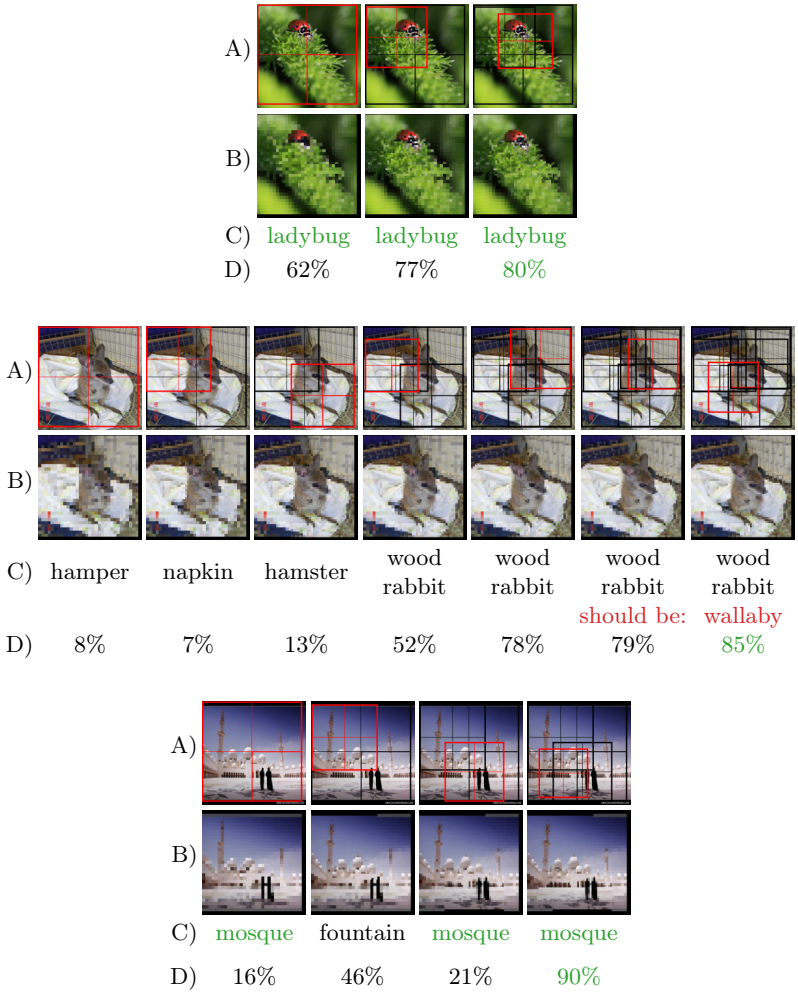


Fig. 2: Image classification on ImageNet-1k step-by-step: AdaGlimpse explores 224×224 images from ImageNet with 32×32 glimpses of variable scale, zooming in on objects of interest and stopping the process after reaching 75% predicted probability. The rows correspond to: A) glimpse locations, B) pixels visible to the model (interpolated from glimpses for preview), C) predicted label, D) prediction probability.

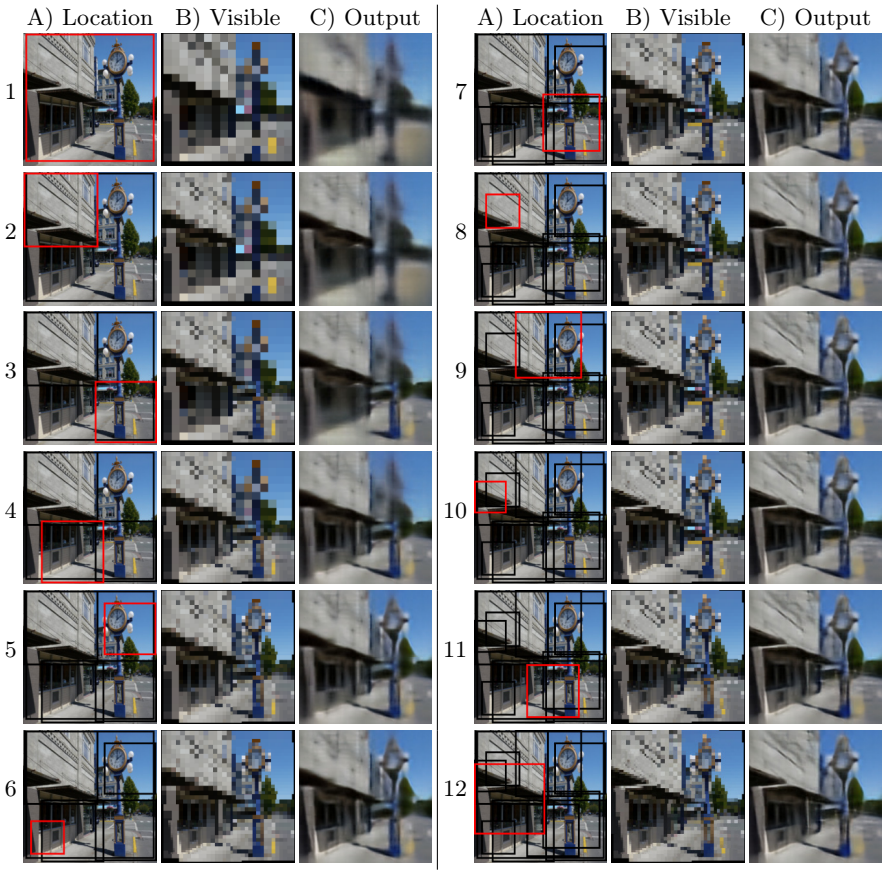


Fig. 3: Image reconstruction on MS COCO step-by-step: AdaGlimpse explores 224×224 images from MS COCO with 8×16^2 glimpses of variable scale, zooming in on objects of interest. Note, that each glimpse consists of a single vision transformer patch. The columns correspond to: A) glimpse locations, B) pixels visible to the model (interpolated from glimpses for preview), C) reconstruction result.

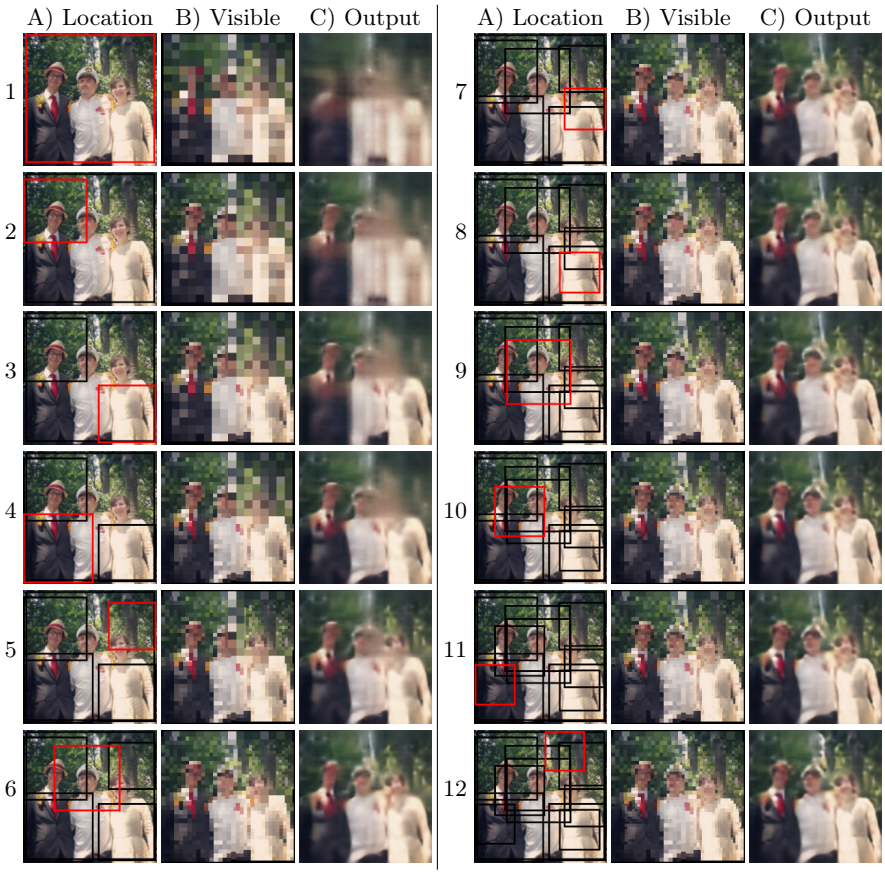


Fig. 4: Image reconstruction on MS COCO step-by-step: AdaGlimpse explores 224×224 images from MS COCO with 8×16^2 glimpses of variable scale, zooming in on objects of interest. Note, that each glimpse consists of a single vision transformer patch. The columns correspond to: A) glimpse locations, B) pixels visible to the model (interpolated from glimpses for preview), C) reconstruction result.

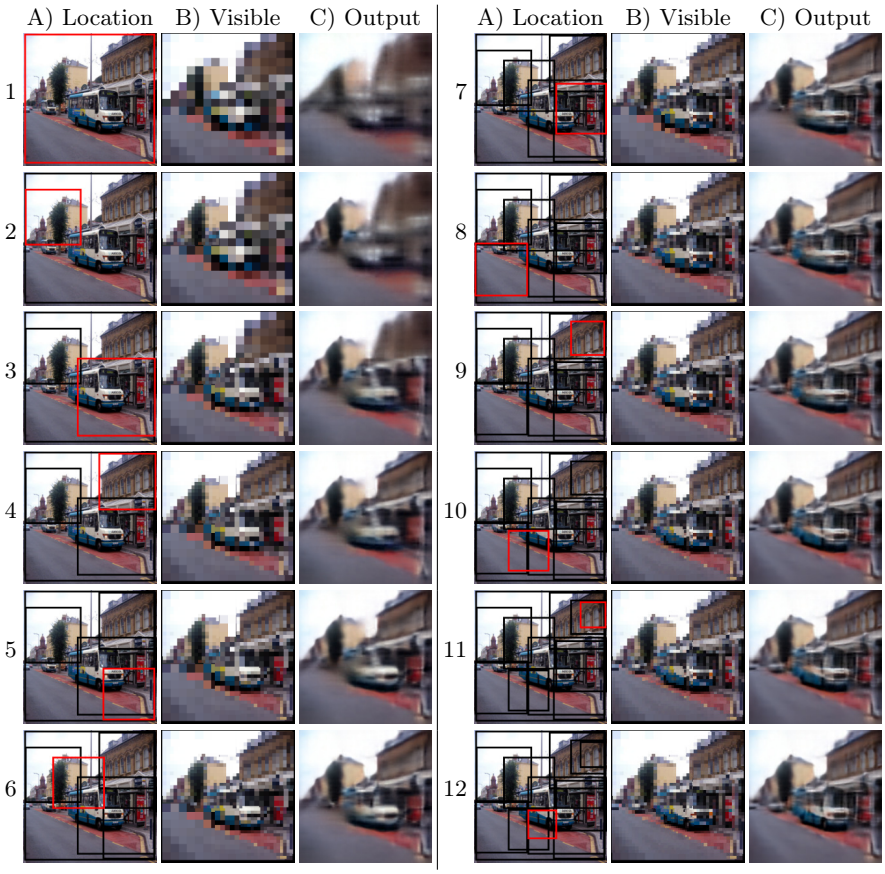


Fig. 5: Image reconstruction on MS COCO step-by-step: AdaGlimpse explores 224×224 images from MS COCO with 8×16^2 glimpses of variable scale, zooming in on objects of interest. Note, that each glimpse consists of a single vision transformer patch. The columns correspond to: A) glimpse locations, B) pixels visible to the model (interpolated from glimpses for preview), C) reconstruction result.

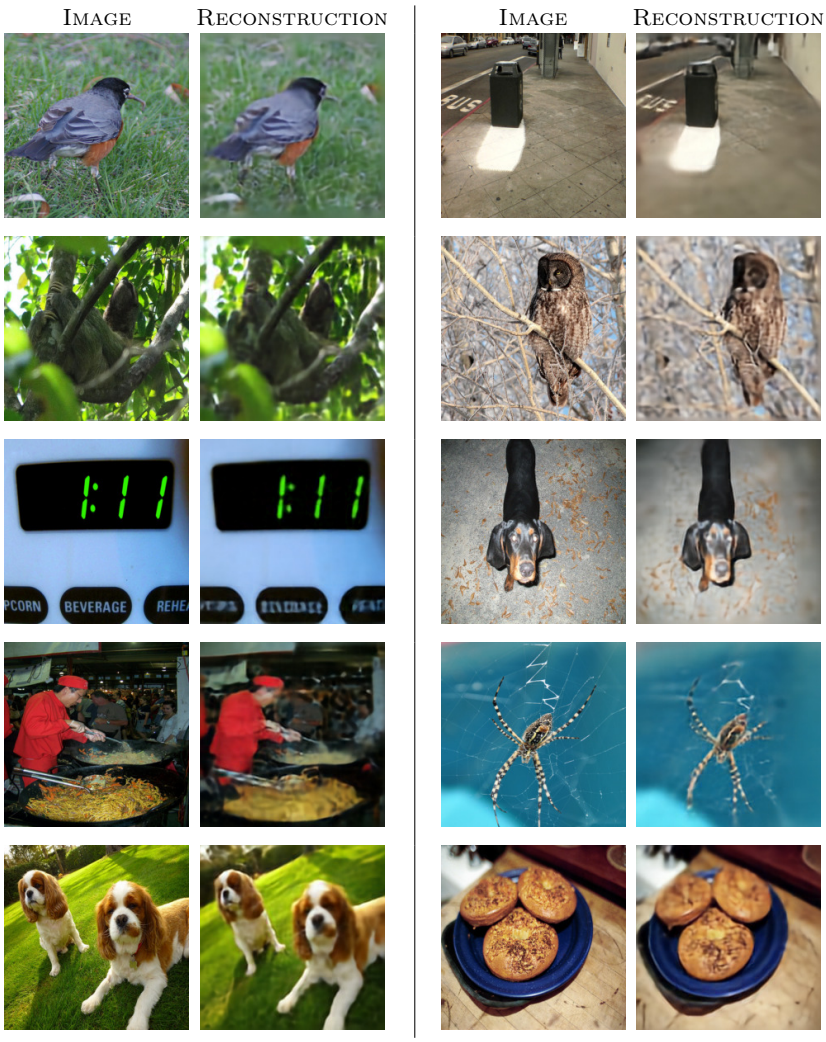


Fig. 6: Image reconstruction on ImageNet-1k: Figure shows the qualitative results of the image reconstruction task. Image size is 224×224 and 12×32^2 adaptive glimpses are used.

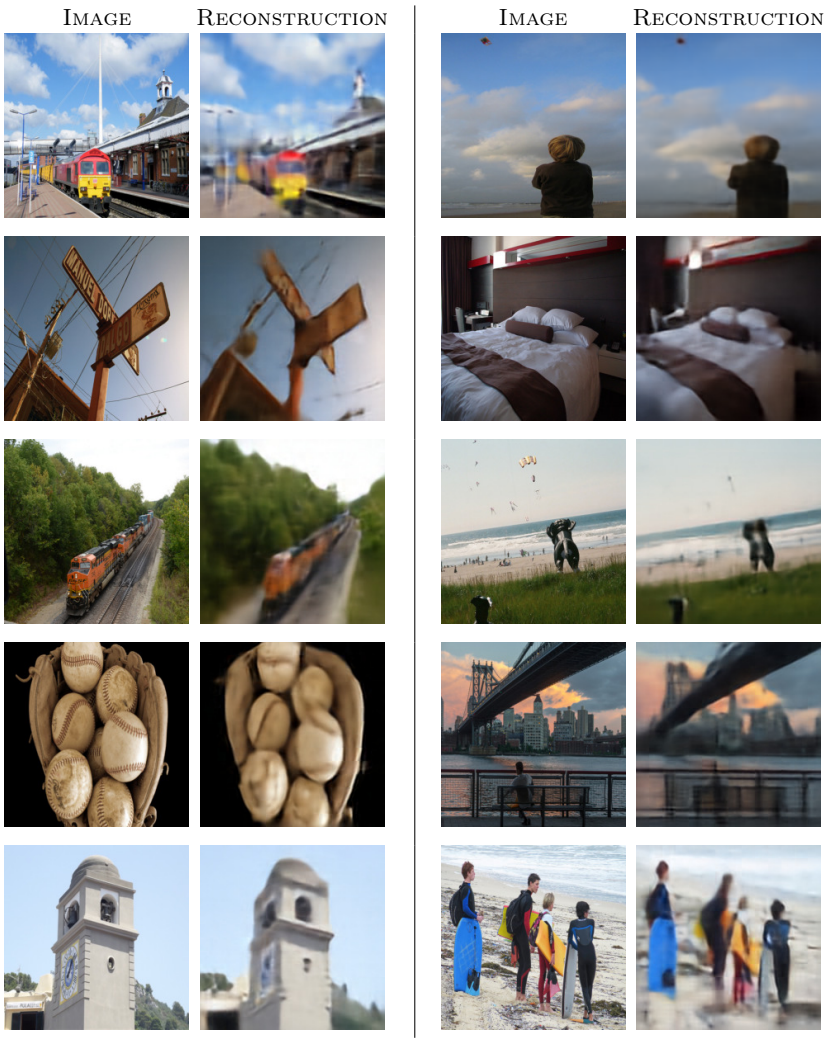


Fig. 7: Image reconstruction on MS COCO: Figure shows the qualitative results of the image reconstruction task. Image size is 224×224 and 12×16^2 adaptive glimpses are used.

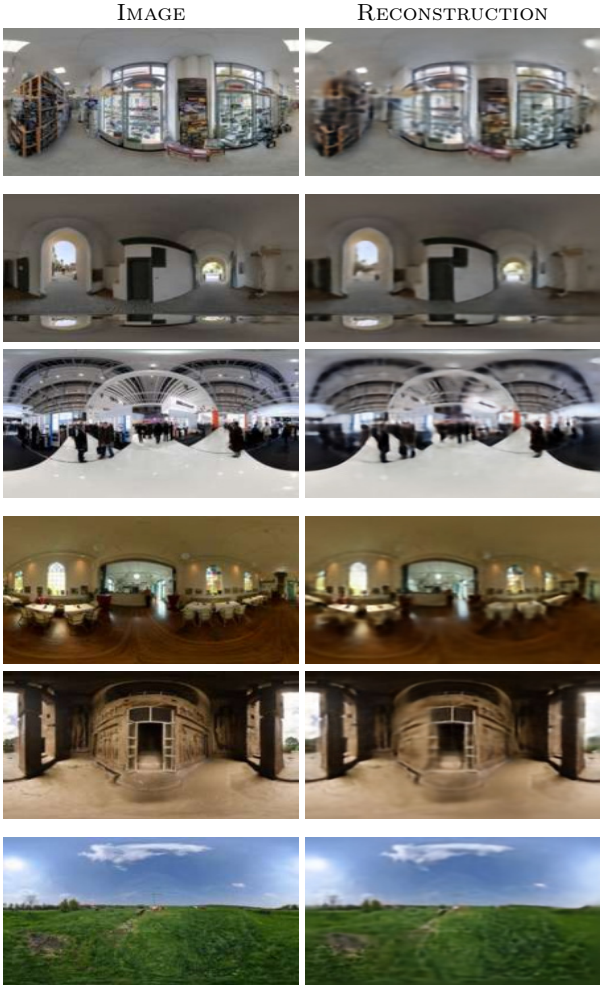


Fig. 8: Image reconstruction on SUN360: Figure shows the qualitative results of the image reconstruction task. Image size is 224×224 and 12×32^2 adaptive glimpses are used. For visualization, images were resized to the original aspect ratio.

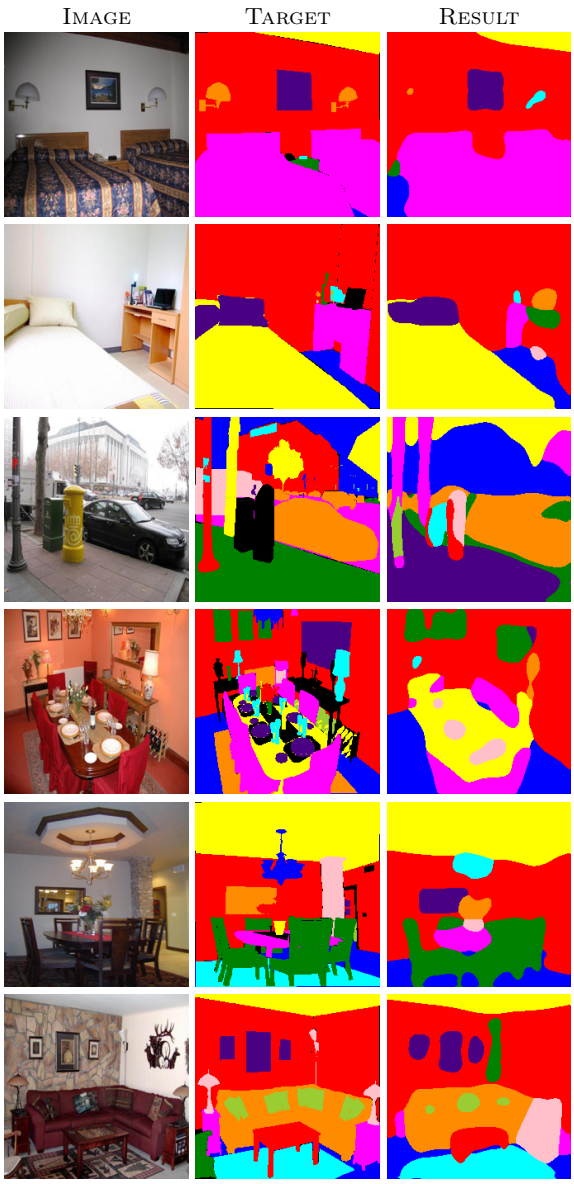


Fig. 9: Image segmentation on ADE20K: Figure shows the qualitative results of the image segmentation task. Image size is 224×224 and 12×48^2 adaptive glimpses are used.

FlySearch: Exploring how vision-language models explore

Adam Pardyl[1,2,3] **Dominik Matuszek**[1,2] **Mateusz Przebieracz**[2] **Marek Cygan**[4,5]

Bartosz Zieliński[2] **Maciej Wolczyk**[1]

Abstract

The real world is messy and unstructured. Uncovering critical information often requires active, goal-driven exploration. It remains to be seen whether Vision-Language Models (VLMs), which recently emerged as a popular zero-shot tool in many difficult tasks, can operate effectively in such conditions. In this paper, we answer this question by introducing FlySearch, a 3D, outdoor, photorealistic environment for searching and navigating to objects in complex scenes. We define three sets of scenarios with varying difficulty and observe that state-of-the-art VLMs cannot reliably solve even the simplest exploration tasks, with the gap to human performance increasing as the tasks get harder. We identify a set of central causes, ranging from vision hallucination, through context misunderstanding, to task planning failures, and we show that some of them can be addressed by finetuning. We publicly release the benchmark, scenarios, and the underlying codebase.

1 Introduction

Vision-Language Models (VLMs) have rapidly emerged as state-of-the-art performers in tasks ranging from image captioning [32, 69] to robotics [7, 44]. However, real-world decision-making requires curiosity, adaptability, and a goal-oriented mindset. Nevertheless, the ability of VLMs to operate in realistic, open-ended environments remains largely untested. In this paper, we propose a benchmark to understand and enhance the exploratory capabilities of VLMs. We draw inspiration from the field of Object Navigation (ObjectNav), which focuses on creating embodied agents capable of finding a specific object in a simulated environment and navigating to it. The object may not be visible from the agent's initial perspective, meaning the agent must perform a careful search to locate it.

There are several significant differences between our benchmark FlySearch and existing counterparts. First, we measure the exploration capabilities of VLMs themselves rather than analyzing more complex systems built on them. Understanding these capabilities is important because gathering information is a crucial aspect of the emerging agentic systems [45]. Second, while most ObjectNav benchmarks take place indoors, we focus on finding objects in a large outdoor space using an Unmanned Aerial Vehicle (UAV). This allows us to examine how VLMs explore large, diverse, and unstructured areas and how they control the altitude to change the search granularity. Finally, we focus on a zero-shot open-ended search setting, i.e., a tested method should not make any prior assumptions about the testing environment, the categories of objects, or the *search process itself*. For example, methods that use detectors of specific objects are out of the scope of our research.

[1]IDEAS NCBR; [2]Jagiellonian University, Faculty of Mathematics and Computer Science; [3]Jagiellonian University, Doctoral School of Exact and Natural Sciences; [4]University of Warsaw; [5]Nomagic.
Correspondence to: Adam Pardyl `adam.pardyl@doctoral.uj.edu.pl`

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

 https://doi.org/10.52202/085713-5594

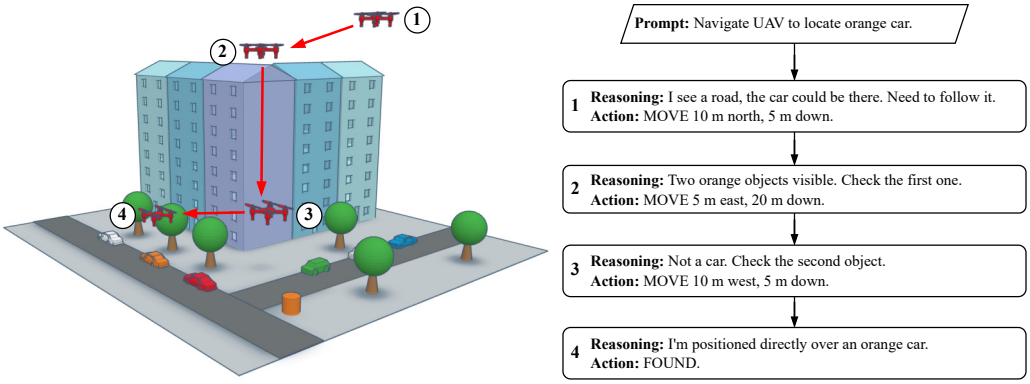


Figure 1: **FlySearch** is a benchmark that evaluates exploration skills using vision-language reasoning. To complete each assessment scenario, a model must locate an object specified in natural language. The agent controls an Unmanned Aerial Vehicle (UAV) by observing images obtained from successive locations of the UAV and providing text commands describing the next move.

FlySearch[1] aims to evaluate the extent to which VLMs can autonomously reason and explore, as well as characterize the limitations of their current capabilities. In FlySearch, the agent (tested method) controls a UAV flying over urban or natural environments to search for objects of interest, such as specific cars, fires, lost people waiting to be rescued, or piles of garbage (see Figure 1). It is built from scratch using the realistic Unreal Engine 5 [15], widely used for video games, providing dynamic 3D scenes with many assets. Moreover, procedural generation allows the generation of an unlimited number of scenarios with varying environmental characteristics, such as time of day, forest density, and UAV launch altitude. The benchmark runs efficiently in headless mode on both modern consumer-grade GPUs (NVIDIA RTX) and standard deep learning clusters (A100/H100), lowering the entry barrier for researchers and practitioners.

FlySearch consists of 3 standardized scenario sets with varying levels of difficulty. **FS-1** tests basic perception and navigation skills, **FS-Anomaly-1** additionally probes the capability of agents to understand the context of the environment, while **FS-2** requires executing a consistent exploration strategy involving a large number of steps to find the object of interest. We evaluate 3 closed-weight and 6 open-weight models on these scenarios and compare the results to scores obtained by humans. We find that VLMs strongly underperform humans on both FS-1 and FS-2. Additionally, while the performance drop between FS-1 and FS-2 is relatively low for humans ($\approx 9\%$), it is massive for VLMs ($\approx 90\%$). Therefore, despite possessing basic navigation and visual comprehension skills, current VLM models fail to form and execute proper exploration strategies, even after GRPO-based fine-tuning.

Our contributions can be summarized as follows:

- We release two high-fidelity outdoor environments built with Unreal Engine 5, enabling realistic and scalable evaluation of embodied agents in complex, unstructured settings.
- We define a suite of object-based exploration challenges designed to isolate and measure the exploration capabilities of VLMs and humans in open-world scenarios.
- We benchmark several popular VLMs in a zero-shot setting and identify consistent failure modes across vision, grounding, and reasoning. Our analysis reveals that these limitations persist even with fine-tuning, suggesting fundamental gaps in current VLM architectures.

2 Related work

ObjectNav. Our environment is focused on Object-Goal Navigation (ObjectNav or ObjectGoal) task, where the goal is to navigate to a specific object type in a given environment [3, 6]. Using this environment, we instantiate challenges that tie into Language-Driven Zero-Shot ObjectNav [12, 17, 39] tasks, as we expect tested methods to be able to perform search for an arbitrary text-based

[1] https://github.com/gmum/FlySearch

Table 1: **Other benchmarks:** We compare FlySearch with other benchmarks focused on VLM evaluation and ObjectNav challenges. We denote cases of partial criterion satisfaction with ✔.

Environment	Photorealistic	Outdoor	Exploration-focused	3D	Focus on VLM eval
Habitat Nav. Challenge [66]	✔	✗	✔	✔	✗
RoboTHOR [11]	✗	✗	✔	✔	✗
AgentBench [35]	✗	✗	✗	✗	✔
LMAct [51]	✗	✗	✗	✗	✔
SmartPlay [63]	✗	✔	✗	✗	✔
BALROG [45]	✗	✔	✗	✗	✔
VisualAgentBench [36]	✔	✔	✗	✔	✔
OpenEQA [40]	✔	✗	✗	✔	✔
FlySearch (ours)	✔	✔	✔	✔	✔

object description. Currently, Habitat-Sim [52, 66], AI2-THOR [11, 25], Gibson Env [64] are the most commonly used environments for the ObjectNav task [57]. All of these environments are indoor-focused, while in FlySearch the agent is embodied within a UAV in an outdoor scenario.

VLN. There also exists a related broad field of vision-and-language navigation (VLN), mainly concerned with embodied agents following natural language instructions that specify a route from point A to point B [70]. We note the existence of outdoor UAV-centric environments for VLN tasks, such as AerialVLN [34], TRAVEL [61] or CityNav [28]. AerialVLN is an example of a VLN task with high-level UAV control and a simulated environment, which follows the above definition. TRAVEL aims to make drone-based VLN more low-level by modifying the agent's action space for direct UAV control. CityNav introduces a VLN environment based on scans of real-life cities from SensatUrban [21] (instead of simulating them). These papers focus on instruction following in navigation (i.e., VLN), while our benchmark puts emphasis on finding objects without further instructions (i.e., ObjectNav).

AirSim and OUTDOOR. Microsoft AirSim [54] is an Unreal Engine 4 plugin that can be used to create UAV simulators, which can also be used for ObjectNav tasks in an outdoor setting. However, to the best of our knowledge, only OUTDOOR [65] considers such application of this software, as ObjectNav is generally dominated by indoor environments. However, even though OUTDOOR considers using UAVs, the action space is still 2-dimensional (horizontal movement only), while FlySearch has a 3-dimensional exploration space (the UAV can also move up and down). This emphasises dynamic altitude control to manage uncertainty and avoid redundant exploration, as the level of visual detail and the size of the visible area varies depending on the altitude. As such, success in FlySearch requires an efficient, emergent search strategy, where the agent must reason about where the object is likely to be, when looking from high altitude. Overall, this is a much more complex problem, requiring pre-existing knowledge of the real world to understand the contextual cues to exploit a wide field of view at high altitudes and fly low only in promising areas.

AVE. Additionally, the problem of exploration is considered from a different perspective in Active Visual Exploration (AVE) tasks [46, 47, 53], where a stationary agent is provided with partial observations derived from a large static image, simulating a limited field of view.

VLM benchmarks. Vision question-answering (VQA) datasets that contain pairs of images and textual questions are among the most popular approaches to assess the capabilities and limitations of VLMs. There are many examples of VQA-oriented benchmarks, such as [19, 23, 37, 41, 56, 68]. However, to fully evaluate VLM's performance, benchmarks that can measure their performance in practical tasks are needed [22, 33]. Recently, benchmarks that require interaction with an environment have gained popularity in the context of evaluating the capabilites of different VLMs. For example, BALROG [45], VisualAgentBench [36] and LMAct [51] evaluate VLM-based agents on several different environments with visual and textual representations. MineDojo [16] can be used to test different VLM-based agents in an interactive, 3D environment based on the popular *Minecraft* game. Other examples of benchmarks include SmartPlay [63] and AgentBench [35], although these are limited only to the textual modality (with images being "encoded" into natural language). There are also examples of embodied question-answering benchmarks for VLMs, such as OpenEQA [40]. We differ from other benchmarks by specifically focusing on gauging VLMs in the ObjectNav, which allows us to evaluate exploration capabilities of VLMs with a high degree of independence from their other capabilities.

https://doi.org/10.52202/085713-5594

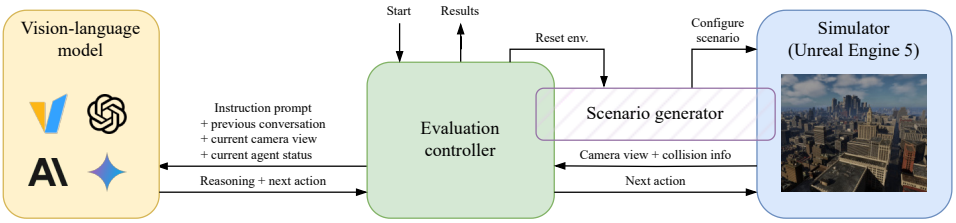


Figure 2: **Evaluation pipeline:** FlySearch consists of three parts besides the vision-language model. The simulator renders near-photorealistic views of a large open-world map and handles basic physics such as collisions. The evaluation controller handles the communication between the evaluated vision-language model and the simulator and performs the evaluation. The scenario generator (functionally integrated with the controller and simulator) procedurally generates new evaluation scenarios.

VLMs in ObjectNav. Recently, we have seen a surge in applications of VLM foundation models in ObjectNav [9, 10, 12, 26, 67] and related tasks [38, 62, 72]. However, we note that these papers use VLMs as tools to create more intricate methods and they do not aim to measure the exploration capabilities of VLMs themselves, which are still largely unknown [57]. In this paper, we attempt to gauge VLMs' performance by treating FlySearch as a VLM benchmark meant to compare different models and to understand their shortcomings.

3 FlySearch

The goal of FlySearch is to evaluate the visual-spatial reasoning and information-gathering abilities of vision-language models. To achieve this, the model is given control of a simulated multirotor unmanned aerial vehicle equipped with a camera and tasked with finding an object described in natural language, see Figure 1.

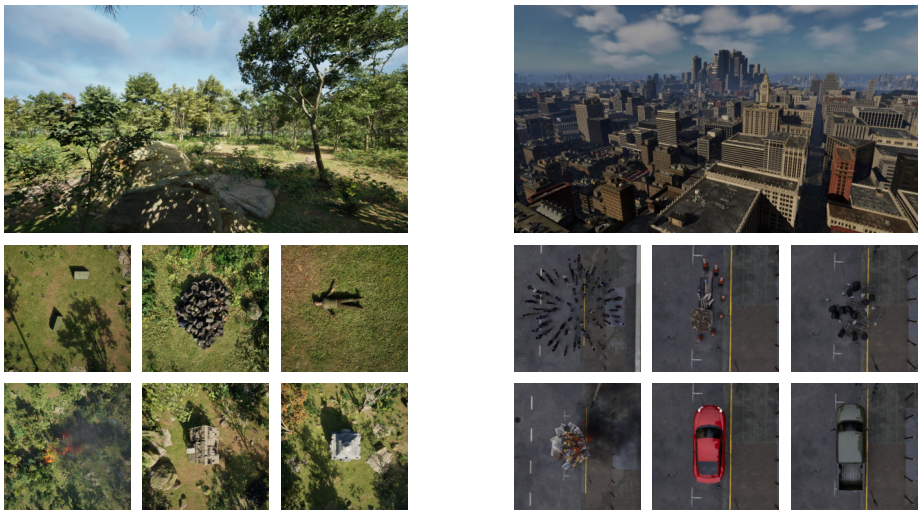
3.1 Evaluation task

Environment. The evaluation environment is a square outdoor area consisting of a fragment of a photorealistic procedurally generated map. The agent starts in the center of the area, at a randomly chosen altitude. Somewhere within the fragment is a target object. The agent must locate it within a limited number of steps and report success.

Starting prompt. The model is given a detailed prompt describing its task, including a brief textual description of the object to be located, e.g. *a red pickup truck*. The prompt also describes the communication format, including how to format the responses. We allow the model to preface each of its responses with a description of its reasoning, effectively allowing a chain of reasoning. We provide the prompt template in Appendix G.

Observation. At each exploration step, we provide the agent with a 500×500 pixel RGB image from the simulated UAV camera. To simplify the task, the camera is always facing the ground. The camera image is overlaid with a grid of coordinates to help the model understand movement directions and distances [29]. Additionally, we provide the agent with its height above the ground. In case of FS-2, we also provide the agent with the image specifying how searched object should look like from above; that is done to focus FS-2 more on search. All data except the images is provided in XML format.

Action. We assume that the simulated UAV is equipped with an autopilot system. Therefore, the agent does not need to provide any low-level control signals. Instead, it can focus on exploration by providing simple text commands with the tag `<action>(X, Y, Z)</action>`, where each of the coordinates represents a relative position change in meters in the corresponding direction. We introduce a collision avoidance system that stops the movement if an obstacle is detected within a 0.5 meter radius of the camera position, or if it tries to move out of the fly zone. When the agent decides that it has completed the task, it should respond with the `FOUND` text and end the exploration. If at any point the model response cannot be parsed, the episode is terminated.



(a) Forest environment. (b) City environment.

Figure 3: **Environments:** Our benchmark consists of two types of evaluation environments, forest and city. For each environment, we can generate an infinite number of procedurally generated test scenarios. The top row shows a preview of the environment, exhibiting the visual fidelity of the simulation. Below are top-down views of objects, matching the perspective of the agent.

Metrics. An episode is considered successfully completed (i.e., the object is found) if the object's center point is within the agent's camera field of view, the agent's position is at most 10 m above the highest point belonging to the object, and the agent returns the `FOUND` action. This metric is related to the one defined in [6], although we use the altitude difference and field of view instead of the Euclidean distance, as it is easier to estimate for human testers and VLMs – one only needs to make sure that the altitude is not higher than 10 m and the object is visible. We provide details on the implementation of the success criterion in Appendix F.

3.2 Evaluation pipeline

The FlySearch system consists of two main components: a simulator based on Unreal Engine 5 and a controller module. The controller is responsible for scenario generation, communication between benchmarked VLMs and the simulator, and result aggregation, as seen in Figure 2.

Simulator. To ensure realistic image input for the evaluated tasks, we chose Unreal Engine 5 as the simulation engine. It provides near-photorealistic graphics through real-time ray tracing and dynamic global illumination, while also supporting large, detailed open worlds. The platform is compatible with all major operating systems, and has an open-source codebase, enabling customization for machine learning applications. Additionally, its procedural content generation facilitates environment randomization by integrating parts of the scenario generator directly into the server, allowing us to place tens of thousands of object meshes in seconds. We also leveraged Unreal's extensive online marketplace of free assets to create evaluation scenarios. As such, these assets can be easily used to further extend the benchmark with new environments, scenarios, or objects of interest.

The simulator can be run using any modern consumer-grade graphics card as well as deep learning dedicated solutions (provided Vulkan is supported), and the engine runs in headless/offscreen mode (without a monitor). As such, we found it to perform well on standard computing clusters. Communication between the Unreal Engine simulator and the evaluation controller is handled via standard TCP/IP networking. The simulator-side implementation is provided as a native Unreal Engine plugin. We build upon the UnrealCV project [50], extending its functionality to allow full use of the aforementioned Unreal Engine 5 features.

Evaluation controller. The final component of FlySearch is the evaluation controller, implemented in Python. This module oversees the entire lifecycle of the benchmarking process, including setting up scenarios and calculating performance metrics. It also handles communication between the simulator

 https://doi.org/10.52202/085713-5594

and the evaluated vision-language model (VLM). Details of the prompts and message templates used in FlySearch are provided in Appendix G. FlySearch supports multiple VLMs, as described in Section 4.1. Additional models can be integrated easily by adding simple adapter code to the controller module or using the open-source vLLM inference server [27].

3.3 Evaluation environments

The benchmark consists of two distinct evaluation environment types, a forest and a city, see Figure 3. Each of them has its own set of target objects to find. **Forest environment** is based on the *"Electric Dreams Environment"* Unreal Engine product demo [14]. It consists of sparse forest scenery, including randomly placed rock formations. The map is procedurally generated entirely at runtime by the scenario generator. Moreover, all vegetation on the map is subject to wind changes. **City environment** is a large, modern, American-style city, based on the *"City sample"* Unreal Engine demo [13]. The city layout is a large, semi-procedurally generated map of roughly 4×4 km. New maps can be generated with the tools provided at build time. Furthermore, random distractor assets (parked cars and walking pedestrians) are spawned by the scenario generator at runtime.

4 Evaluation

FlySearch can be used to generate scenarios suited to the user's needs, e.g., finding cars at night in the city or locating stranded people in a very dense forest. However, to facilitate further research and enable fair comparisons, we propose three standardized challenges, i.e., sets of reproducible scenes that can be used to evaluate agents under the same conditions. Although standardized, it's important to highlight that these challenges are not static datasets, as the environment will dynamically react to the actions of the agent.

FS-1 contains 400 scenes of finding particular objects in the city and forest environments. The agent is explicitly instructed to find an object with a unique description in at most 10 actions. Certain distracting objects may appear, e.g., if we ask the model to find a yellow sports car, there might be cars of other colors or types in the scene. The goal of this challenge is to measure the general search capabilities of the model across a wide array of objects that might be encountered in various UAV applications, e.g., search-and-rescue missions and fire detection. The search area is limited to $400 \times 400 \times 120$ m, and the starting altitude to between 30 and 100 m, and the agent may not fly outside its field of view. We ensure, that the target object is within the field of view of the starting position and its center is not obscured by hard obstacles (it may not be distinguishable due to distance). The agent has to find one of the following:

- in the city: road construction works, crowd, large trash pile, fire, vehicle (variable type).
- in the forest: campsite, trash pile, person, forest fire, building.

In both scenarios, all target objects are positioned on the ground, in semantically correct locations, e.g., a car will be placed in a parking spot, not on the roof. All objects have multiple variations, randomly selected on scenario generation.

FS-Anomaly-1 is a set of 200 scenes in the city and the forest, where the agent is instructed to find an object that seems out of place, e.g., a giraffe in the city or a UFO in the forest. The goal of this challenge is to measure both the search capabilities of the models as well as their knowledge about what is and is not expected in certain environments. All other settings follow those of the previously mentioned FS-1. Anomalies are placed on the ground level. The anomalous objects to be located are:

- in the forest: UFO (flying saucer), small airplane, helicopter, large dinosaur, airliner,
- in the city: UFO (flying saucer), small airplane, helicopter, medium dinosaur, tank, giraffe.

FS-2 consists of 200 additional harder scenarios in the city environment. Base setting is the same as in FS-1; however, starting altitude range is raised to between 100 and 125 m and the search are is limited to (-starting altitude, +starting altitude) range in X and Y axis. Moreover, dynamic scene lightning is enabled, simulating different times of day and the maximum allowed altitude is 300 m. Most importantly, we allow the object to be obscured by obstacles (but still visible from the sky) and we allow the model to move beyond its field of view at each step to check models navigation capabilities.

Table 2: **Benchmark results:** Success rates ($\pm$ standard errors) of the evaluated models for FS-1 and FS-2 challenges. We observe that, overall, Gemini 2.0 Flash outperforms all evaluated models. Notably, small open-source models largely fail to solve the test scenarios, while larger models such as the Pixtral 124B achieve better performance. Note that the last row shows the model fine-tuned on the Forest environment (but not the specific FS-1 scenarios), which might lead to overfitting. To signal this, we mark the corresponding result gray.

Model	FS-1			FS-2
	Overall (%)	Forest (%)	City (%)	Overall (%)
Human (untrained)	–	–	66.7 ± 4.5	60.8 ± 6.9
GPT-4o	39.5 ± 2.4	45.5 ± 3.5	33.5 ± 3.3	3.5 ± 0.9
Claude 3.5 Sonnet	41.2 ± 2.5	$\mathbf{52.0 \pm 3.5}$	30.5 ± 3.3	$\mathbf{6.5 \pm 1.2}$
Gemini 2.0 flash	$\mathbf{42.0 \pm 2.5}$	42.5 ± 3.5	$\mathbf{41.5 \pm 3.5}$	6.0 ± 1.1
Phi 3.5 vision	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	–
InternVL-2.5 8B MPO	2.0 ± 0.7	2.5 ± 1.1	1.5 ± 0.9	–
Llava-Interleave-7b	0.8 ± 0.4	0.0 ± 0.0	1.5 ± 0.9	–
Qwen2.5-VL 7B	3.8 ± 1.0	6.0 ± 1.7	1.5 ± 0.9	0.0 ± 0.0
Qwen2-VL 72B	17.2 ± 1.9	16.5 ± 2.6	18.0 ± 2.7	–
Llava-Onevision 72b	9.5 ± 1.5	12.5 ± 2.3	6.5 ± 1.7	–
Pixtral-Large	29.8 ± 2.3	38.0 ± 3.4	21.5 ± 2.9	3.0 ± 0.8
Qwen2.5-VL 7B, GRPO on Forest	–	57.0 ± 3.5	27.0 ± 3.1	0.0 ± 0.0

4.1 Baselines

We evaluate a range of popular models. We select three proprietary models: OpenAI GPT-4o (2024-08-06 release) [2, 24], Anthropic Claude 3.5 Sonnet [4], and Google Gemini 2.0 flash (experimental) [18, 58]. Furthermore, we select four small open-weight models, with below 11B parameters: Phi-3.5-vision [1], InternVL2.5-8B-MPO [60], Llava-Interleave-Qwen-7B-dpo-hf [31], and Qwen2.5-VL 7B [5]. Finally, we select three more open-weight models with more than 11B parameters: Qwen2-VL-72B-Instruct [59], Llava-Onevision-Qwen2-72B-ov-hf [30], and Pixtral-Large-Instruct-2411 124B [43]. All models were selected based on their ability to process a full evaluation run, keeping all steps in context. During selection, we tested and rejected Llama-3.2 [42]. Although it architecturally supports handling multiple images, the publicly available model fails to form coherent responses when there is more than one image in the context.

Human study. Furthermore, we provide a human baseline for FS-1 City and FS-2 based on a user study of respectively 111 and 51 samples. The study was conducted using an online service, where participants were provided with the benchmark prompt and had to perform the same actions as the VLM. We provide more details about human baselines in Appendix D.

4.2 Results

FS-1. Table 2 contains the main aggregated results of our study. We find that the state-of-the-art VLMs achieve significantly worse results than non-trained humans on FS-1. While humans score 67% on average, the best-performing Gemini 2.0 manages to find the object in 42% of cases, with Claude and GPT-4o closely following. Large open-weight models fall behind significantly, with Pixtral, the best-performing model in this category, losing 10 percentage points on average to the proprietary models. Finally, the small open-weight models do not show any signal at all, as none of them exceeded 4%. We find that their poor performance can be largely attributed to their inability to follow instructions. Figure 4 shows that small models often do not claim that they have found the object even if it is within range at the end of the episode. As such, we exclude the small models from further analysis. We provide additional results, including measuring the impact of the action representation and the grid overlay, in Appendix H.

FS-2. Strikingly, the situation looks much different on FS-2. While on FS-1 humans outperformed best VLMs by 60%, on FS-2 this number is closer to 835%. We attribute this gap to the lack of systematic exploration abilities in VLMs. Since in FS-1 the object should be visible in the initial

https://doi.org/10.52202/085713-5594

Table 3: **FS-Anomaly-1 results:** Success rates ($\pm$ standard errors) of the evaluated models. Full results are in Appendix H.

Model	FS-Anomaly-1
	Overall (%)
GPT-4o	27.0 ± 3.1
Claude 3.5 Sonnet	27.5 ± 3.2
Gemini 2.0 flash	$\mathbf{35.5 \pm 3.4}$
Phi 3.5 vision	0.0 ± 0.0
InternVL-2.5 8B MPO	3.5 ± 1.3
Llava-Interleave-7b	0.0 ± 0.0
Qwen2.5-VL 7B	2.8 ± 1.2
Qwen2-VL-72B	7.5 ± 1.9
Llava-Onevision 72b	8.5 ± 2.0
Pixtral-Large	15.0 ± 2.5

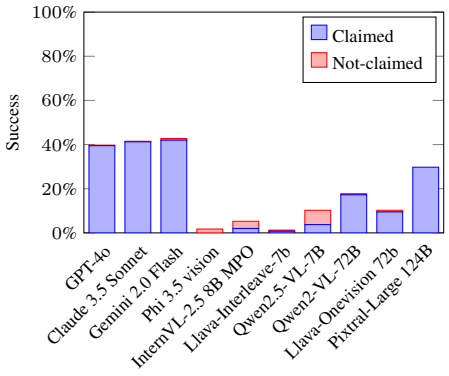


Figure 4: **Not claimed successes in FS-1:** In this figure we compare the number of cases, where the agent located the object, but failed to claim the `FOUND` action. We observe, that small models often fail to format the text output to report success.

Table 4: Performance of Gemini 2.0 Flash and Pixtral-Large in ablation studies. Surprisingly, increasing the number of steps might lead to worse performance. Additionally, explicitly specifying the object type in FS-Anomaly-1 improves the results as the task becomes easier.

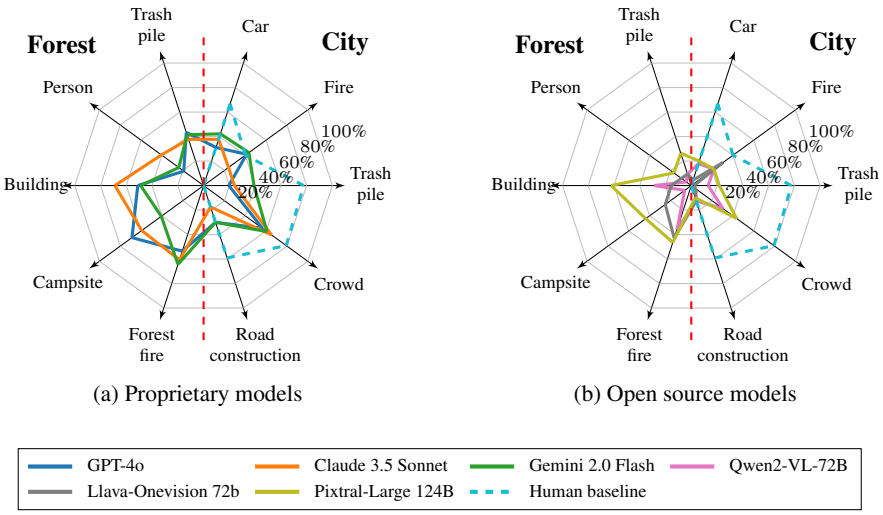
	Setting	City		Forest		Overall	
		Gemini	Pixtral	Gemini	Pixtral	Gemini	Pixtral
FS-1	5 steps limit	34.5%	15.5%	41.5%	33.5%	38.0%	24.5%
	10 steps limit (baseline)	**41.5%**	**21.5%**	42.5%	**38.0%**	**42.0%**	**29.8%**
	20 steps limit	33.5%	13.5%	**45.5%**	36.0%	39.5%	24.8%
FS-Anomaly-1	**Searching for an anomaly (baseline)**	25.0%	4.0%	46.0%	26.0%	35.5%	15.0%
	Searching for explicit object types	**34.0%**	**7.0%**	**59.0%**	**34.0%**	**46.5%**	**20.5%**

frame, it requires mostly object recognition and spatial reasoning abilities. On the other hand, since the object might not be within the field of view in FS-2, it requires the agent to implement a strategy that extensively explores the environment over multiple timesteps. While humans intuitively start following the streets in search of the object in question, VLMs mostly wander aimlessly in random directions. The problem is exacerbated due to difficulties with handling long contexts, see the episode length ablation in a later paragraph.

Fine-tuning results. We find that although some of these shortcomings can be addressed through simple fine-tuning, the problems with systematic exploration are more fundamental. We finetune Qwen2.5-VL-7B [5] using GRPO [55] on Forest environment in an offline mode, using a synthetic set of randomly generated flight trajectories, see details in Appendix E. The resulting model, presented in the last rows in Table 2, vastly improves upon the base model's performance on FS-1 City scenario, boosting the score by 14 times, from roughly 1.5% to 21.5%. However, the fine-tuning does not impact the results on FS-2 at all, where we still never see any successes.

Qualitative analysis. We perform further qualitative analysis of the larger models, see Figure 6 and Appendix J for example trajectories. By manually analyzing failed exploration trajectories on FS-1, we observe that even the most advanced models struggle with spatial reasoning. For instance, when a model loses sight of an object, it often backtracks its moves or starts hallucinating rather than moving toward the object's last known location. In case of FS-2, these issues are aggravated by the additional need of carrying out a systematic search. For example, GPT-4o often flies to the ground and hallucinates the existence of a searched object, whilst not performing any reasonable exploration pattern at all. Moreover, all models fail to handle collisions properly, often attempting to redo the same action when it previously resulted in hitting an object. In one of the trajectories, Gemini states "*I am unable to navigate without hitting the building. I guess I will give up.*"

Object-specific analysis. In Figure 5 we zoom in on the specific object classes that appear in FS-1. As expected, classes with large objects, such as buildings have a higher success rate than those with



(a) Proprietary models (b) Open source models

Figure 5: **Success rate per class:** In this figure, we show the model success rate per each target class for both the Forest and the City environments. Classes with large visible objects such as *Building* for Forest or *Crowd* for City are significantly easier for most models to locate. On the other hand, classes that are difficult to distinguish from the background, such as both garbage classes, are only located by more advanced models. Human baseline for City is marked with dashed line.

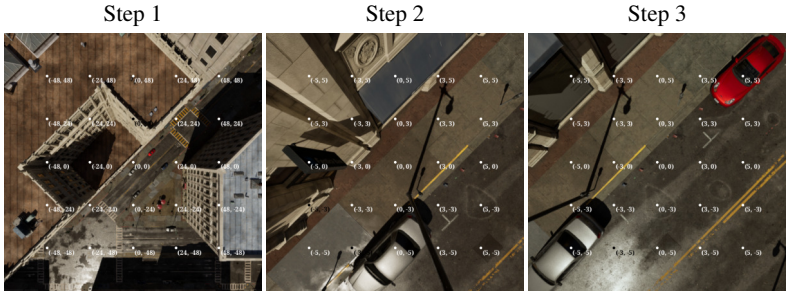


Figure 6: **Example of a successful trajectory in FS-1:** GPT-4o navigates to *red sports car* object by first descending and then moving to the right. The first row shows the model's visual inputs, and the second actions it has taken. Note the presence of the grid overlay on images. Best viewed zoomed in.

smaller figures, such as single persons. However, even finding a building poses a significant challenge since the best-performing Claude 3.5 Sonnet does not manage to find approximately 30% of objects in this class. Interestingly, VLMs tend to be more successful at finding trash piles in the forest rather than in the city, where they are more visible from a distance. Lastly, the road construction site class is one of the hardest classes, even though the name provides a clue as to where the model should look.

FS-Anomaly-1. The results for FS-Anomaly-1, presented in Table 3 are on average significantly lower than in FS-1, suggesting that the models struggle with the additional challenge of figuring out which object is out of place. Indeed, as we confirm on a subset of the models, the performance substantially increases once we explicitly name the anomaly object to be found, see Appendix H. Without explicit instructions, more often than not, VLMs find one of many typical, although visually distinct objects to be out of place, while ignoring obvious anomalies. For example, in one of the city scenarios, all closed-source VLMs wrongly identified a *yellow taxi* to be the anomaly, ignoring a *tank* standing next to it. Moreover, we find that the models sometimes tend to misidentify the anomaly objects as something more expected in a given situation, e.g., one of the models assumed that a giraffe walking around city streets is a dog.

https://doi.org/10.52202/085713-5594

Impact of the number of steps. Finally, we study the impact of changing the length of each episode in FS-1, i.e., the number of actions that can be taken before the trajectory automatically ends in failure. We find that reducing the number of steps from the baseline 10 to 5 reduces the results by 10% (4pp) for Gemini and 17% (5pp) for Pixtral. More interestingly, we also observe performance deterioration as we increase the step limit to 20 – Gemini's performance falls by 6% (2.5pp) and Pixtral's by 17% (5pp). Breaking these results by category, we find that increasing the number of steps only leads to significant deterioration in the visually cluttered City environment. We discover that models tend to fail when expected to reason and gather information over longer timeframes, which we also observed in FS-2. See more details in Table 4 and Appendix H.

5 Conclusion

In this paper, we introduce FlySearch, a dynamic benchmark designed for evaluating exploration capabilities of VLMs. Navigating three-dimensional environments and finding objects of interest are everyday real-world tasks that remain underrepresented in VLM benchmarking. To address this gap, FlySearch leverages Unreal Engine 5, a highly realistic video game engine to procedurally generate scenarios of searching for objects in urban and natural environments. Using the three standardized challenges, FS-1, FS-Anomaly-1, and FS-2, we show that VLMs underperform compared to human baseline, especially when it comes to more complex exploration tasks. At the same time, this study has certain limitations that offer interesting directions for future work. In this paper, we purposefully avoid testing more sophisticated ObjectNav methods [9, 26, 67], since our main focus lies in understanding pure VLM capabilities. At the same time, checking their performance in FlySearch could bring interesting insights to the field. Additionally, we use a simple prompting technique, and one could possibly get better results out of VLMs by leveraging few-shot learning [8, 48] or prompt optimization tools [49, 73].

Acknowledgments

This paper has been supported by the Horizon Europe Programme (HORIZONCL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. This research was funded by National Science Centre, Poland (grant no. 2023/50/E/ST6/00469 and Sonata Bis grant no 2024/54/E/ST6/00388). The research was supported by a grant from the Faculty of Mathematics and Computer Science under the Strategic Programme Excellence Initiative at Jagiellonian University. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017483. Some experiments were performed on servers purchased with funds from the Priority Research Area (Artificial Intelligence Computing Center Core Facility) under the Strategic Programme Excellence Initiative at Jagiellonian University.

References

[1] M. Abdin, J. Aneja, H. Awadalla, et al. Phi-3 technical report: A highly capable language model locally on your phone. Technical Report MSR-TR-2024-12, Microsoft, August 2024.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report, 2024.

[3] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[4] Anthropic. Introducing claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`, Jun 2024.

[5] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[6] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020.

[7] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv.2410.24164*, 2024.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[9] W. Cai, S. Huang, G. Cheng, Y. Long, P. Gao, C. Sun, and H. Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5228–5234, 2024.

[10] Y. Cai, X. He, M. Wang, H. Guo, W.-Y. Yau, and C. Lv. Cl-cotnav: Closed-loop hierarchical chain-of-thought for zero-shot object-goal navigation with vision-language models. *arXiv preprint arXiv:2504.09000*, 2025.

[11] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, L. Weihs, M. Yatskar, and A. Farhadi. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3161–3171, Los Alamitos, CA, USA, Jun 2020. IEEE Computer Society.

[12] V. S. Dorbala, J. F. Mullen, and D. Manocha. Can an embodied agent find your "cat-shaped mug"? llm-based zero-shot object navigation. *IEEE Robotics and Automation Letters*, 9(5):4083–4090, May 2024.

[13] Epic Games. City sample. an overview of the features used to create the city sample learning example. `https://dev.epicgames.com/documentation/en-us/unreal-engine/city-sample-project-unreal-engine-demonstration`, Nov 2024.

[14] Epic Games. Electric dreams environment in unreal engine | unreal engine 5.5 documentation. `https://dev.epicgames.com/documentation/en-us/unreal-engine/electric-dreams-environment-in-unreal-engine`, Nov 2024.

[15] Epic Games. Unreal Engine 5.5. `https://unrealengine.com`, Nov 2024.

[16] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[17] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23171–23181, 2023.

[18] Google. Introducing gemini 2.0: Our new ai model for the agentic era. `https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/`, Dec 2024.

[19] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6325–6334, 2017.

[20] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[21] Q. Hu, B. Yang, S. Khalid, W. Xiao, N. Trigoni, and A. Markham. Sensaturban: Learning semantics from urban-scale photogrammetric point clouds. *International Journal of Computer Vision*, 130(2):316–343, 2022.

[22] J. Huang and J. Zhang. A survey on evaluation of multimodal large language models. *arXiv preprint arXiv:2408.15769*, 2024.

 https://doi.org/10.52202/085713-5594

[23] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019.

[24] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*, 2024.

[25] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[26] Y. Kuang, H. Lin, and M. Jiang. Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models. *arXiv preprint arXiv:2402.10670*, 2024.

[27] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA, 2023. Association for Computing Machinery.

[28] J. Lee, T. Miyanishi, S. Kurita, K. Sakamoto, D. Azuma, Y. Matsuo, and N. Inoue. Citynav: Language-goal aerial navigation dataset with geographic information. *arXiv preprint arXiv:2406.14240*, 2024.

[29] X. Lei, Z. Yang, X. Chen, P. Li, and Y. Liu. Scaffolding coordinates to promote vision-language coordination in large multi-modal models. *arXiv preprint arXiv:2402.12058*, 2024.

[30] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, and C. Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025.

[31] F. Li, R. Zhang, H. Zhang, Y. Zhang, B. Li, W. Li, Z. Ma, and C. Li. LLaVA-NeXT-Interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

[32] J. Li, D. Li, C. Xiong, and S. Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 17–23 Jul 2022.

[33] L. Li, G. Chen, H. Shi, J. Xiao, and L. Chen. A survey on multimodal benchmarks: In the era of large ai models. *arXiv preprint arXiv:2409.18142*, 2024.

[34] S. Liu, H. Zhang, Y. Qi, P. Wang, Y. Zhang, and Q. Wu. AerialVLN: Vision-and-Language Navigation for UAVs. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15338–15348, Los Alamitos, CA, USA, Oct 2023. IEEE Computer Society.

[35] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang, S. Zhang, X. Deng, A. Zeng, Z. Du, C. Zhang, S. Shen, T. Zhang, Y. Su, H. Sun, M. Huang, Y. Dong, and J. Tang. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*, 2023.

[36] X. Liu, T. Zhang, Y. Gu, I. L. Iong, S. XiXuan, et al. VisualAgentBench: Towards large multimodal models as visual foundation agents. In *The Thirteenth International Conference on Learning Representations*, 2025.

[37] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2025.

[38] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong. InstructNav: Zero-shot system for generic instruction navigation in unexplored environment. In *8th Annual Conference on Robot Learning*, 2024.

[39] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra. ZSON: zero-shot object-goal navigation using multimodal goal embeddings. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[40] A. Majumdar, A. Ajay, X. Zhang, P. Putta, S. Yenamandra, M. Henaff, S. Silwal, P. Mcvay, O. Maksymets, S. Arnaud, et al. OpenEQA: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16488–16498, 2024.

[41] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[42] MetaAI. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. `https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/`, 2024.

[43] Mistral. Pixtral large. `https://mistral.ai/news/pixtral-large/`, Jan 2025.

[44] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. In *International Conference on Machine Learning*, pages 37321–37341. PMLR, 2024.

[45] D. Paglieri, B. Cupiał, S. Coward, U. Piterbarg, M. Wolczyk, A. Khan, E. Pignatelli, Ł. Kuciński, L. Pinto, R. Fergus, J. N. Foerster, J. Parker-Holder, and T. Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *The Thirteenth International Conference on Learning Representations*, 2025.

[46] A. Pardyl, G. Rypeść, G. Kurzejamski, B. Zieliński, and T. Trzciński. Active visual exploration based on attention-map entropy. In E. Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 1303–1311, 8 2023. Main Track.

[47] A. Pardyl, M. Wronka, M. Wołczyk, K. Adamczewski, T. Trzciński, and B. Zieliński. Adaglimpse: Active visual exploration with arbitrary glimpse position and scale. In *Computer Vision – ECCV 2024*. Springer Nature Switzerland, 2024. Main Track.

[48] A. Parnami and M. Lee. Learning from few examples: A summary of approaches to few-shot learning. *arXiv preprint arXiv:2203.04291*, 2022.

[49] R. Pryzant, D. Iter, J. Li, Y. T. Lee, C. Zhu, and M. Zeng. Automatic prompt optimization with" gradient descent" and beam search. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[50] W. Qiu, F. Zhong, Y. Zhang, Z. X. Siyuan Qiao, T. S. Kim, Y. Wang, and A. Yuille. UnrealCV: Virtual worlds for computer vision. *ACM Multimedia Open Source Software Competition*, 2017.

[51] A. Ruoss, F. Pardo, H. Chan, B. Li, V. Mnih, and T. Genewein. LMAct: A benchmark for in-context imitation learning with long multimodal demonstrations. In *Forty-second International Conference on Machine Learning*, 2025.

[52] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

[53] S. Seifi, A. Jha, and T. Tuytelaars. Glimpse-Attend-and-Explore: Self-Attention for Active Visual Exploration. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16117–16126, Los Alamitos, CA, USA, Oct 2021. IEEE Computer Society.

[54] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In M. Hutter and R. Siegwart, editors, *Field and Service Robotics*, pages 621–635, Cham, 2018. Springer International Publishing.

 https://doi.org/10.52202/085713-5594

[55] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[56] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[57] J. Sun, J. Wu, Z. Ji, and Y.-K. Lai. A survey of object goal navigation. *IEEE Transactions on Automation Science and Engineering*, 22:2292–2308, 2025.

[58] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[59] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[60] W. Wang, Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, J. Zhu, X. Zhu, L. Lu, Y. Qiao, and J. Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024.

[61] X. Wang, D. Yang, Z. Wang, H. Kwan, J. Chen, W. Wu, H. Li, Y. Liao, and S. Liu. Towards realistic uav vision-language navigation: Platform, benchmark, and methodology. *arXiv preprint arXiv:2410.07087*, 2024.

[62] P. Wu, Y. Mu, K. Zhou, J. Ma, J. Chen, and C. Liu. Camon: Cooperative agents for multi-object navigation with llm-based conversations. *arXiv preprint arXiv:2407.00632*, 2024.

[63] Y. Wu, X. Tang, T. Mitchell, and Y. Li. Smartplay : A benchmark for LLMs as intelligent agents. In *The Twelfth International Conference on Learning Representations*, 2024.

[64] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[65] Q. Xie, T. Zhang, K. Xu, M. Johnson-Roberson, and Y. Bisk. Reasoning about the unseen for efficient outdoor object navigation. *arXiv preprint arXiv:2309.10103*, 2023.

[66] K. Yadav, J. Krantz, R. Ramrakhya, S. K. Ramakrishnan, J. Yang, A. Wang, J. Turner, A. Gokaslan, V.-P. Berges, R. Mootaghi, O. Maksymets, A. X. Chang, M. Savva, A. Clegg, D. S. Chaplot, and D. Batra. Habitat challenge 2023. `https://aihabitat.org/challenge/2023/`, 2023.

[67] B. Yu, H. Kasaei, and M. Cao. L3MVN: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3554–3560, 2023.

[68] W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, and L. Wang. MM-Vet: evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[69] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao. VinVL: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588, June 2021.

[70] Y. Zhang, Z. Ma, J. Li, Y. Qiao, Z. Wang, J. Chai, Q. Wu, M. Bansal, and P. Kordjamshidi. Vision-and-language navigation today and tomorrow: A survey in the era of foundation models. *Transactions on Machine Learning Research*, 2024. Survey Certification.

[71] Y. Zhao, J. Huang, J. Hu, X. Wang, Y. Mao, D. Zhang, Z. Jiang, Z. Wu, B. Ai, A. Wang, W. Zhou, and Y. Chen. SWIFT: A scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29733–29735, Apr. 2025.

[72] G. Zhou, Y. Hong, and Q. Wu. NavGPT: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7641–7649, 2024.

[73] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2022.

 https://doi.org/10.52202/085713-5594

A Impact Statement

In this paper, we focus on the abstract problem of visual exploration – how to interact with the environment to locate the objects of interest. To evaluate this ability in VLMs, we propose a benchmark designed around flying a UAV to find objects in urban and natural environments. This is relevant to real-world applications with positive societal impact, such as search-and-rescue missions, forest fire detection, or personal assistance. However, autonomous UAVs also pose risks as they can be misused by bad agents for surveillance or military operations. We implore users to use their best judgment for the use of the benchmark. Our benchmark is not designed for surveillance or military applications. We urge other researchers to ensure that their benchmarks (including FlySearch derivatives) and models are not directly or indirectly optimized for malicious objectives. We encourage researchers to evaluate their systems for potential biases or unintended behaviors that could lead to misuse. Finally, we would like to highlight the importance of regulatory oversight and the need for clear ethical guidelines in the deployment of autonomous UAVs.

B Funding Transparency Statement

Funding. All financial activities supporting the submitted work are listed in the Acknowledgments section of the main paper.

Competing Interests. At the time of the publication, Maciej Wołczyk is working at Google. However, he did the majority of the work on this paper while at IDEAS NCBR and has not been involved in benchmarking Google models. The authors declare that they have no competing interests and no financial relationships with any entity that could be perceived as influencing the content of this work.

C Benchmark specification

In this section we present the parameters of benchmark scenario generation and evaluation details.

C.1 Scenario parameters

Table S1: **Parameters in scenario generation:** In this table, we present value ranges for variables that have an impact on the search trajectory. We sample from them uniformly at random while creating a scenario. FS-Anomaly-1 has identical value ranges to FS-1, except for the searched object type and searched object asset. Furthermore, in FS-1 and FS-Anomaly-1 there will be a clear line of sight from the starting position of the agent to the object. In FS-2, we relax this condition and only validate that the object is not under an obstacle (e.g. is not hidden under a bridge, but can be behind a building). Lastly, objects can be placed anywhere in the forest environment and in one of 31852 semantically correct locations in the city environment.

Parameter	FS-1		FS-2
	Value range (Forest)	Value range (City)	Value range (City)
Seed	$\mathbb{N} \cap [0, 1000000000]$	$\mathbb{N} \cap [0, 1000000000]$	$\mathbb{N} \cap [0, 1000000000]$
Agent's starting height (h) [m]	$\mathbb{N} \cap [30, 100]$	$\mathbb{N} \cap [30, 100]$	$\mathbb{N} \cap [100, 125]$
Agent's starting position offset [m]	$[-0.5h, 0.5h]$	$[-0.5h, 0.5h]$	$[-0.95h, 0.95h]$
Searched object type	{campsite, trash, person, fire, building}	{construction works, crowd, trash, fire, car}	
Searched object coordinates	Anywhere on the map	38152 possible placements	
Searched object asset	Dependent on the object type	Dependent on the object type	
Sun elevation angle	$[10, 90]^\circ$ over horizon	45° over horizon	$[10, 90]^\circ$ over horizon
Sun azimuth angle	$[0, 360]^\circ$	110°	$[0, 360]^\circ$
Tree density	$[0.0, 0.3]$	N/A	N/A
Rock density	$[0.0, 0.1]$	N/A	N/A
Object visibility	Visible from agent's starting position		Not under an obstacle

We present all variables that can be used to generate a new scenario in Table S1, such as whether the searched object should be visible from the UAV's initial location. In Table S2, we describe additional parameters that do not impact the scenario generation, but govern details of evaluation – such as:

- Search area bounds – a rectangle centered on agent's initial position, describing bounds inside of which agent can move. Note that this only serves as a limitation on agent's movement and does not impact the scenario generation process,
- Maximum altitude – if an agent's action would bring it above that threshold, it is considered invalid and the agent is asked to issue a new instruction,
- Whether agent can move beyond its *current* view – in FS-1 and FS-Anomaly-1 the agent is prevented from leaving visible area, discouraging hallucination,
- Number of available actions,
- Number of possible *consecutive* retries if agent's action is considered invalid and needs to be redone,
- Whether the agent should also receive an image showcasing how the target object should look like from above.

Table S2: **Additional parameters:** In this table, we present configurations used during evaluations in FS-1, FS-Anomaly-1 and FS-2. Number of possible retries in case of invalid action was introduced to prevent LLMs from "hanging" by constantly performing invalid actions. As such, this limit was not present while evaluating human baselines. Similarly, we allow humans to fly out of their current view even in FS-1. We note that FS-1 and FS-Anomaly-1 use the exact same configuration, while more search-oriented FS-2 uses a slightly different one. In this table, h denotes starting height of the UAV.

Parameter	FS-1 / FS-Anomaly-1	FS-2
Search area bounds [m]	400×400	$2h \times 2h$
Maximum altitude [m]	120	300
Agent can move beyond its current view	No (Yes for humans)	Yes
Number of available actions	10	20
Number of possible retries if action is invalid	5 (∞ for humans)	5 (∞ for humans)
Object type specification modality	Text	Text + Image

C.2 Target details

The targets in the City environment are as follows:

- Road construction works – a construction zone at the side of a road,
- Crowd – a group of over 30 randomly generated people standing close together,
- Large trash pile – a random pile of trash (bags, car tires, barrels, metal sheets, etc.),
- Fire – a burning car or pile of trash with fire and smoke,
- Car – a randomized car or truck with a specific color (we provide the agent with the color and type).

While in the Forest the goal is to find:

- Campsite – a randomly generated set of camping equipment, including at least one tent,
- Trash pile – a pile of car tires, barrels, metal scrap, and other waste (different than in the forest environment),
- Person – a human laying flat on the ground, representing an injured hiker,
- Forest fire – a burning forest area, emitting a large cloud of smoke,
- Building – an abandoned building in various styles.

D Human baselines

In order to evaluate human performance on FlySearch we provide a web-based interface for human testers. The interface is designed to be consistent with VLM evaluation procedure, while allowing for efficient human interaction. Therefore, instead of using XML formatted text for communication, the interface uses standard HTML forms and buttons. In Fig. S1 the welcome screen is presented.

https://doi.org/10.52202/085713-5594

The user is shown the model prompt (with XML formatting instructions omitted) and can select the between FS-1 and FS-2. In Fig. S2 and Fig. S3 the communication screens for FS-1 and FS-2 respectively are shown. On each step the evaluator has to either fill in X, Y, Z action coordinates and press the `MOVE` button, or if they believe they have fulfilled the task criterion they have to press the `FOUND` button. Finally, after clicking `FOUND` or exhausting possible moves the user is notified if they have succeeded or failed the scenario. During the entire evaluation, users can report any issues with a built-in bug report form.

Source code of the interface and associated back-end server is included in FlySearch codebase. The study was conducted with participation of human manual software testers (employees). The study did not pose any risks to participants. No Personally Identifiable Information was gathered in the process. The study was conducted according to institutional regulations on data gathering and processing and was subject to internal approval process.

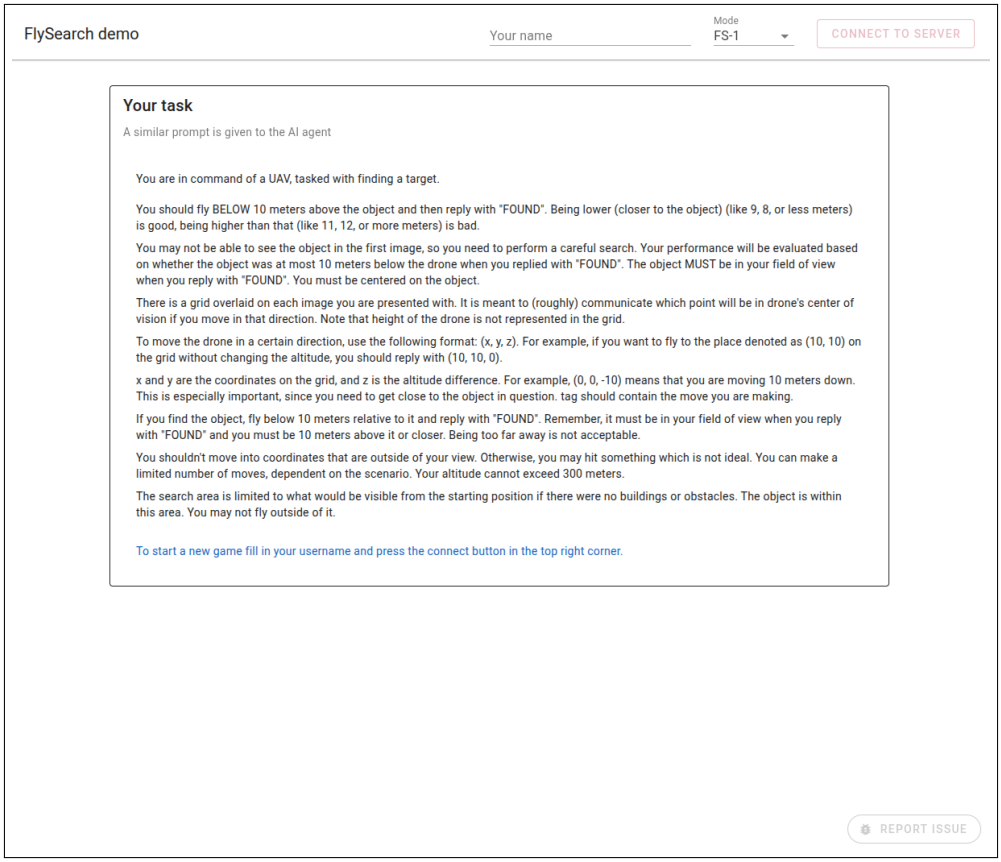


Figure S1: **Human study – start screen**: Screenshot of the FlySearch human interface welcome screen, containing instructions derived from the benchmark prompt. The screen also contains a field for the participant identifier (nickname), a scenario generator switch (FS-1/FS-2), connection button and bug report button.

E Fine-tuning details

To provide a reasonable baseline score for fine-tuning VLMs for spatial reasoning we train the Qwen VL 2.5 7B [5] model on the Forest environment and evaluate it on the City environment in both FS-1 and FS-2. The Qwen VL 2.5 7B model is selected as it is the best performing model in the *small open-source model* category in our evaluation. Since the City and Forest environments share almost no graphical assets and are vastly different visually, training the model on Forest allows for a fair comparison with other models on the City environment. That is, the model cannot learn to visually recognize relevant objects and their placement in the evaluated scenario.

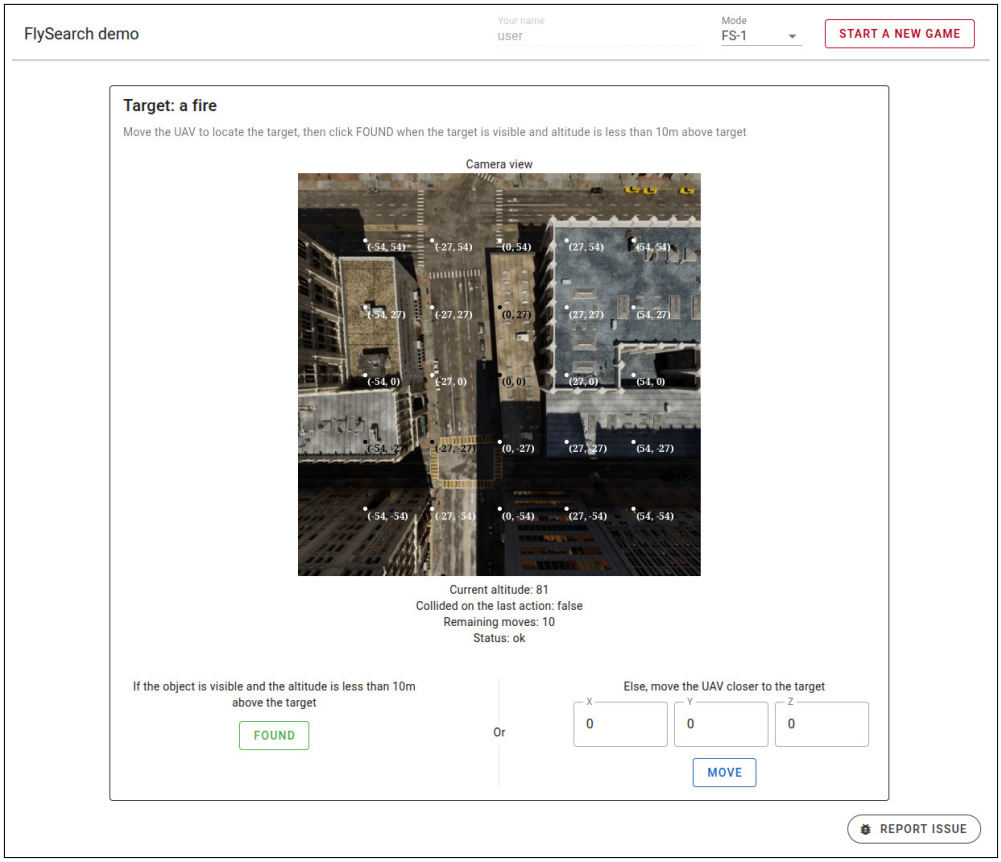


Figure S2: **Human study – FS-1 screen**: Screenshot of the FlySearch human interface FS-1 screen. The page contains the target description, agent camera view, altitude, object and boundary collision information, and action controls. The image is overlaid with grid coordinates, exactly as the image provided to the evaluated VLM. All state and action elements are the same as those provided to the model, but for the formatting (a form instead of XML formatted text).

Using the Forest environment, we generate 6750 unique exploration scenarios. In each scenario, we pre-record a flight trajectory from the starting point to the target. The trajectory is created by sampling 30 evenly spaced steps over a straight line. To each flight step a random position offset in range of $[-10, 10]$ m, and capture camera views using our simulator. Finally, we augment the dataset by generating 10 new trajectories from each captured episode, randomly dropping some of the flight steps. Resulting trajectories have between 1 and 10 steps, depending on the distance between start and target. Each trajectory is further cut at random step and formatted as a conversation to be completed by the VLM, resulting in 67500 training samples.

Standard supervised fine-tuning does not provide sufficient improvement in results (11.1% on FS-1 City, compared to 1.5% of the base model). This is likely due to the fact, that each scenario can be solved in multiple ways, not just the ideal trajectory. Therefore, we apply GRPO fine-tuning [55] in step-wise, offline manner. That is, using the pre-generated training dataset we train the model to predict one next action. We define the reward function for the step with the following pseudo-code:

Listing 1: Reward for GRPO fine-tuning (pseudo-code).

```
def reward(model_output):
    if model_output is not parsable:
        return 0

    reasoning, action = parse(model_output)

    reasoning_reward = min(1, len(reasoning) / 100)
```


https://doi.org/10.52202/085713-5594

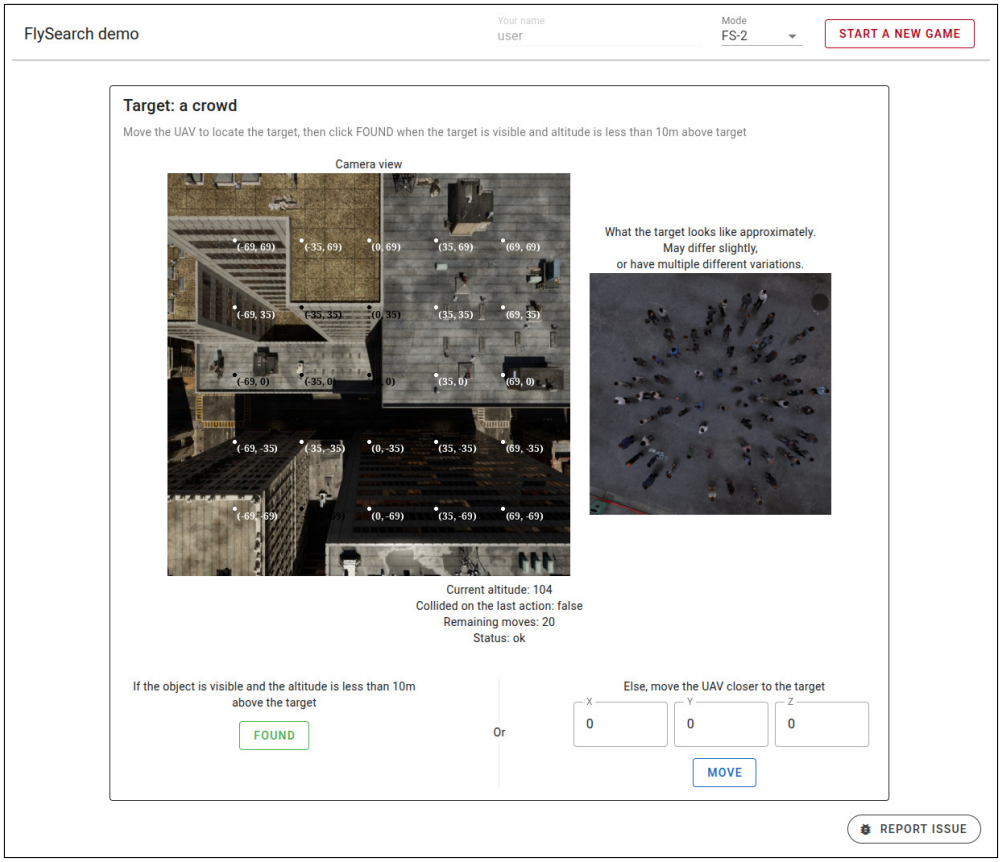


Figure S3: **Human study – FS-2 screen**: Screenshot of the FlySearch human interface FS-2 screen. The page contains the same elements as for FS-1 plus an additional visual description of the target (a neutral background and lightning picture, not an actual image of the searched object).

```
if action == 'FOUND':
    if target_located:
        return 1 + reasoning_reward
    else:
        return 0 + reasoning_reward

action_reward = (current_distance - next_distance) / current_distance
action_reward = max(min(action_reward, 1), 0) # clip between 0 and 1

if target_located:
    # action should be FOUND, decrease reward for further moves
    mod = max(0., (altitude - targed_height compare)) / 10
    action_reward = (mod ** .5) * action_reward

return reasoning_reward + action_reward
```

Therefore, the model is incentivized to both provide at least 100 tokens of reasoning output and to move closer to the target in each step. When the model can respond with `FOUND` action the reward for further moves towards the target is decreased. Finally, if the agent responds with `FOUND` it receives a binary reward based on the success.

We perform the GRPO fine-tuning using the Swift library [71], and use LoRA [20] to speed up the training. Moreover, we freeze the vision encoder part of the model to prevent overfitting on visual features. Bellow are the full parameters of the training:

Listing 2: GRPO fine-tuning command.

```
NPROC_PER_NODE=2 CUDA_VISIBLE_DEVICES=0,1,2,3 swift rlhf \
  --rlhf_type grpo \
  --model Qwen/Qwen2.5-VL-7B-Instruct \
  --train_type lora \
  --dataset $DATA_PATH \
  --num_train_epochs 1 \
  --per_device_train_batch_size 8 \
  --per_device_eval_batch_size 8 \
  --learning_rate 1e-5 \
  --gradient_accumulation_steps 1 \
  --eval_steps 100 \
  --save_steps 100 \
  --save_total_limit 2 \
  --logging_steps 5 \
  --dataloader_num_workers 4 \
  --attn_impl flash_attn \
  --use-hf \
  --torch_dtype bfloat16 \
  --deepspeed zero2 \
  --target_modules all-linear lm_head \
  --lora_alpha 32 \
  --lora_rank 8 \
  --external_plugins rewardplugin.py \
  --reward_funcs fly_search \
  --max_completion_length 1024 \
  --warmup_ratio 0.05 \
  --num_generations 16 \
  --dataset_num_proc 4 \
  --temperature 0.9 \
  --log_completions true \
  --use_vllm true \
  --vllm_gpu_memory_utilization 0.9 \
  --vllm_limit_mm_per_prompt '{"image": 10}' \
  --num_infer_workers 2 \
  --async_generate true \
  --seed 42 \
  --split_dataset_ratio 0
```

The entire process takes several hours using 4 NVIDIA H100 GPUs. After fine-tuning the model is evaluated with standard FS-1 City and FS-2 settings, achieving 27.0% and 0.0% accordingly. Moreover, it achieves 57.0% accuracy on the Forest environment it was trained upon.

F Success criterion implementation

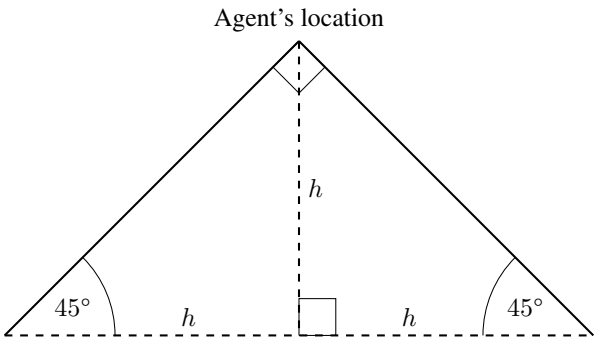


Figure S4: **Success criterion illustration:** The searched object's center must be inside of the camera's cone of view. We assume the camera's field of view is $90°$.

https://doi.org/10.52202/085713-5594

For a trajectory to be considered successful, at the end of it the agent must be located less than 10 meters above the object's highest point. Furthermore, the object must be seen from the agent's location.

To check whether the agent can see the object, we calculate agent's cone of view and check whether object's center is inside of the cone of view. This calculation is straightforward, because camera's field of view is $90°$. It easily follows that agent at altitude h sees an area of $2h \times 2h$ meters. We provide a drawing to illustrate that fact in Fig. S4.

In some cases, this implementation may cause misclassifications as a failure, due to the agent „seeing" an object in a human sense (e.g. a significant part of it's edges), but not having object's center in its cone of view. To avoid this issue, we specify in the prompt that agent should be *centered* on the object.

G Prompt templates

In this section we present the prompt templates for all benchmark tasks.

G.1 Prompt template for FS-1 and FS-Anomaly-1

Listing 3: Prompt template for FS-1 and FS-Anomaly-1.

```
<Context>
    You are in command of a UAV, tasked with finding {TARGET}.
</Context>

<Objective>
    You should fly BELOW 10 meters above the object and then reply with "FOUND".
     Being lower (closer to the object) (like 9, 8, or less meters) is good, being
     higher than that (like 11, 12, or more meters) is bad.

    You may not be able to see the object in the first image, so you need to perform
      a careful search. Your performance will be evaluated based on whether the
     object was at most 10 meters below the drone when you replied with "FOUND". The
      object MUST be in your field of view when you reply with "FOUND". You must be
     centered on the object.
</Objective>

<Coordinates>
    There is a grid overlaid on each image you are presented with. It is meant to (
     roughly) communicate which point will be in drone's center of vision if you
     move in that direction. Note that height of the drone is not represented in the
      grid.
</Coordinates>

<Controls>
    <Action space>
        To move the drone in a certain direction, use the following format:
        <Action>(x, y, z)</Action>. For example, if you want to fly to the place
     denoted as (10, 10) on the grid without changing the altitude, you should reply
      with <Action>(10, 10, 0)</Action>.

        x and y are the coordinates on the grid, and z is the altitude difference.
     For example, <Action>(0, 0, -10)</Action> means that you are moving 10 meters
     down. This is especially important, since you need to get close to the object
     in question.

    </Action space>

    <Formatting>

        Your each response should contain XML <Reasoning> tag and <Action> tag.
        <Reasoning> tag should contain your reasoning for the move you are making.
```

```
      <Action> tag should contain the move you are making.

      If you find the object, fly below 10 meters relative to it and reply with "
    FOUND". Remember, it must be in your field of view when you reply with "FOUND"
    and you must be 10 meters above it or closer. Being too far away is not
    acceptable.

      For example:

      <Reasoning>This yellow point might be the object in question. I need to go
    lower to check for that. If it's not the object in question, I will continue
    the search. I will also slightly go to the north.</Reasoning>
      <Action>(5, 0, -30)</Action>

   </Formatting>

   <Limitations>
      You shouldn't move into coordinates that are outside of your view. Otherwise,
     you may hit something which is not ideal.
      You can make at most 9 moves. Your altitude cannot exceed 120 meters. Your
    search area is 400x400m from the drone's starting position.
   </Limitations>
</Controls>
```

G.2 Prompt template for FS-2

Listing 4: Prompt template for FS-2.

```
<Context>
   You are in command of a UAV, tasked with finding {TARGET}.
</Context>

<Objective>
   You should fly BELOW 10 meters above the object and then reply with "FOUND".
    Being lower (closer to the object) (like 9, 8, or less meters) is good, being
    higher than that (like 11, 12, or more meters) is bad.

   You may not be able to see the object in the first image, so you need to perform
     a careful search. Your performance will be evaluated based on whether the
    object was at most 10 meters below the drone when you replied with "FOUND". The
     object MUST be in your field of view when you reply with "FOUND". You must be
    centered on the object.
</Objective>

<Coordinates>
   There is a grid overlaid on each image you are presented with. It is meant to (
    roughly) communicate which point will be in drone's center of vision if you
    move in that direction. Note that height of the drone is not represented in the
     grid.
</Coordinates>

<Controls>
   <Action space>
      To move the drone in a certain direction, use the following format: <Action
    >(x, y, z)</Action>. For example, if you want to fly to the place denoted as
    (10, 10) on the grid without changing the altitude, you should reply with <
    Action>(10, 10, 0)</Action>.

      x and y are the coordinates on the grid, and z is the altitude difference.
    For example, <Action>(0, 0, -10)</Action> means that you are moving 10 meters
    down. This is especially important, since you need to get close to the object
    in question.

   </Action space>
```


https://doi.org/10.52202/085713-5594

```
    <Formatting>

       Your each response should contain XML <Reasoning> tag and <Action> tag.
       <Reasoning> tag should contain your reasoning for the move you are making.
       <Action> tag should contain the move you are making.

       If you find the object, fly below 10 meters relative to it and reply with "
    FOUND". Remember, it must be in your field of view when you reply with "FOUND"
    and you must be 10 meters above it or closer. Being too far away is not
    acceptable.

       For example:

       <Reasoning>This yellow point might be the object in question. I need to go
    lower to check for that. If it's not the object in question, I will continue
    the search. I will also slightly go to the north.</Reasoning>
       <Action>(5, 0, -30)</Action>

    </Formatting>

    <Limitations>
       You shouldn't move into coordinates that are outside of your view. Otherwise,
     you may hit something which is not ideal.
       You can make at most {glimpses - 1} moves. Your altitude cannot exceed 300
    meters.

       The search area is limited to what would be visible from the starting
    position if there were no buildings or obstacles. The object is within this
    area. You may not fly outside of it.
    </Limitations>
</Controls>
```

Furthermore, along with the textual description of the searched object we pass the image, showcasing how it should look like from above, with a following annotation:

Listing 5: Annotation of searched object's image in FS-2.

```
The object you're looking for is similar to this. This is NOT the drone's current
    view.
```

H Additional Experiments

In this section we present additional experimental results, evaluating design choices of our benchmark.

Table S3: Performance of Gemini 2.0 Flash and Pixtral-Large on ablations.

	Setting	City		Forest	
		Gemini	**Pixtral**	**Gemini**	**Pixtral**
FS-1	**Baseline**	**41.5%**	**21.5%**	**42.5%**	**38.0%**
	Compass actions	17.5%	21.0%	17.5%	22.0%
	No grid overlay	17.0%	15.5%	31.5%	20.0%
FS-Anomaly	**Baseline**	25.0%	4.0%	46.0%	26.0%
	Anomaly with ID	**34.0%**	**7.0%**	**59.0%**	**34.0%**

Impact of the action space. We test how changing the action format affects performance by replacing the default Cartesian (x, y, z) movements with compass-style commands that specify a direction (north, south, east, west, up, down) and distance. This type of action space is common in previous works. As shown in Table S3, this change significantly degrades performance for both

Table S4: **FS-Anomaly-1 results:** Success rates (± standard errors) of the evaluated models.

Model	FS-Anomaly-1		
	Overall (%)	Forest (%)	City (%)
GPT-4o	27.0 ± 3.1	39.0 ± 4.9	15.0 ± 3.6
Claude 3.5 Sonnet	27.5 ± 3.2	37.0 ± 4.9	18.0 ± 3.9
Gemini 2.0 flash	$\mathbf{35.5 \pm 3.4}$	$\mathbf{46.0 \pm 5.0}$	$\mathbf{25.0 \pm 4.4}$
Phi 3.5 vision	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
InternVL-2.5 8B MPO	3.5 ± 1.3	6.0 ± 2.4	1.0 ± 1.0
Llava-Interleave-7b	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Qwen2.5-VL 7B	2.8 ± 1.2	3.7 ± 2.1	2.0 ± 1.4
Qwen2-VL-72B	7.5 ± 1.9	10.0 ± 3.0	5.0 ± 2.2
Llava-Onevision 72b	8.5 ± 2.0	11.0 ± 3.1	6.0 ± 2.4
Pixtral-Large	15.0 ± 2.5	26.0 ± 4.4	4.0 ± 2.0

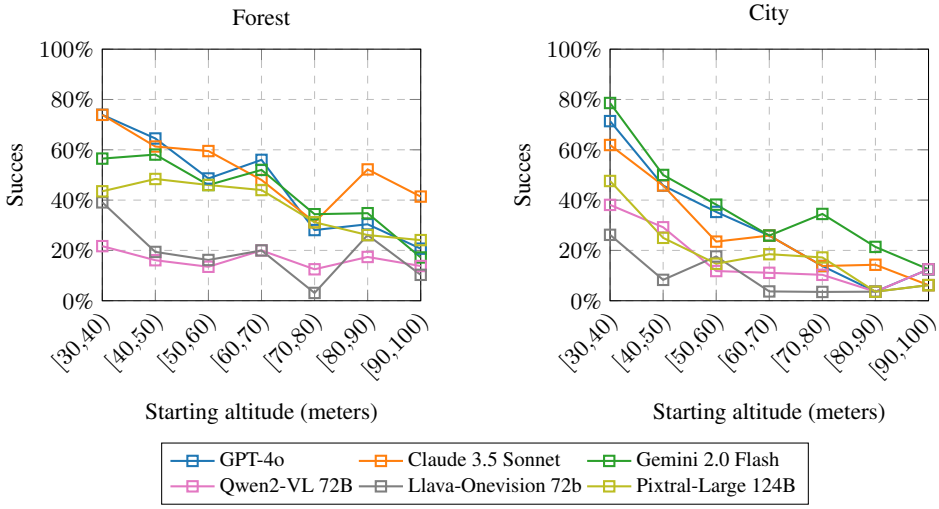


Figure S5: **Success rate per starting altitude interval:** In this figure, we plot the success rates of the models as a function of the initial altitude interval for both the Forest and the City environments in the FS-1 challenge. The performance drop that comes with increasing altitude is more pronounced for the City environment.

Gemini 2.0 Flash and Pixtral-Large. Gemini's accuracy drops from 42.5% to 17.5% in Forest and from 41.5% to 17.5% in City. Pixtral falls from 38% to 22% in Forest, with little change in City (21.5% to 21%). These results highlight the importance of a flexible, continuous action space. Compass-style movement restricts fine-grained control and appears to limit the models' ability to effectively search.

Impact of the grid overlay. To assess the importance of the grid overlay on each glimpse, we run an ablation where glimpses are shown without the grid. As shown in Table S3, removing the grid leads to a notable drop in performance for both Gemini 2.0 Flash and Pixtral-Large. Gemini's success rate falls from 42.5% to 31.5% in Forest, and more dramatically in City—from 41.5% to 17%. Pixtral shows a similar trend, dropping from 38% to 20% in Forest and from 21.5% to 15.5% in City. These results suggest that the grid plays a key role in helping models maintain spatial awareness, especially in more complex environments. Without it, both localization and goal-oriented planning appear to suffer. This result is consistent with previous works on vision-language coordination [29].

Impact of hidden vs explicit anomaly categories. In the FS-Anomaly setting, the model must identify an object that doesn't belong in the surrounding environment. In the base version, the

https://doi.org/10.52202/085713-5594

anomaly category is implicit, requiring the model to infer what is out of place. In this ablation, we explicitly tell the model what object to look for.

As shown in Table S3, providing the anomaly identity significantly improves performance. Gemini's accuracy increases from 46% to 59% in Forest and from 25% to 34% in City. Pixtral also improves, from 26% to 34% in Forest and from 4% to 7% in City.

This result suggests that models often struggle with anomaly detection due to semantic ambiguity rather than visual perception. Explicitly stating the anomaly class helps disambiguate the task and leads to more focused and successful exploration.

FS-Anomaly-1 full results. We provide full results of the FS-Anomaly-1 benchmark with distinction between Forest and City environment performance in Table S4.

Impact of the starting height. Finally, we verify how the results vary depending on the starting height. As seen in Figure S5, the success rate decreases as the starting altitude (and therefore distance to the target) increases. This is particularly significant in the City environment, where the success rate for the best models drops by half, likely due to the high number of distracting objects. In contrast, in the generally easier Forest environment, this decline is less pronounced and occurs mainly in the upper half of the altitude range.

I Sim2real gap

The main design goal of FlySearch is the evaluation of VLM exploration and 3D spatial reasoning capabilities, with UAV object navigation serving as a realistic assessment task. However, we acknowledge the fact that FlySearch can serve as a platform for testing methods designed specifically for UAV control, including non-VLM solutions. Therefore, in this section, we discuss in depth the differences between FlySearch and real-life UAV control, underlining issues that were simplified or omitted in order for the problem to be addressable with existing foundational VLMs.

Environment. FlySearch uses Unreal Engine 5 to provide near-photorealistic simulation of highly complex environments, with both procedurally generated and handcrafted content, using a vast set of high-resolution assets and generation rules. This setup allows for world simulation on par with the latest video games. However, the real world presents far greater complexity and unpredictability than any video game. This might lead to discrepancies between the behavior of VLM in simulated and real-world environments.

World dynamics. The real world contains a vast number of dynamic elements (moving cars, walking people, flying birds, etc.) that cannot be fully simulated even with modern video game engines. Moreover, FlySearch omits some of the flight-related dynamics, such as sudden gusts of wind or abrupt weather changes, that were deemed too complex for current foundational VLMs.

Sensor simulation. Our benchmark limits sensor inputs to a downward-facing camera and altitude reading with 1 m accuracy. The camera sensor is simulated using real-time ray-tracing capabilities of the engine, accounting for camera auto-exposure, depth of view, light direction (time of the day), fixed light cloud cover, and atmospheric haze. This enables an overall realistic simulation of a modern digital camera, but simplifies the weather element and omits any data loss or low-light condition emulation. Furthermore, real-life platforms may feature multiple cameras, LIDAR/RADAR, GPS units, IMU, and other sensor modalities that are not simulated in FlySearch.

Target objects. The searched object is stationary (may be animated but remains in the same location), unique for the scenario (visually similar but semantically different objects may exist on the map), and visible from the top. In real-world situations, the object can move freely in the environment, including moving inside buildings, and the description of the object provided to the agent can match multiple objects. It may require more complex search strategies and the cooperation of multiple agents.

UAV control. FlySearch uses high-level relative movement actions with 1 m resolution. This can be implemented in real-life UAVs using existing autopilot software such as PX4 Autopilot or DroneKit, which can accommodate wind changes and drift. However, a more direct control of the platform

could speed up the exploration process in practice and make it less dependent on GPS or inertial navigation systems. At the same time, this makes the task more complex for both VLM models and human evaluators.

Other simplifications. The benchmark limits agent interaction with the world to up to 20 steps due to the model context length limitations of most open-source VLMs. In real-life problems, the location of an object in a large environment may require significantly more observations. In addition, real-world UAVs are subject to hardware and software failures, none of which are simulated in FlySearch. In future work, when VLMs can solve our toughest challenge set, we plan to address some of the above limitations, making our benchmark more realistic.

J Example trajectories

The sample trajectories of different VLMs on FS-1 and FS-2 scenarios are provided in a separate file in the supplementary material, as well as at: `https://github.com/gmum/FlySearch/blob/main/docs/example-trajectories.pdf`

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope, as they consistently present the research objectives, methodology, and key findings, aligning with the detailed content and conclusions provided in the main body of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

https://doi.org/10.52202/085713-5594

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There are no theoretical results in the paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experiments are described in the main body of the paper. Moreover, there are full details in the appendix section. Full source code is provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The data and code are publicly available.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The full details are presented in the core paper, the appendix, or available in the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

https://doi.org/10.52202/085713-5594

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main empirical results are accompanied by standard errors.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information about the type of compute resources used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research does conform with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The societal impacts are presented in Appendix A due to space restrictions.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The benchmark poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

https://doi.org/10.52202/085713-5594

Answer: [Yes]

Justification: The creators and original owners of assets that are used in the paper are properly credited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: New assets are documented and the documentation is provided alongside them.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: The instructions given to human subjects are the same as those given to the models in the benchmark. Full details are presented in Appendix D.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: The paper includes benchmark scores achieved by human manual software testers (employees). As clarified in Appendix D, the study does not pose any risk to participants. The study was conducted according to institutional regulations on data gathering and processing and was subject to internal approval process.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The LLMs are only used for editing and as a test subject.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-5594