UAS Visual Navigation in Large and Unseen Environments via a Meta Agent

Yuci Han[a], Charles Toth[b], Alper Yilmaz[a]

[a] Photogrammetric Computer Vision Laboratory,
[b] Satellite Positioning and Inertial Navigation Laboratory
The Ohio State University, Columbus, OH 43210, USA
{han.1489, toth.2, yilmaz.15}@osu.edu

Keywords: Visual Navigation, Deep Reinforcement Learning, Unmanned Aerial System.

Abstract

The aim of this work is to develop an approach that enables Unmanned Aerial System (UAS) to efficiently learn to navigate in large-scale urban environments and transfer their acquired expertise to novel environments. To achieve this, we propose a meta-curriculum training scheme. First, meta-training allows the agent to learn a master policy to generalize across tasks. The resulting model is then fine-tuned on the downstream tasks. We organize the training curriculum in a hierarchical manner such that the agent is guided from coarse to fine towards the target task. In addition, we introduce Incremental Self-Adaptive Reinforcement learning (ISAR), an algorithm that combines the ideas of incremental learning and meta-reinforcement learning (MRL). In contrast to traditional reinforcement learning (RL), which focuses on acquiring a policy for a specific task, MRL aims to learn a policy with fast transfer ability to novel tasks. However, the MRL training process is time consuming, whereas our proposed ISAR algorithm achieves faster convergence than the conventional MRL algorithm. We evaluate the proposed methodologies in simulated environments and demonstrate that using this training philosophy in conjunction with the ISAR algorithm significantly improves the convergence speed for navigation in large-scale cities and the adaptation proficiency in novel environments. The project page is publicly available at `https://superhan2611.github.io/isar_nav/`.

1. Introduction

Long-range navigation is a complex cognitive task that relies on robust localization techniques using GPS, as well as a variety of subtasks, such as detection and mapping. This work investigates autonomous navigation using only visual observations from a single monocular camera, mimicking the way that humans would navigate in an unfamiliar environment without GPS [Lee et al., 2015] [Wang et al., 2018a]. Visual navigation has potential for practical applications in scenarios where other sensors are inaccessible.

With the development of deep reinforcement learning (DRL), various end-to-end DRL approaches have been developed for visual autonomous navigation tasks in small-scale indoor environments [Zhu et al., 2017] [Mousavian et al., 2019] [Wang et al., 2018b] [Wang et al., 2019] [Han and Yilmaz, 2021] [Han et al., 2022] [Hong et al., 2020]. However, long-range visual navigation in urban environments is significantly more challenging than indoor environments and has not yet been fully explored. In particular, urban environments are characterized by their complexity, diverse scenes, and numerous obstacles. It is difficult to apply traditional DRL training techniques designed for indoor-scale tasks to urban navigation due to data-inefficient and high training costs. In addition, low-altitude UAS navigation allows free flight in open spaces, introducing higher complexity and flexibility that further complicates the challenge.

In this paper, our aim is to develop an approach for learning to navigate on a large scale in less time and to adapt quickly to new environments. To this end, we introduce a meta-curriculum training scheme. Inspired by meta-learning theory [Finn et al., 2017] [Nichol and Schulman, 2018] [Wortsman et al., 2018], the proposed training strategy engages in ‘learn how to learn’ and ‘learn to navigate’. More concretely, we decompose the training into two phases: meta-training and curriculum fine-tuning (see Fig. 1). Meta-training allows the agent to develop essential navigation skills by exploring a variety of small tasks. This prior knowledge can then be transferred to new tasks, enabling fast adaptation. In the next phase, we fine-tune the meta-agent on the target task using a curriculum training scheme. This strategy increases the complexity of hierarchical training tasks by providing the agent with progressively more difficult environments that help the agent learn to reach more distant destinations. In addition, we propose an incremental self-adaptive reinforcement learning algorithm (ISAR) to accelerate the learning speed of the traditional MRL approach for long-episode tasks with sparse rewards. ISAR learns an interaction loss and an adaptation loss incrementally within an episode, allowing the agent to optimize its policy through self-adaptive exploration, leading to improved learning and exploration efficiency.

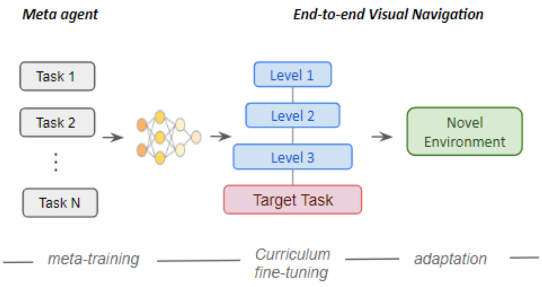

Figure 1. **Overview of learning framework.** The navigation task consists of two phases: meta-training and curriculum fine-tuning. Meta-training allows the agent to learn a master navigation policy. The hierarchical-structured curriculum adapts the meta-policy to the target task. This meta-policy can further be transferred to novel environments.

https://doi.org/10.5194/isprs-annals-X-2-2024-105-2024

The contributions of this paper are summarized as follows: **(a)** We present a two-stage framework for training an end-to-end low-altitude UAS visual navigation system in urban environments. Curriculum training improves the efficiency of learning the high-complexity long-range navigation task, and meta-training enables the generalization ability to new environments. **(b)** We introduce ISAR, an improved MRL algorithm for the long-episode reinforcement learning task. The key components of ISAR are the inner-episode adaptation loss design and the incremental update procedure, which allow the agent to learn to adapt from their own experiences. **(c)** We present extensive experimental validation in the AirSim simulation environment [Shah et al., 2018]. The results verify the superiority of our training strategy over other traditional reinforcement learning approaches. Additionally, we demonstrate that the ISAR algorithm outperforms the MRL baseline in terms of convergence speed. Furthermore, the meta-training policy shows a notable transfer ability to novel environments.

2. Related Work

Deep Reinforcement Learning Navigation. Reinforcement learning (RL) algorithms have been applied to a wide range of domains, including autonomous driving [Han and Yilmaz, 2022], robotics [Luo et al., 2020] [Kahn et al., 2017], and video games [Mirowski et al., 2016]. Recently, RL has increased in popularity for navigation tasks due to its flexibility in learning complex decision-making tasks. However, most of the published literature is focused on indoor simulated environments [Wu et al., 2020] [Shen et al., 2019]. Li et al. [Li et al., 2019a] used graph attention memory (GAM) to address the long-term memory limitation of reactive deep reinforcement learning. Kwon et al. [Kwon et al., 2021] investigated a similar problem and introduced visual graph memory (VGM) that can learn a goal-oriented policy directly from the graph representation. Wu et al. [Wu et al., 2019] improved the generalizability of semantic visual navigation agents by using Bayesian Relational Memory (BRM). However, these models are fixed during the transfer process and do not generalize to new environments. On the other hand, Wortsman et al. [Wortsman et al., 2018] proposed a self-adaptive visual navigation model that can learn during the training and inference stages. Like this work, we develop a transferable model for the urban navigation task. Unlike this work, we propose a more effective training framework and an ISAR algorithm that achieve faster convergence in large-scale urban navigation and adapt quickly to novel environments.

Meta-Learning. Meta-learning, also known as ‘learning to learn,’ is a rapidly growing field in machine learning and artificial intelligence [Nichol et al., 2018] [Rusu et al., 2019] [Yin et al., 2020]. The core idea of meta-learning is to learn algorithms or models that can quickly adapt to new tasks or environments with a small amount of data. Gradient-based meta-learning [Hochreiter et al., 2001] [Frans et al., 2018] [Ravi and Larochelle, 2016] [Nichol et al., 2018] has shown promising results in various domains, especially for the few-shot learning problem [Li et al., 2017b] [Jamal and Qi, 2019]. The basic idea behind gradient-based meta-learning is to optimize the model’s parameters with the gradients of the loss function over a set of tasks so that it can quickly adapt to new tasks. Andrychowicz et al. [Andrychowicz et al., 2016] proposed a meta-learning approach that learns to learn by optimizing the learning algorithm itself and achieves state-of-the-art results on several meta-learning benchmarks. Lee et al. [Lee et al., 2019] proposed MetaOptNet, which learns feature embedding that generalizes well under a linear classification rule for novel categories. Finn et al. [Finn et al., 2017] proposed a meta-learning approach called Model-Agnostic Meta-Learning (MAML) that can be applied to any differentiable model or network architecture. Li et al. [Li et al., 2017a] and Yu et al. [Yu et al., 2018] augmented the MAML algorithm by synthesizing virtual testing domains and simulating domain shift during training so that it transfers supervision in one domain to another. Our proposed training scheme relies on the MAML model during the meta-training stage. Furthermore, our ISAR algorithm integrates meta-learning and incremental learning into a per-episode exploration process of the conventional MRL algorithm, which improves the learning and exploration efficiency.

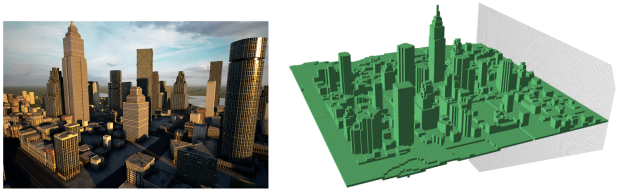

Figure 2. The illustration of AirSim simulated urban environment.

3. Methodology

3.1 Urban Visual Navigation Task

The objective of the visual navigation task of the UAS is to train an autonomous agent to navigate from an arbitrary starting point to the designated destination. For this project, we use AirSim, a simulated environment that consists of diverse realistic scenes and numerous objects, as the deployment platform [Brushify - Build A City In UE4, n.d.] (see Fig. 2).

The objectives of this work are to: (1) train the UAS agent to perform low-altitude ($15m$) urban navigation; and (2) assess the transfer performance of the meta-policy in a novel environment. We formalize the navigation task as a Markov Decision Process $(\mathscr{S}, \mathscr{A}, p, r)$ where $\mathscr{S}$ is the state space, which consists of only image inputs captured by a downward looking camera; $\mathscr{A}$ is the action space, which includes *moving forward*, *moving right*, *moving left*, *moving backward*; p is the transition function between states and r is the reward. In the navigation task, the agent starts the episode from a randomly selected initial position and explores the environment to reach the destination in as few steps as possible. During exploration, the agent selects a series of actions from $\mathscr{A}$, and receives corresponding rewards from r until the episode ends. We assign a reward of 5 points for successful reaching the destination, otherwise, -0.01 points is applied as time penalty.

3.2 Meta-Curriculum Training Framework

As illustrated in Fig. 1, the learning process comprises the meta-training stage and the curriculum fine-tuning stage. The meta-training prepares a meta-agent for downstream tasks, and the curriculum fine-tuning adapts the meta-policy to the target task.

Meta-Training. The key idea of meta-training is to enable the agent to build up its understanding of the navigation rules by practicing on smaller tasks that can be mastered quickly and easily. More precisely, we use g_ψ to represent the meta-policy,

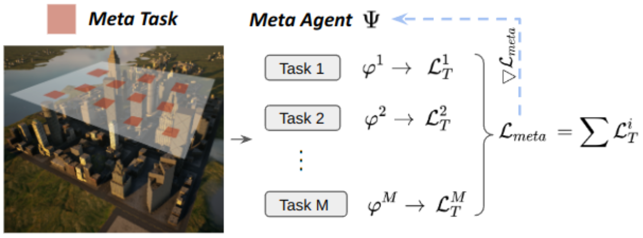
Meta Task
Meta Agent Ψ
$\nabla\mathcal{L}_{meta}$
Task 1
Task 2
Task M
$\varphi^1 \rightarrow \mathcal{L}_T^1$
$\varphi^2 \rightarrow \mathcal{L}_T^2$
$\varphi^M \rightarrow \mathcal{L}_T^M$
$\mathcal{L}_{meta} = \sum \mathcal{L}_T^i$

Figure 3. The illustration of meta-training process. $\mathscr{L}_T^i$ is the loss for the i^{th} meta-task. The meta-loss $\mathscr{L}_{meta}$ is the sum of $\mathscr{L}_T^i$, which is used to update the meta-agent ψ.

where ψ is the weight of a deep learning model. We generate M meta-tasks as the training task set $\mathscr{T}_{meta}$. Each task is defined as $\mathscr{T}^i = \{\mathscr{L}_T^i(\mathscr{S}, \mathscr{A}, \mathscr{R}), \mathscr{E}_T^i\}$ that consists of task loss $\mathscr{L}_T^i$ and task environment $\mathscr{E}_T^i$. In the context of urban navigation task, we train the meta-agent at an altitude of $300m$ (see Fig. 3). This is because the drone can capture a wider and more comprehensive view of the environment at this level, which benefits its adaptation to lower-altitude navigation. Consequently, as illustrated in Figure 3, $\mathscr{E}_T^i$ is a subregion generated at the altitude of $300m$. Each $\mathscr{T}^i$ is an indivisual navigation task with a designated target. We draw meta-tasks in batches from $\mathscr{T}_{meta}$ in sequential order for training. To begin with, we employ the RL algorithm to learn the navigation policy $\pi(\varphi^i)$ with parameters φ^i for the meta-task $\mathscr{T}^i$ and then calculate $\mathscr{L}_T^i$. The meta-policy ψ is trained by optimizing performance across all meta-tasks. Therefore, the meta-loss $\mathscr{L}_{meta}$ is defined as the sum of all losses from meta-tasks: $\sum \mathscr{L}_T^i$. We utilize stochastic gradient descent (SGD) with a learning rate of η to update the meta-agent (see Fig. 3):

$$\psi' = \psi - \eta \nabla_\psi \sum_{\mathscr{T}_i \sim \mathscr{T}_{meta}} \mathscr{L}_T^i \quad (1)$$

Following this step, we update the parameters of $\pi(\varphi^i)$ with ψ' and repeat this process until convergence. According to the meta-learning theory [Finn et al., 2017], this approach encourages the discovery of parameters ψ sensitive to changes such that slight modifications to the parameters lead to significant improvements in new tasks. Therefore, the meta-agent can be fine-tuned through a gradient-based approach, enabling it to make rapid adaptation on novel tasks.

Hierarchical Curriculum Training. In this phase, we fine-tune the meta-policy to the target task. Compared to indoor navigation, urban navigation is significantly more challenging due to its larger scale of environment and more complex scenarios. Additionally, navigating at lower altitudes is even harder than at higher altitudes, as there are numerous obstacles and narrower perspectives (with a downward camera) (see Fig. 4).

Our proposed curriculum training scheme aims to accelerate the learning speed by guiding the agent to learn the policy from 'coarse to fine.' As illustrated in Fig. 4, the number of obstacles decreases with higher altitudes. Consequently, our training curriculum is structured hierarchically starting from higher altitudes and gradually moving towards lower altitudes as task difficulty increases. In particular, we start by initializing the agent with a meta-policy and fine-tune it at an altitude of 75 meters with sparse obstacles. As the agent gradually refines its policy, we subsequently reduce the flight height to 45 meters. Ultimately, we set the drone's height at 15 meters to complete the target task.

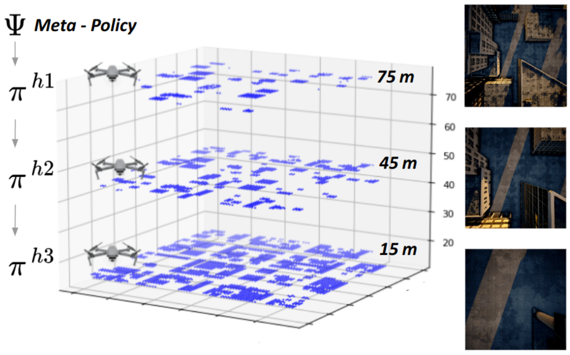

Figure 4. The illustration of the obstacles distribution and the observation of the same location at altitudes of $75m$, $45m$ and $15m$. The agent starts with the meta-policy and learns hierarchical policies from coarse-to-fine.

3.3 Incremental Self-Adaptive Reinforcement Learning

Previous works [Wortsman et al., 2018] [Luo et al., 2020] [Li et al., 2019b] [Mayo et al., 2021] use meta reinforcement learning for transfer navigation in unseen environments. Generally, MRL can learn a policy with rapid adaptation ability by introducing a meta-loss at the end of the episodes or tasks. Our proposed ISAR algorithm is essentially an optimization of conventional MRL for long-episode task such as large-scale navigation. Like MRL, ISAR can leverage any existing RL algorithms, such as actor-critic (A3C) [Mnih et al., 2016], double Q-learning [Hasselt, 2010], or PPO [Schulman et al., 2017], as the base model. Before formally presenting the ISAR algorithm, we begin with an overview of the traditional reinforcement learning process applied to the navigation task.

We use s_t to denote the agent's state at time t, which is an RGB image observed by the drone. The navigation policy $\pi(a_t|s_t, s_{target})$ is a distribution over actions given s_t and the target observation s_{target}. In the DRL model, this policy is represented by a network with parameters θ, denoted as $\pi_\theta(a_t|s_t, s_{target})$. We use ResNet18 to extract the features of s_t and s_{target}. The combined feature is then fed into the policy network to generate the navigation policy and value. During an exploration episode, the agent starts from s_0, takes a sequence of actions according to the policy $\pi_\theta(a_t|s_t, s_{target})$ and receives feedback in the form of rewards r which guides the agent towards the target. Our goal is to maximize the sum of discounted rewards by minimizing the loss $\mathscr{L}_{episode}$ throughout the episode.

The proposed ISAR algorithm accelerates this learning process by incremental self-adaptive exploration (Figure 5). The policy update in traditional MRL is typically performed on episode-basis or task-basis using episodes-loss or tasks-loss as the meta-loss $\mathscr{L}_{meta}$. In our model, we enable the agent to learn the navigation policy incrementally by allowing it to update based on a small number of interactions at a time within the episode rather than doing a one-time update until the end of the episode or the task.

More concretely, we learn two kinds of policies in the ISAR algorithm, the interaction policy and the adaptive policy. We define the interaction policy using a function f_θ with parameters θ. We start by initializing f_θ with θ_0 and conduct a N-step interaction using f_{θ_0}, which generates a trajectory segment $(s_0, a_0, r_0, ..., s_N)$, denoted as $\mathscr{S}_0$. Similarly, the interaction

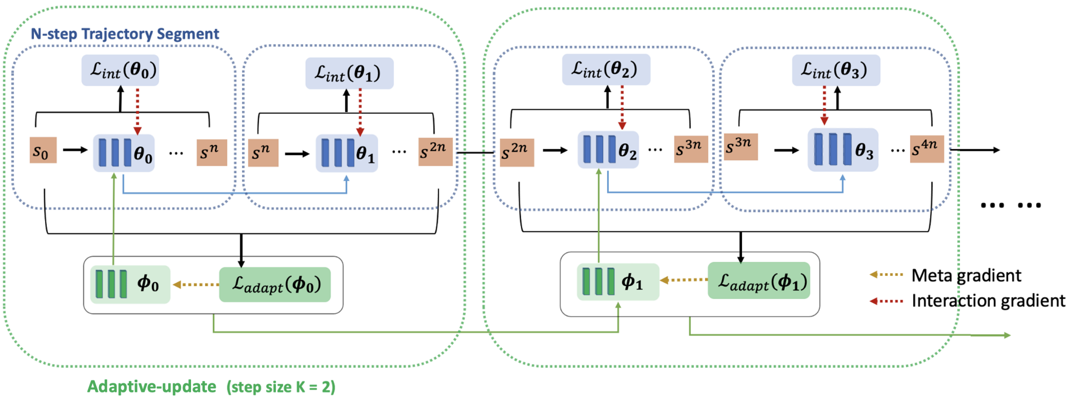
N-step Trajectory Segment
$\mathcal{L}_{int}(\theta_0)$
$\mathcal{L}_{int}(\theta_1)$
$\mathcal{L}_{int}(\theta_2)$
$\mathcal{L}_{int}(\theta_3)$
s_0
θ_0
s^n
s^n
θ_1
s^{2n}
s^{2n}
θ_2
s^{3n}
s^{3n}
θ_3
s^{4n}
… …
ϕ_0
$\mathcal{L}_{adapt}(\phi_0)$
ϕ_1
$\mathcal{L}_{adapt}(\phi_1)$
Meta gradient
Interaction gradient
Adaptive-update (step size K = 2)

Figure 5. The illustration of the ISAR framework with adaptive-update step size = 2. During exploration, we learn two types of losses: the interaction loss $\mathscr{L}_{int}$ to update the interaction policy f_θ, and the adaptive loss $\mathscr{L}_{adapt}$ based on trajectory segment with length N to update the adaptation policy f_ϕ.

policy for trajectory segment $\mathscr{S}_i$ is denoted as f_{θ_i}. We calculate the interaction loss $\mathscr{L}_{int}(f_{\theta_i})$ for $\mathscr{S}_i$ and update the policy f_{θ_i} using one gradient update with learning rate α:

$$\theta_{i+1} = \theta_i - \alpha \nabla_{\theta_i} \mathscr{L}_{int}(f_{\theta_i}) \tag{2}$$

$\mathscr{L}_{int}$ is similar to the regular $\mathscr{L}_{episode}$ that incentivizes the agent to explore the policy to reach the target. Additionally, we consider an adaptive policy in our approach. This policy learns the rules for self-adaptive exploration for the remainder of the episode. For clarity, we define the adaptive policy using the same function f_ϕ with different parameters ϕ. Therefore, f_θ and f_ϕ share the same network structure and can exchange weights. Note that, at the beginning of the episode, we initialize θ_0 with ϕ_0. We assume an adaptive-update process with a step size of K, which means we collect K trajectory segments $\mathscr{S}_{i \sim i+K-1}$ for adaptation policy update. The adaptation loss $\mathscr{L}_{adapt}$ is defined as the sum of the interaction losses $\mathscr{L}_{int}$ for trajectory segments $\mathscr{S}_{i \sim i+K-1}$. Therefore, the objective function of the adaptive policy f_{ϕ_i} is defined as follows:

$$\underset{\theta}{min} \sum_{i}^{i+K-1} \mathscr{L}_{int}(f_{\theta_i}) = \sum_{i}^{i+K-1} \mathscr{L}_{int}(f_{\theta_{i-1} - \alpha \nabla_\theta \mathscr{L}_{int}(f_{\theta_{i-1}})}) \tag{3}$$

We update the adaptive policy f_{ϕ_i} using one gradient update with learning rate β:

$$\phi_{i+1} = \phi_i - \beta \nabla_\phi \sum_{i}^{i+K-1} \mathscr{L}_{int}(f_{\theta_i}) \tag{4}$$

According to Eq. 4, we note that the adaptive policy is trained by optimizing the performance of the interaction policy f_{θ_i} with respect to θ_i within the set trajectory segments $\mathscr{S}_{i \sim i+K}$. The intuition behind this approach is that by interacting a short trajectory with a small number of gradient steps, we estimate the agent's overall performance periodically within the episode and refine the navigation policy using adaptive loss and the interaction loss. As such, the agent can learn not only the sequence of actions for a specific trajectory but also become broadly applicable for the future exploration steps. This self-adaptive process enables the agent to dynamically adjust its exploration strategy which accelerates the learning process. We then update the parameters of $f_{\theta_{i+K}}$ with ϕ_{i+1} and repeat the incremental update process until the end of the episode. The complete algorithm for ISAR is outlined in Algorithm 1.

Algorithm 1 ISAR

```
Require: $\alpha, \beta$: learning rate
Require: $N$: trajectory segment length
Require: $K$: meta-update step size
randomly initialize $\theta, \phi$
$j \leftarrow 0$
while not converged do
    $i \leftarrow 0$
    $t \leftarrow 0$
    $\theta_i \leftarrow \phi_j$
    while not done do
        $\theta_i \leftarrow \phi_j$
        Take action $a$ sampled from $f_{\theta_i}(s_t)$
        $j \leftarrow j + 1$
        if $t$ is divisible by N then
            $\theta_{i+1} = \theta_i - \alpha \nabla_{\theta_i} \mathscr{L}_{int}(f_{\theta_i})$
            $i \leftarrow i + 1$
        if $t$ is divisible by $K \times N$ then
            $\phi_{j+1} = \phi_j - \beta \nabla_{\phi_j} \sum_i^{i+K} \mathscr{L}_{int}(f_{\theta_i})$
            $\theta_i \leftarrow \phi_{j+1}$
            $j \leftarrow j + 1$
return $\theta, \phi$
```

4. Experiments

In this section, we evaluate the performance of the proposed methodologies for the urban navigation task in the AirSim environment. We seek answers to the following questions: (1) Can ISAR accelerate the learning process for the navigation task in comparison to traditional meta reinforcement learning? (2) Does the meta curriculum training scheme improve the learning efficiency for large-scale urban navigation? (3) Does the meta-training policy exhibit adaptation ability to new environments? For this experiment, we train the network with PyTorch using multi-process [Paszke et al., 2019] on NVIDIA GeForce RTX 3090.

4.1 Learning to Navigate using ISAR

This experiment aims to validate the efficacy of the ISAR algorithm in terms of convergence speed compared to MRL. The baseline used for comparison in this study is the **Self Aaptive Visual Navigation (SAVN)** [Wortsman et al., 2018] model. SAVN is an implementation of traditional MRL for the indoor navigation task. It optimizes the policy through the meta-loss

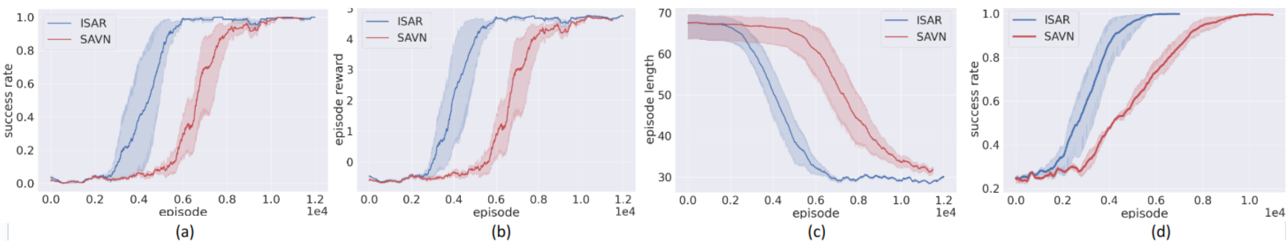

Figure 6. Figure (a), (b), (c) illustrate the learning process of the ISAR and SAVN models in the subarea of AirSim urban environment. Figure (d) shows the training process of the meta-policy.

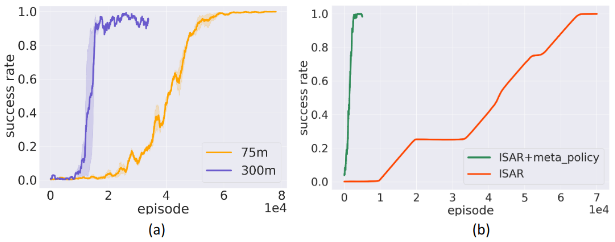

Figure 7. (a) Learning process at altitudes of $300m$ and $75m$. (b) The comparison between the transfer process of meta-agent and the standard agent that learns from scratch.

without incremental policy updates. SAVN has been used as the base model in several recent works [Luo et al., 2020] [Li et al., 2019b] [Mayo et al., 2021].

In the experiment setup, we choose a sub-region in the AirSim urban environment as the task environment. To ensure a fair comparison, both models are trained to learn to navigate from scratch to the same destination. We use the same A3C [Mnih et al., 2016] architecture as the foundational model for both ISAR and SAVN algorithms. The training process continues until convergence, indicating that the agent is capable of reaching the destination from any randomly selected starting position in the environment. We set the trajectory segment length (N in Algorithm 1) to 5 and the adaptive-update step size (K in Algorithm 1) to 3. The maximum episode length is set to 70, meaning that if the target is not reached within 70 steps, the episode will be forced to terminate. We use SGD for interaction policy updates, and Adam [Kingma and Ba, 2014] for adaptation policy updates.

We replicate the experiment with random seeds and evaluate the learning performance of both models in terms of episode length, average success rate and episode reward. Figure. 6 (a), (b), (c) compare the learning process of different models. The results show that our ISAR algorithm demonstrates a significant improvement in convergence speed compared to SAVN. The ISAR model reaches convergence in approximately $5K$ learning episodes, which is 50% less than the $10K$ episodes required by SAVN. Both models achieve similar average episode lengths at convergence.

4.2 Learning to Navigate with Meta-Curriculum Training

In this section, we present the meta-curriculum training scheme for the low-altitude urban navigation task.

Meta-Training. In this stage, our objective is to train a meta-agent to 'learn to learn to navigate', which can then be quickly transferred to the target environment. For meta-training, we set the agent's altitude at 300 meters and randomly select subareas within this level as meta-tasks. Each subarea spans an area of $150m \times 150m$. In this study, we train the meta-agent with 15 meta-tasks and implement both ISAR and SAVN for meta-training. As illustrated in Figure. 6 (d), Our ISAR model converges in approximately 40% fewer episodes ($6K$) compared to the SAVN meta-reinforcement learning model ($10K$). This result further validates the efficacy of the incremental update procedure and adaptive policy in the ISAR algorithm.

Hierarchical Curriculum Training. In this phase, we aim to adapt the meta-agent to an lower altitude of 15 meters. Before presenting the details, we conduct an exploration of the learning expense at different levels by training the agent to navigate from scratch at altitudes of $300m$ and $75m$. As shown in Figure. 7 (a), the difficulty of the navigation task increases significantly as the altitude decreases. It takes around 5 times more learning iterations to learn the policy at the $75m$ level than at the $300m$ level due to the increasing complexity of the environment. Based on this observation, we implement curriculum training by gradually introducing increasingly complex tasks to the agent in a hierarchical structure, starting from high altitude and progressing to lower altitude.

Specifically, we initialize the agent with the meta-policy and conduct training at the altitude of 75 meters to navigate the entire environment. The length of an exploration episode is set to 120 steps. We then lower the drone to 45 meters and continue training. During this process, we allows the agent to query the higher level policy at same location. Finally, we set the agent at 15 meters level and learn to navigate to the same destination. Figure. 8 shows our two-stage training process. For comparison, we train another standard agent which is directly placed at the target level and learns from scratch without the meta-policy. The results demonstrate the effectiveness of the meta-curriculum training scheme in accelerating the training speed. As illustrated in Figure. 8, the total number of episodes required for convergence of meta-training and curriculum fine-tuning is approximately $35K$, whereas the standard agent require at least $60K$ episodes on average to achieve the same performance. An example of a navigation path is illustrated in Figure 10 (a).

4.3 Learning to Navigate in Unseen Environment

Our objectives for this experiment are to: (1) deploy the proposed approach in the real world settings; and (2) test the transfer ability of the meta-training policy to novel environments. We build an environment using Google Maps aerial photography that consists two different scenes (see Fig. 9). Scene A covers a residential area, whereas scene B spans the university campus with distinctive landscape. We don't include obstacles in this real world environment due to limited access to the real 3D data.

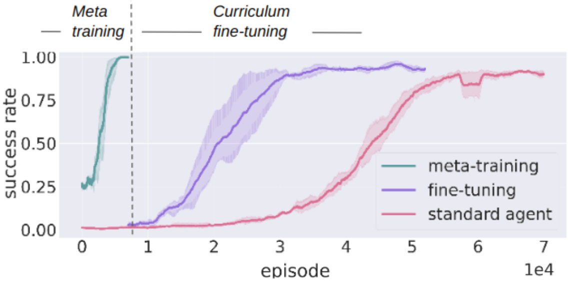

Figure 8. The illustration of the meta-training phase, curriculum fine-tuning phase, and the training process of the standard agent that learns directly at target level without meta-policy.

We first conduct meta-training in scene A with 25 meta-tasks. Then, we transfer the meta-policy for navigation in scene B through fine-tuning. For comparison, we train another agent from scratch without the meta-policy in scene B with the same destination. Figure. 7 (b) shows the transfer process of the meta-agent and the normal agent. The result demonstrates the efficacy of our methods in real scenarios. Moreover, the meta-training model exhibits significantly rapid transfer ability to unseen environments, converging within around $2K$ episodes and achieving a convergence speed 25 times faster than learning from scratch ($50K$ episodes). However, it has limitations in certain situations. We observe that if the new environment is significantly different from the meta-training environment, such as from urban to wild (see Fig. 10 (b)), the meta-training exhibits poor transfer performance.

4.4 Ablation Discussion

In this section, we perform an ablation discussion to gain further insights into our proposed approaches.

Impact of trajectory segment length N. The trajectory segment length N refers to the number of interactions the agent takes before updating its interaction policy each time. We observe that too short trajectories provide insufficient information to update the interaction policy, while excessively long trajectories reduce efficiency. Notably, the optimal value of N is similar to the average path length.

Impact of meta-update step size K. Our ISAR algorithm updates the adaptation policy with K trajectory segments each time. We observe that updating the adaptive policy too frequently with insufficient information or too infrequently can reduce efficiency. The optimal value of K strikes the best balance between exploration and learning update.

Impact of size and number of the meta-task. When larger and more meta-tasks are used during meta-training, the meta-agent gains a broader view and more comprehensive understanding of the environment, which results in better generalization ability. However, training with more meta-tasks would take longer. Therefore, the balance between training and transfer efficiency should be considered.

5. Conclusion

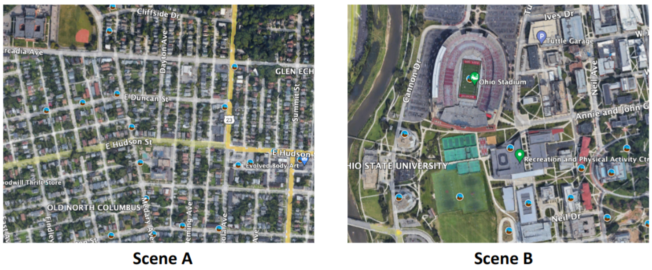

Figure 9. The demonstration of real world environment. We conduct meta-training in scene A and transfer the meta-policy to scene B.

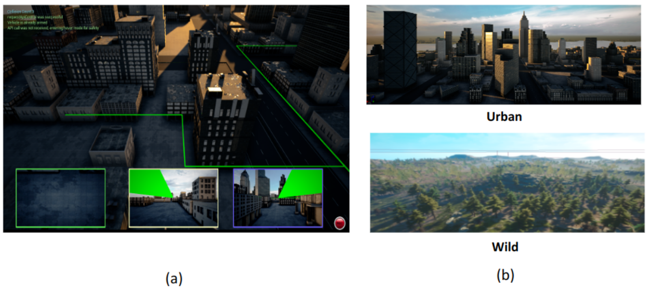

Figure 10. (a) The demonstration of a navigation path. (b) The urban and wild scenes in the simulation environment.

In this paper, we present a meta-curriculum training strategy to facilitate the learning process of large-scale low-altitude urban navigation task. By presenting progressively more complex environments to the agent, curriculum training allows the agent to quickly refine its policy to the target task. Additionally, we propose an incremental self-adaptive reinforcement learning (ISAR) algorithm to accelerate the learning speed of traditional meta-reinforcement learning. ISAR leverages the incremental update procedure, interaction loss, and adaptation loss to facilitate self-adaptive exploration within the episode. Our experiments show that the ISAR model outperforms traditional MRL approaches in terms of convergence speed for long-range navigation tasks. Moreover, we not only demonstrate the efficacy of our training scheme for large-scale environment navigation tasks, but also show that the meta-training policy has the ability to rapidly adapt to novel environments.

6. Acknowledgement

This work was supported in part by the U.S. Army Research Office under Grant AWD-110906.

References

Andrychowicz, M., Denil, M., Colmenarejo, S. G., Hoffman, M. W., Pfau, D., Schaul, T., de Freitas, N., 2016. Learning to learn by gradient descent by gradient descent. *NIPS*.

Brushify - Build A City In UE4, n.d. `https://www.youtube.com/watch?v=_UgeDCNadHQ`.

Finn, C., Abbeel, P., Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning*.

Frans, K., Ho, J., Chen, X., Abbeel, P., Schulman, J., 2018. META LEARNING SHARED HIERARCHIES. *International Conference on Learning Representations*.

Han, Y., Wei, J., Yilmaz, A., 2022. Uas navigation in the real world using visual observation. *2022 IEEE Sensors*, 1–4.

Han, Y., Yilmaz, A., 2021. Dynamic Routing for Navigation in Changing Unknown Maps Using Deep Reinforcement Learning. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 51, 145-150.

Han, Y., Yilmaz, A., 2022. Learning to drive using sparse imitation reinforcement learning. *2022 26th International Conference on Pattern Recognition (ICPR)*, 3736–3742.

Hasselt, H. V., 2010. Double q-learning. *NIPS*.

Hochreiter, S., Younger, A. S., Conwell, P. R., 2001. Learning to learn using gradient descent. *International Conference on Artificial Neural Networks*.

Hong, Y., Wu, Q., Qi, Y., Rodriguez-Opazo, C., Gould, S., 2020. VLNBERT: A Recurrent Vision-and-Language BERT for Navigation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1643-1653.

Jamal, M., Qi, G., 2019. Task agnostic meta-learning for few-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11711–11719.

Kahn, G., Villaflor, A., Ding, B., Abbeel, P., Levine, S., 2017. Self-Supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1-8.

Kingma, D. P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Kwon, O., Kim, N., Choi, Y., Yoo, H., Park, J., Oh, S., 2021. Visual graph memory with unsupervised representation for visual navigation. *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, IEEE, 15870–15879.

Lee, K., Maji, S., Ravichandran, A., Soatto, S., 2019. Meta-learning with differentiable convex optimization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lee, S. W., O’Doherty, J. P., Shimojo, S., 2015. Neural computations mediating one-shot learning in the human brain. *PLoS biology*, 13(4), e1002137.

Li, D., Yang, Y., Song, Y.-Z., Hospedales, T. M., 2017a. Learning to generalize: Meta-learning for domain generalization. *AAAI Conference on Artificial Intelligence*.

Li, D., Zhao, D., Zhang, Q., Zhuang, Y., Wang, B., 2019a. Graph Attention Memory for Visual Navigation. *2022 4th International Conference on Data-driven Optimization of Complex Systems (DOCS)*, 1-7.

Li, J., Wang, X. E., Tang, S., Shi, H., Wu, F., Zhuang, Y., Wang, W. Y., 2019b. Unsupervised Reinforcement Learning of Transferable Meta-Skills for Embodied Navigation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12120-12129. https://api.semanticscholar.org/CorpusID:208139239.

Li, Z., Zhou, F., Chen, F., Li, H., 2017b. Meta-SGD: Learning to Learn Quickly for Few Shot Learning. *CoRR*, abs/1707.09835. http://arxiv.org/abs/1707.09835.

Luo, Q., Sorokin, M., Ha, S., 2020. A Few Shot Adaptation of Visual Navigation Skills to New Observations using Meta-Learning. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 13231-13237.

Mayo, B., Hazan, T., Tal, A., 2021. Visual Navigation with Spatial Attention. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16893-16902. https://api.semanticscholar.org/CorpusID:233307395.

Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A. J., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., Hadsell, R., 2016. Learning to Navigate in Complex Environments. *CoRR*, abs/1611.03673.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning. M. Balcan, K. Q. Weinberger (eds), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*.

Mousavian, A., Toshev, A., Fišer, M., Košecká, J., Wahid, A., Davidson, J., 2019. Visual representations for semantic target driven navigation. *2019 International Conference on Robotics and Automation (ICRA)*, 8846–8852.

Nichol, A., Achiam, J., Schulman, J., 2018. On First-Order Meta-Learning Algorithms. *CoRR*, abs/1803.02999.

Nichol, A., Schulman, J., 2018. Reptile: a Scalable Metalearning Algorithm. *arXiv: Learning*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. *Neural Information Processing Systems*.

Ravi, S., Larochelle, H., 2016. Optimization as a model for few-shot learning. *International Conference on Learning Representations*.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R., 2019. Meta-learning with latent embedding optimization. *International Conference on Learning Representations*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Shah, S., Dey, D., Lovett, C., Kapoor, A., 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. M. Hutter, R. Siegwart (eds), *Field and Service Robotics*, Springer International Publishing, Cham, 621–635.

Shen, B. W., Xu, D., Zhu, Y., Guibas, L. J., Li, F.-F., Savarese, S., 2019. Situational Fusion of Visual Representation for Visual Navigation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2881-2890.

Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., Botvinick, M. M., 2018a. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21, 860-868.

Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y., Wang, W., Zhang, L., 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6622–6631.

Wang, X., Xiong, W., Wang, H., Wang, W. Y., 2018b. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. *Computer Vision – ECCV 2018*, Springer International Publishing, Cham, 38–55.

Wortsman, M., Ehsani, K., Rastegari, M., Farhadi, A., Mottaghi, R., 2018. Learning to Learn How to Learn: Self-Adaptive Visual Navigation Using Meta-Learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6743-6752.

Wu, Q., Gong, X., Xu, K., Manocha, D., Dong, J., Wang, J., 2020. Towards Target-Driven Visual Navigation in Indoor Scenes via Generative Imitation Learning. *IEEE Robotics and Automation Letters*, 6, 175-182.

Wu, Y., Wu, Y., Tamar, A., Russell, S. J., Gkioxari, G., Tian, Y., 2019. Bayesian Relational Memory for Semantic Visual Navigation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2769-2779.

Yin, M., Tucker, G., Zhou, M., Levine, S., Finn, C., 2020. Meta-learning without memorization. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net.

Yu, T., Finn, C., Dasari, S., Xie, A., Zhang, T., Abbeel, P., Levine, S., 2018. One-shot imitation from observing humans via domain-adaptive meta-learning. H. Kress-Gazit, S. S. Srinivasa, T. Howard, N. Atanasov (eds), *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., Farhadi, A., 2017. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. *IEEE International Conference on Robotics and Automation*.