Generative Meta-Adversarial Network for Unseen Object Navigation

Sixian Zhang[1,2], Weijie Li[1,2], Xinhang Song[1,2], Yubing Bai[1,2], and Shuqiang Jiang[1,2]

[1] Key Lab of Intelligent Information Processing Laboratory of the Chinese Academy of Sciences (CAS), Institute of Computing Technology (ICT), Beijing
[2] University of Chinese Academy of Sciences (UCAS), Beijing
{sixian.zhang, weijie.li, xinhang.song, yubing.bai}@vipl.ict.ac.cn
sqjiang@ict.ac.cn

Abstract. Object navigation is a task to let the agent navigate to a target object. Prevailing works attempt to expand navigation ability in new environments and achieve reasonable performance on the seen object categories that have been observed in training environments. However, this setting is somewhat limited in real world scenario, where navigating to unseen object categories is generally unavoidable. In this paper, we focus on the problem of navigating to unseen objects in new environments only based on limited training knowledge. Same as the common ObjectNav tasks, our agent still gets the egocentric observation and target object category as the input and does not require any extra inputs. Our solution is to let the agent "imagine" the unseen object by synthesizing features of the target object. We propose a generative meta-adversarial network (GMAN), which is mainly composed of a feature generator and an environmental meta discriminator, aiming to generate features for unseen objects and new environments in two steps. The former generates the initial features of the unseen objects based on the semantic embedding of the object category. The latter enables the generator to further learn the background characteristics of the new environment, progressively adapting the generated features to approximate the real features of the target object. The adapted features serve as a more specific representation of the target to guide the agent. Moreover, to fast update the generator with a few observations, the entire adversarial framework is learned in the gradient-based meta-learning manner. The experimental results on AI2THOR and RoboTHOR simulators demonstrate the effectiveness of the proposed method in navigating to unseen object categories. The code is available at `https://github.com/sx-zhang/GMAN.git`.

Keywords: Object navigation, Unseen object, Adversarial learning

1 Introduction

Visual object navigation is a task that requires an agent to navigate to the target object depending on the visual environment information. Recent works are typically trained by reinforcement learning (RL) to predict actions in real-time, with

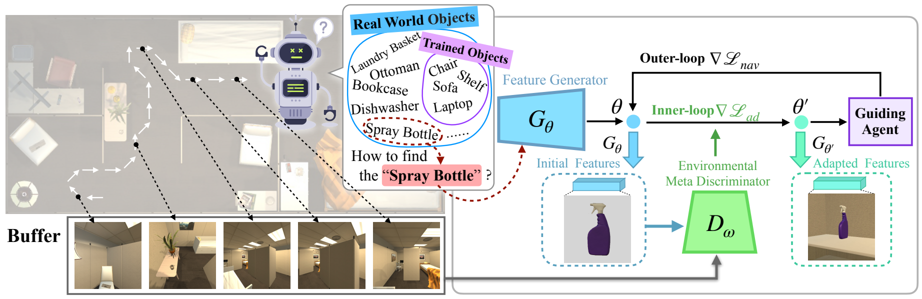


Fig. 1. Overview. We focus on navigating to the unseen objects in unseen environments. Given an unseen object category, our method is to firstly generate the initial features, and then adapt the generator to be compatible with the environment. The adapted features serve as a more specific goal to guide the agent.

the input of observed visual information and target object embedding. Existing object navigation methods achieve reasonable performance on seen objects (observed in training environments) [5,60,56,62]. However, these seen objects make up only a subset of all real world objects. As illustrated in Fig. 1, imagine a situation where the agent is trained to find several indoor furnitures (e.g. chair, sofa, shelf), when adapting to more practical applications, it's unavoidable that the user may require it to find the unseen objects (e.g. spray bottle). Since the actions of the agent are mainly driven by the correlation between visual representation and target semantic embedding, while the unseen object categories have not been trained to correlate with such visual information. Thus, the navigation ability of the agent to unseen objects is significantly limited.

Now that the agent is unfamiliar with the unseen target, a great challenge here is how to associate the unseen target with the limited "knowledge" learned from training. Some previous works rely on the helpful visual semantic information (e.g. object detection or instance segmentation) to establish semantic SLAM [7,5] or relation knowledge graph [62,10]. These methods may be unsuitable for unseen object navigation since unseen object detection or segmentation is not supported in those works. For other works focusing on visual embedding [35] or policy learning [56,33], although they can process the unseen objects (similar to seen objects) by using the same pipelines as seen objects, the performance is still limited, for that the visual representation and semantic embedding of unseen objects are neither modeled nor correlated during training. Therefore, an intuitive idea is to introduce some priors about the unseen targets, e.g. the object relationships [60] (involving seen and unseen objects) from the external dataset. Such way may be helpful in guiding agent to get close to unseen objects according to the relationships with other seen objects. However, only getting close may not be capable of locating the precise location of the target, since the visual characteristics of unseen objects are essentially unknown to the agent (i.e. the agent does not know what unseen objects look like).

In order to precisely locate the unseen objects, the agent may be required to "imagine" what the unseen objects look like. In some generative methods [13,25,58,59], the model learns the mapping from the semantic embedding to the visual features on seen object categories. Then given the semantic embedding of the unseen object, the model could "imagine" its visual features by analogy. Those works focus on generating the representations of the object itself, which are effective in static tasks (e.g. object recognition and fine-grained classification) that do not require many environment descriptions. However, in the navigation task, the visual observations typically contain objects and background. In this case, only generating objects features is not enough, and predicting precise navigation actions also requires considering the current environmental background.

In general, the key challenge of unseen object navigation is how to generate the unseen object representations the current working environment. In particular, the visual characteristics of both objects and environment (foreground and background) are critical to the navigation. Motivated by the challenges of generating comprehensive representations of unseen objects in new environments, we investigate our researches mainly from the following two aspects: 1) generating initial visual features of the unseen object from its semantic embedding; 2) proposing the generative adversarial learning model within the meta-learning structure to fast adapt the generator to the current working environment.

In this paper, we propose a generative meta-adversarial network (GMAN) for unseen object navigation, which consists of a feature generator (FG) and an environmental meta discriminator (EMD). The FG is pre-trained in advance to generate the unseen object features by learning the mapping from semantic embedding to the visual features on seen objects. Furthermore, to obtain the background information of current working environment, the EMD is proposed to adapt the FG to fit the environment, with an adversarial loss between the real-time observation features and generated object features. Significantly, a gradient-based meta-learning method is implemented to rapidly adapt the FG based on a few observations, which is shown as Fig. 1, where the adaptation of the FG the current environment can be regarded as the inner-loop, and maximizing the navigation reward serves as the outer-loop. The experimental results on AI2THOR [29] and RoboTHOR [8] simulators demonstrate the effectiveness of our GMAN on unseen object navigation.

2 Related Work

Visual object navigation. Goal-driven visual navigation can be categorized [2] into PointGoal [19,6,50,55], AreaGoal [30,57] and ObjectGoal. Several ObjectGoal works set an image as the goal [7,49,63], which contains more environment information, while our work sets the object category as the target (following most works) and our agent does not know about the unseen environments at all. Previous map-based methods typically construct a map in advance or in real-time [26,11,49,53]. Recently, learning-based methods are mainly composed of visual embedding and policy learning. The basic visual representation generally utilizes the ResNet [21]. [60] extracts a knowledge graph from external dataset and uses GCN [28] to embed the egocentric view and the knowledge graph. [35] proposes the attention probability module. Some works use more visual semantic information (e.g. object detection, instance segmentation) to establish semantic SLAM [5], spatial layout [39], prior knowledge graph [62,61] and scene memory[12]. [5] projects the segmentation of first-person view into a top-down semantic map. [62,61] utilize object detection to construct prior objects relationships. [39] takes both semantic segmentation and object detection to jointly train models on real and simulated data to realize sim-to-real transfer. [12] proposes a memory-based transformer policy to embed the RGB-D observation and segmentation. As to the policy learning, [56] adopts meta-reinforcement learning so that the agent can dynamically adapt to an unseen scene. These works mainly focus on navigating to seen objects in the unseen environments, while our work focuses on unseen objects in the unseen environments. So far, there are few researches on navigating to unseen objects. [60] first proposes the unseen objects (namely novel objects) and builds a knowledge graph (from the external dataset) which provides the spatial and visual relationships between seen and unseen objects. Only the seen objects are used for training. During testing, the agent can infer the location of unseen objects according to the prior spatial relationships with seen objects. [61] employs similar settings. These two works both provide the strong prior knowledge to correlate the unseen objects with seen objects. Our work transfers the knowledge (i.e. mapping the semantic embedding to the visual features and navigating with the generated features) from seen to unseen objects without such strong priors. Therefore, our task is more general and challenging.

Feature generation. Feature generation for unseen objects has been widely studied in zero-shot recognition tasks [23,1,17,16,47,46,58]. Early works [23,31] learn attribute classifiers to associate seen and unseen categories. Some works [1,16,47] learn matching functions between visual representation and semantic representation. Recently, generative adversarial network [18] has been used in the generalized zero-shot learning (GZSL) to synthesize unseen category features to train GZSL classifiers [13,32,58,59]. Our idea of generating features for the unseen object based on its semantic embedding is motivated by [58], while we focus on the navigation task more challenging than the static classification task.

Meta-learning. Meta-learning (learning to learn) adapts to new tasks efficiently through experience learned from multiple tasks. The previous methods are as follows: 1) *gradient-based* methods [42,22,45,15,3,4,14] optimize through gradient updates. 2) *metric-based* methods [51,52,54] adapt to new tasks through significant distance metrics. 3) *memory-based* methods [37,40,43,48] store the past experience as memory to learn efficiently. [14] proposes a model-agnostic algorithm MAML, which learns the parameters initialization that can perform well in new tasks within a few gradients updates. Recently, some object navigation works use the gradient-based meta-learning for adapting in new scenes [56,35] and multi-task [33]. Our work also adopts the gradient-based meta-learning, while we aim at fast adaptation in adversarial learning and effectively initial parameters learning for both feature generator and policy.

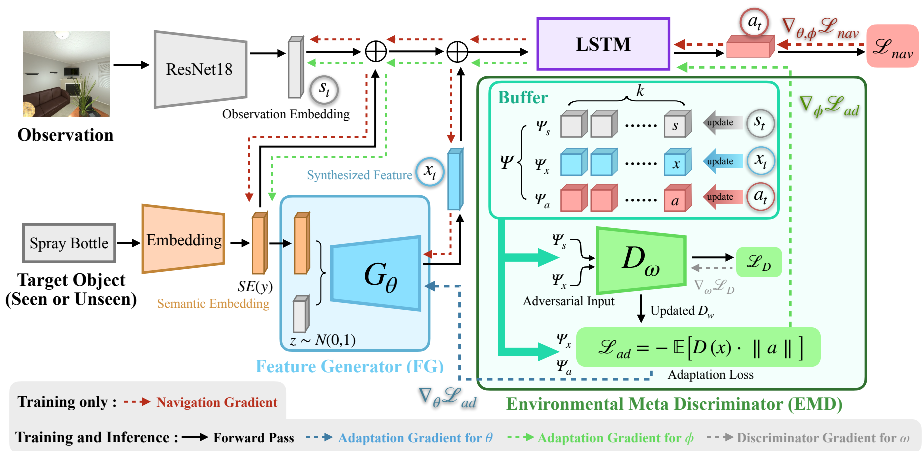


Fig. 2. Framework. Our generative meta-adversarial network (GMAN) mainly consists of a feature generator (FG) and an environmental meta discriminator (EMD). The FG synthesizes initial features of the target, then the EMD adversarially optimizes the FG to incorporate environmental features into object features. During navigation, the observation embedding s_t, generated features x_t and the action a_t are continually saved into a buffer Ψ, which is used to fast adapt the FG with meta-learning.

3 Unseen Object Navigation

3.1 Task Definition

The prevailing object navigation task requires the agent to navigate to the seen target objects in new environments. In our work, the target categories in evaluation involve both seen and unseen object classes.

Formally, let $\mathcal{Y}^s = \{y_1^s, \ldots, y_M^s\}$ denote the set of M seen target object classes. Let $\mathcal{Y}^u = \{y_1^u, \ldots, y_N^u\}$ denote the set of N unseen classes. These two sets have no intersection. Considering a set of scenes, in each navigation episode, the agent is initialized at a random location p in an environment $e \in Env$ given the target object y ($y \in \mathcal{Y}^s \cup \mathcal{Y}^u$). The agent captures an egocentric RGB image (embedded as s_t) at the timestamp t and is trained to learn a policy $\pi(a_t|s_t, y)$ that predicts an action $a_t \in \mathcal{A}$. At each time t, the agent takes action a_t until executing the termination action. **The successful episode** is defined as the situation, where the agent finally gets close to the target object within a threshold of distance and the target is visible in agent's egocentric view.

Note that there are two "unseen" concepts in our task: 1) unseen scene (environment); 2) unseen object class. In the training stage, the agent is trained with seen object classes in the seen environments, while during the evaluation stage, the agent is tested in the unseen scenes given the target category which may refer to a seen or unseen object.

3.2 A3C Baseline Model

The conventional object navigation methods [63,60,56,35] employ the Asynchronous Advantage Actor-Critic (A3C) [38] model as a baseline to learn the policy $\pi(a_t|s_t, y)$ at each timestamp. The inputs of A3C model are the current egocentric RGB image embedding (typically obtained with ResNet18 [21] pre-trained on ImageNet [9]) and the semantic embedding of the target object. The embeddings are then input to a GRU or LSTM to predict the action and the value. Generally, the agent is trained to minimize the supervised actor-critic navigation loss $\mathcal{L}_{nav}$ [63,36,56], which is used to optimize the whole model. In this paper, our GMAN follows the framework of the A3C model and additionally synthesizes the target object features to guide the unseen object navigation.

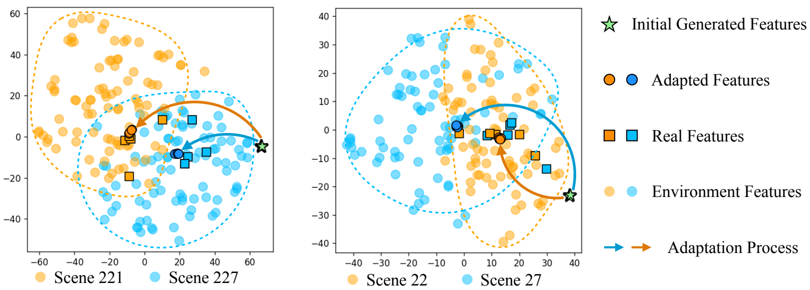
4 Generative Meta-Adversarial Network

4.1 Feature Generator

The Generative Meta-Adversarial Network (GMAN) is illustrated in Fig. 2. The feature generator (FG) module contains a generator G that synthesizes features of the unseen objects based on their semantic embedding and a random noise. The generator is formulated as $G\left(SE\left(y\right), z\right)$, where $y \in \mathcal{Y}^s \cup \mathcal{Y}^u$ is the target category, z is a random Gaussian noise and $SE\left(\cdot\right)$ is the embedding module that converts the object category into a class-specific semantic vector. The generator is pre-trained with the dataset $D_{train} = \{(x^s, SE\left(y^s\right)) \,|x^s \in \mathcal{X}^s, y^s \in \mathcal{Y}^s\}$, where $\mathcal{X}^s$ is the set of seen object features and $\mathcal{Y}^s$ is the seen object labels. We collect the dataset D_{train} in the training scenes of the AI2THOR and RoboTHOR simulators, where the agent collects several egocentric RGB images for each object $y^s \in \mathcal{Y}^s$. The images are then extracted by ResNet18 pre-trained on ImageNet to obtain object feature $x^s \in \mathcal{X}^s$. The pre-training process teaches the generator to learn the mapping from the semantic embedding to the visual features. Therefore, the pre-trained generator could synthesize the unseen objects features based on their semantic embedding. This pre-training process is similar with [58] and detailed in supplements.

4.2 Environmental Meta Discriminator

The pre-trained G initially generates the class-specific object features according to semantic embedding. However, such features imply a general representation of all seen environments, rather than the current specific unseen environment. As shown in Fig. 3, the initial generated features are far from the real features of the target (different in different scenes). These initial features are not informative to guide the agent due to the lack of current environment information. Therefore, we propose the environmental meta discriminator (EMD) to optimize the generator G to learn the environment information during navigation. The EMD consists of a navigation buffer $\Psi = [\Psi_s, \Psi_x, \Psi_a]$ and a discriminator D, where Ψ_s records the embedded observation s_t during navigation, Ψ_x records the generated target



(a) The unseen target is **armchair** (b) The unseen target is **coffee machine**

Fig. 3. The T-SNE [34] visualization of initial features and adapted features. The orange and blue colors mean different environments. The light colored circles represent the environment features (sampled by the random agent in the egocentric view). Given an unseen target, the squares represent the real features (sampled from 5 different views around the target object). The arrows represent the process of features adaptation, where the initial features are generated by the initial generator, and the adapted features are generated by the optimized generator that is adapted by our environmental meta discriminator.

features $x_t = G\left(SE\left(y\right), z\right)$ and Ψ_a records the action a_t output by the policy π. The capacity of Ψ_s, Ψ_s and Ψ_s are all set to k. Set that θ denotes the parameters of the pre-trained generator G, ω denotes the parameters of the discriminator D and ϕ denotes the remaining parameters of our model.

The generator G and discriminator D are a pair of adversarial learners and trained in a self-supervised way. We adopt the classical WGAN [20] to optimize G and D. The D aims to accurately distinguish the embedded observation features and the generated features, which is optimized by maximizing the following

$$\mathcal{L}_D = \mathbb{E}[D(s)] - \mathbb{E}[D(x)] - \lambda \mathbb{E}[(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2] \tag{1}$$

where $s \in \Psi_s$, $x \in \Psi_x$, λ is the penalty coefficient, $\tilde{x} = \varepsilon s + (1 - \varepsilon)\, x$ with $\varepsilon \in U(0, 1)$, and the $\mathbb{E}\left(\cdot\right)$ represents the mathematical expectation. The generator tries to generate realistic object features that are close to the environment features. The optimization objective of generator is to minimize the following

$$\mathcal{L}_G = -\mathbb{E}[D\left(x\right)] \tag{2}$$

As illustrated in Fig. 3, the adversarial learning (see the arrows) narrows the gap between the generated features and the real features by reducing the distance between the generated features and the environment features. The adversarial learning makes the generator capture the feature distributions of current environment, thus helping generate more informative features of the unseen objects.

Furthermore, since the navigation trajectory is limited and the navigation process is dynamic compared to various static images in classification tasks [58,41], a fast adaptation of the generator, with reference to only a few observations, is also necessary to be considered. The MAML [14] provides an algorithm

Algorithm 1 The training of our GMAN.

Input: Pre-trained parameters θ. Randomly initial parameters ω and ϕ. Buffer $\Psi = [\Psi_s, \Psi_x, \Psi_a]$. The buffer length k. The learning rate α_1, α_2, β. The distribution over training tasks $p(\mathcal{T})$.

while not converged **do**
  Sample batch of tasks $\tau_i \sim p(\mathcal{T})$
  for all τ_i **do**
    $\theta_i \leftarrow \theta$, $\phi_i \leftarrow \phi$, $t \leftarrow 0$
    while termination action is not issued **do**
      Obtain the observation embedding s_t
      Generate target features $x_t \leftarrow G_{\theta_i}$
      Take action a_t from $\pi_{\theta_i,\omega,\phi_i}$
      $t \leftarrow t + 1$
      if t is not divisible by k **then**
        Update Ψ_s by $\Psi_s \cup \{s_t\}$
        Update Ψ_x by $\Psi_x \cup \{x_t\}$
        Update Ψ_a by $\Psi_a \cup \{a_t\}$
      if t is divisible by k **then**
        Calculate $\mathcal{L}_D$ with Ψ_s, Ψ_x (Eq. 1)
        $\omega \leftarrow \omega + \alpha_1 \nabla_\omega \mathcal{L}_D(\omega, \theta_i)$
        Calculate $\mathcal{L}_{ad}$ with Ψ_x, Ψ_a (Eq. 3)
        $\theta_i \leftarrow \theta_i - \alpha_2 \nabla_{\theta_i} \mathcal{L}_{ad}(\omega, \theta_i, \phi_i)$
        $\phi_i \leftarrow \phi_i - \alpha_2 \nabla_{\phi_i} \mathcal{L}_{ad}(\omega, \theta_i, \phi_i)$
        Empty Ψ_s, Ψ_x and Ψ_a
  $\theta \leftarrow \theta - \beta \sum_{\tau_i \sim p(\mathcal{T})} \nabla_\theta \mathcal{L}_{nav}(\omega, \theta_i, \phi_i)$
  $\phi \leftarrow \phi - \beta \sum_{\tau_i \sim p(\mathcal{T})} \nabla_\phi \mathcal{L}_{nav}(\omega, \theta_i, \phi_i)$

Output: θ, ω, ϕ

to find the optimal initial parameters, which could fit the sub-tasks within only a small number of adaptation steps. Inspired by MAML, we introduce the meta-learning into the adversarial learning. The MAML and its variants consist of an inner-loop and an outer-loop. The inner-loop updates the initial parameters through a few gradient steps to achieve great performance on a specific task. The outer-loop is to minimize the total loss on all tasks. The inner-loop is executed in both training and inference, while the outer-loop is only conducted in training.

In our case, we regard each episode in navigation as a new task. In the inner-loop, we expect to obtain a well-adapted generator and a wise policy which takes the generated features as the input. Therefore, we propose the adaptation loss $\mathcal{L}_{ad}$ to optimize the generator and the policy by minimizing the following

$$\mathcal{L}_{ad} = -\mathbb{E}[D\left(x\right) \cdot \|a\|] \tag{3}$$

where $a \in \Psi_a$ represents the action output by the policy function π. Each dimension of $a \in \mathbb{R}^{1\times 6}$ denotes the probability of each action. The adaptation loss $\mathcal{L}_{ad}$ is based on $\mathcal{L}_G$ (Eq. 2) with additional optimization to the policy function π rather than only to the generator. The intuition behind multiplying $-D\left(x\right)$

and $\|a\|$ is to encourage those policies whose decisions are made based on more realistic features (i.e. $-D(x)$ is lower). When calculating the gradient $\nabla_\phi \mathcal{L}_{ad}$ for the policy (line 19 in Algorithm 1), $-D(x)$ is regarded as the weight of $\nabla_\phi \mathbb{E}[\|a\|]$ (i.e. the policy with more realistic features will have a lower loss). Considering $p(\mathcal{T})$ as the distribution of training episodes, in every training sample $\tau_i \sim p(\mathcal{T})$, the generator parameters and policy parameters are firstly optimized by adaptation loss $\mathcal{L}_{ad}$ and updated to θ_i and ϕ_i. Then the outer-loop minimizes the total loss over all episodes. The training objective of the outer-loop is given by

$$\min_{\theta,\phi} \sum_{\tau_i \sim p(\mathcal{T})} \mathcal{L}_{nav}(\omega, \theta_i, \phi_i) \tag{4}$$

where $\mathcal{L}_{nav}$ is the navigation loss. Algorithm 1 summarizes the details of our method for training. Note that the D (with parameter ω) is optimized over all training episodes and the G is optimized in each episode with only a few iterations different from many iterations in prevailing works [58,41,59]. The inference process is similar to the training, except that the line 21 and 22 are removed.

5 Experiments

5.1 Experiment Setup

Datasets. We employ two editable simulators AI2THOR [29] and RoboTHOR [8], which are detailed in supplements. To guarantee that unseen objects do not appear in the seen (training) scenes, we edit the simulators to remove the unseen objects from the training scenes. We choose 24 types seen objects and 12 types unseen objects in AI2THOR, and 10 types seen objects and 4 types unseen objects in RoboTHOR. The split of the seen and unseen object categories is detailed in supplements. For both simulators, each validation set is used to select the best model, which is then respectively evaluated for 1000 episodes on seen and unseen objects. All experimental results are repeated three times and presented with the mean and standard deviation (in small gray font). Additionally, although our method focuses on the navigation task of unseen objects, the evaluations for seen objects (typically in the conventional navigation task) are also included.

Semantic Embedding. Semantic embedding serves as a bridge that transfers the knowledge of visual feature generation from seen to unseen objects. Therefore, selecting an informative semantic embedding for object categories is critical for generating discriminative object features. Previous object navigation works [56,60,10] typically utilize the Glove [44] or FastText [24] as the semantic embedding for the object category, while zero-shot learning works [58,41] generally employ the attribute vector as the semantic embedding. Each dimension of the attribute vector represents the probability of containing such an attribute. We evaluate two semantic embeddings: Glove and attribute vector (detailed in the supplements), and find that the attribute vector achieves better navigation performance. Therefore, we adopt the attribute vector as semantic embedding.

Table 1. The impact of different rewards. We compare the effect of the navigation reward R_n, distance reward R_{n+d}, similarity reward R_{n+s} and the mixed reward R_{n+d+s}.

Reward	Unseen Objects				Seen Objects			
	SR↑ (%)	SPL↑ (%)	EPA (%)	DTS↓ (m)	SR↑ (%)	SPL↑ (%)	EPA (%)	DTS↓ (m)
R_n	14.50 1.15	6.95 0.52	5.65 0.74	1.24 0.02	27.03 0.72	12.46 0.66	7.88 0.08	1.20 0.01
R_{n+d}	18.43 0.50	8.23 0.11	5.92 0.61	1.20 0.04	30.83 0.85	14.14 0.56	9.08 0.25	1.11 0.02
R_{n+s}	21.43 0.75	**9.60** 0.38	6.51 0.45	1.16 0.01	29.53 0.35	13.14 0.61	9.24 0.29	1.11 0.01
R_{n+d+s}	**21.50** 1.23	9.36 0.66	8.56 0.15	**1.09** 0.01	**31.37** 1.00	**14.19** 0.30	9.36 0.18	**1.06** 0.01

Every input object category is converted into an attribute vector through the semantic embedding module. Note that such converting process is pre-defined and does not require the user's involvement. More details of the attribute vector are also introduced in the supplements.

Rewards. We experiment with the following four rewards to train the agent.

Navigation reward R_n. The previous works [62,56,60,35,10] typically employ the navigation reward R_n that penalizes each step with -0.01 and rewards agent for successfully finding the target object with 5.

Distance reward R_{n+d}. The distance reward R_{n+d} is based on the R_n with an additional reward $r_d = max\left(0.1 \cdot \left(Dis\left(s_t, y\right) - Dis\left(s_{t+1}, y\right)\right), 0\right)$ for each step, where $Dis\left(s_t, y\right)$ computes the Euclidean distance from state s_t to target y, and s_{t+1} is the transferred state after performing the action a_t.

Similarity reward R_{n+s}. To enhance the transfer ability from seen to unseen objects, the similarity reward R_{n+s} is designed to add an additional reward r_s to R_n, whenever the object (e.g. orange) found by the agent at last is similar to the target (e.g. tangerine) in some semantic aspects. The r_s is defined as $r_s = 0.1 S_{jaccard}\left(SE\left(y\right), SE\left(V\left(s_{done}\right)\right)\right)$, where $V\left(s_{done}\right)$ represents the visible object categories when the agent executes the action *Done*, $SE\left(\cdot\right)$ represents the semantic embedding (i.e. the attribute vector) of an object, and $S_{jaccard}\left(\cdot\right)$ computes the Jaccard similarity. When $V\left(s_{done}\right)$ contains multiple objects, we only consider the one that maximizes the $S_{jaccard}$.

Mixed reward R_{n+d+s}. We also combine the distance reward R_{n+d} and the similarity reward R_{n+s} for the experiment.

Evaluation metrics. We evaluate models using Success Rate (SR), Success weighted by Path Length (SPL), Exploration Area (EPA) and Distance To Goal in meters (DTS). These metrics are detailed in the supplements and the "Success" (i.e. successful episode) is defined in Sec. 3.1. In the following results, ↑ indicates that the larger value is better, while ↓ indicates the opposite.

Implementation details. Following primary recommendation of [60,56,35], the action set is defined as $\mathcal{A} = \{MoveAhead, RotateLeft, RotateRight, LookUp, LookDown, Done\}$. The horizontal rotation angle is set as 45 degrees while the pitch angle is 30 degrees. The action *Done* is decided by the agent rather than the simulator. The generator G and discriminator D are implemented as 2 fully connected layers. The G following [58,59] has 4096 hidden units, while D is adjusted to have 1024 hidden units. We train our model using reinforcement

Table 2. The ablation studies on different components (FG and EMD) with two baselines (A3C and A3C†(i.e. A3C with the PS module)).

PS	FG	EMD	Unseen Objects SR↑ (%)	Unseen Objects SPL↑ (%)	Unseen Objects EPA (%)	Unseen Objects DTS↓ (m)	Seen Objects SR↑ (%)	Seen Objects SPL↑ (%)	Seen Objects EPA (%)	Seen Objects DTS↓ (m)
			21.50 1.23	9.36 0.66	8.56 0.15	1.09 0.01	31.37 1.00	14.19 0.30	9.36 0.18	1.06 0.01
	✓		23.20 1.82	8.87 0.41	8.68 0.03	1.09 0.03	34.20 1.42	15.11 0.72	8.85 0.27	1.06 0.01
	✓	✓	28.03 0.91	13.02 0.14	8.71 0.52	1.08 0.02	39.30 0.87	15.61 0.14	10.46 0.17	0.98 0.02
✓			32.70 0.75	20.30 0.38	8.02 0.07	1.17 0.01	48.80 0.17	26.85 0.42	10.09 0.06	1.02 0.01
✓	✓		34.20 1.56	19.64 0.24	10.99 0.06	1.13 0.02	50.73 0.38	26.38 1.32	12.03 0.48	0.99 0.01
✓	✓	✓	**48.83** 0.60	**25.09** 0.37	10.04 0.09	**0.93** 0.01	**57.80** 0.78	**28.41** 0.66	14.12 0.15	**0.91** 0.03

Table 3. Comparisons on different EMD variants. "CosSim" replaces the discriminator with the cosine similarity. "w/o meta" removes the meta-learning from our EMD.

EMD variants	Unseen Objects SR↑ (%)	Unseen Objects SPL↑ (%)	Unseen Objects EPA (%)	Unseen Objects DTS↓ (m)	Seen Objects SR↑ (%)	Seen Objects SPL↑ (%)	Seen Objects EPA (%)	Seen Objects DTS↓ (m)
CosSim	46.47 0.31	24.20 0.26	14.58 0.28	0.98 0.02	53.63 0.58	24.66 0.56	15.73 1.07	0.92 0.03
w/o meta	41.93 0.80	21.13 0.40	10.73 0.41	1.01 0.01	53.47 0.92	25.68 0.15	13.09 0.21	0.93 0.03
EMD (ours)	**48.83** 0.60	**25.09** 0.37	10.04 0.09	**0.93** 0.01	**57.80** 0.78	**28.41** 0.66	14.12 0.15	**0.91** 0.03

learning with 12 asynchronous workers. The inner-loop is updated by SGD, while the outer-loop is optimized by Adam [27]. The learning rates (α_1, α_2, β) are all set to 10^{-4}. The penalty coefficient is $\lambda = 10$. The buffer length is set to $k = 20$.

5.2 Methods For Comparison

We compare the following methods: 1) **Random**: The agent adopts a random action at each step. 2) **A3C**: The baseline model described in Sec. 3.2. 3) **SP** [60]: The agent navigates using scene priors knowledge graph extracted from the external dataset. 4) **SAVN** [56]: The agent minimizes the self-supervised loss to optimize the policy function with MAML for fast adaptation in unseen environments. 5) **EOTP** [35]: The agent is based on SAVN with additional attention probability module, which encodes semantic and spatial information.

Inspired by the effectiveness of the similarity rewards R_{n+s} on unseen objects (discussed in Sec. 5.3), an intuitive idea is that the similarity of the semantic embedding between the target object $SE(y)$ and the objects appearing in current observation $SE(V(s_t))$ may be beneficial for navigating to unseen objects. Therefore, a simple module PS is proposed to take the current egocentric view as the input and Predicts the Semantic embedding of all contained objects. The PS is pre-trained using collected images and the semantic embedding ground truth $\mathbb{E}_{y' \in V(I)}\left(SE\left(y'\right)\right)$, where $V(I)$ is the set of visible objects in the image I. The PS is implemented with the ResNet pre-trained on ImageNet. The pre-training only uses the seen objects in the seen environments. Thus, there is another experimental group. 6) **A3C**†: The output of the PS is concatenated with the semantic embedding of the target object, together input to the LSTM. The A3C† (the A3C model equipped with the PS) is defined as another baseline. 7) **SP**†**, SAVN**†**, EOTP**†: All original methods are equipped with the PS.

There are other methods [5,10,7,61] for object navigation. However, these methods require pre-trained visual clues such as object detection or instance segmentation to construct object relation graphs [10,61] or semantic maps [5,7]. These methods are inapplicable to unseen objects because they require unseen object detection or segmentation, so that we modify them by fairly adding the PS module and the mixed reward. The comparisons are detailed in the supplements.

5.3 Evaluation Results

The impact of rewards. We use the A3C baseline to investigate the effect of rewards R_n, R_{n+d}, R_{n+s} and R_{n+d+s} , as shown in Tab. 1. Compared to the navigation reward R_n, the distance reward R_{n+d} provides the distance information of the target. Thus, the R_{n+d} improves the efficiency of RL training, thereby significantly improving the performance on both seen and unseen objects. Since the similarity reward R_{n+s} gives additional rewards which encourage the agent to find semantically similar objects. Thereby R_{n+s} enhances the transfer of navigation ability from seen to unseen objects (correlated through semantic embedding) and achieves more improvement on the unseen objects. Additionally, combining the advantages of R_{n+d} and R_{n+s}, the mixed reward R_{n+d+s} achieves the best performance on both seen and unseen objects. As a result, we choose the mixed reward R_{n+d+s} to train all models, including our method and the related works.

Ablation studies on different modules. We choose two baselines (A3C and A3C†) for ablation studies as shown in Tab. 2. Directly using the FG brings a slight improvement for both two baselines, which demonstrates that directly employing feature generation methods without considering current environmental background is not enough for the unseen object navigation task. Comparatively, combining FG with EMD gains significant improvement especially on unseen objects, indicating that the continuous adaptation of the FG to obtain more environment information is necessary. Furthermore, the baseline A3C† also significantly outperforms A3C, which shows the effectiveness of the PS module, further indicating that our semantic embedding of the unseen object is meaningful so that the semantic similarity in the PS module can play its value. Besides, combining FG, EMD and PS obtains the best performance on both seen and unseen objects, again indicating that these modules could complement each other.

Comparisons on some EMD variants. To further explore the optimal structure of the proposed EMD, based on the GMAN† (line 6 in Tab. 2), some EMD variants are considered as shown in Tab. 3. The first variant (line 1) replaces the discriminator D with cosine similarity to calculate the similarity between the generated features x_t and the environment features s_t. The results indicate that the cosine similarity does bring some improvements compared with that of no discriminator (line 5 in Tab. 2), while the improvement is less than

Table 4. Comparisons with the related works for navigation in unseen environments on AI2THOR simulator. The "†" indicates the combination with the PS module.

Method	Unseen Objects				Seen Objects			
	SR↑ (%)	SPL↑ (%)	EPA (%)	DTS↓ (m)	SR↑ (%)	SPL↑ (%)	EPA (%)	DTS↓ (m)
Random	6.70 1.01	3.58 0.67	4.01 0.03	1.57 0.02	6.23 0.60	3.63 0.30	3.89 0.06	1.53 0.04
A3C	21.50 1.23	9.36 0.66	8.56 0.15	1.09 0.01	31.37 1.00	14.19 0.30	9.36 0.18	1.06 0.01
SP [60]	22.43 1.71	9.60 1.25	7.21 0.35	1.16 0.01	34.00 0.87	13.23 0.87	8.33 0.11	1.13 0.02
SAVN [56]	17.63 0.59	4.69 0.61	9.76 1.74	1.19 0.04	35.87 1.48	13.47 0.58	10.99 0.37	1.02 0.02
EOTP [35]	19.90 0.40	4.36 0.41	11.89 0.07	**0.99** 0.01	36.97 0.39	14.56 0.32	11.03 0.16	0.98 0.02
GMAN (ours)	**28.03** 0.91	**13.02** 0.14	8.71 0.52	1.08 0.02	**39.30** 0.87	**15.61** 0.14	10.46 0.17	**0.98** 0.02
A3C†	32.70 0.75	20.30 0.38	8.02 0.07	1.17 0.01	48.80 0.17	26.85 0.42	10.09 0.06	1.02 0.01
SP†	38.00 0.40	21.77 1.38	8.98 0.45	1.06 0.01	49.40 0.61	26.84 0.49	9.81 0.04	1.02 0.02
SAVN†	41.47 1.26	18.97 0.90	15.64 0.14	1.00 0.01	53.63 0.68	23.31 0.43	16.22 0.07	**0.89** 0.01
EOTP†	38.57 1.19	15.44 0.25	11.41 0.06	1.03 0.01	52.87 1.19	**29.50** 0.58	10.34 0.10	0.95 0.02
GMAN† (ours)	**48.83** 0.60	**25.09** 0.37	10.04 0.09	**0.93** 0.01	**57.80** 0.78	28.41 0.66	14.12 0.15	0.91 0.03

the proposed EMD. Compared to the fixed similarity measurement (cosine distance), the discriminator D seems to be a better "learnable measurement". The second variant without meta-learning (only through adversarial learning) can also obtain improvements than that of no discriminator (line 5 in Tab. 2), while is still inferior to our method. The results indicate that meta-learning is indeed a powerful tool to improve performance.

5.4 Comparisons with the Related Works

The experimental results in AI2THOR and RoboTHOR are shown in Table 4 and 5. The SP attempts to navigate to unseen objects, which is task-related to our GMAN. Both SAVN and EOTP are structured in MAML-liked reinforcement learning, which is framework-related to our GMAN. However, SAVN and EOTP focus on seen objects and achieve poor performance on unseen objects. For a fair comparison, we enhance all related works from two aspects. 1) **The mixed reward**. All related works are implemented with the mixed reward that is conducive to unseen objects. Therefore, the SAVN and EOTP can improve the navigation ability on unseen objects although the performances are still lower than the SP. The SP benefits from its object relation graph and outperforms the A3C, SAVN and EOTP on unseen objects. 2) **The PS module**. All related works are equipped with the PS (i.e. under the A3C† baseline), which are denoted with the superscript †. Since the PS compares the semantic similarity of current view and the target object, all methods gain a significant improvement on navigating to unseen objects. Besides, the MAML-liked methods SAVN† and EOTP† outperform the SP†, which indicates that the MAML-liked methods get more benefit from the semantic similarity information.

Comparing our GMAN with the related works on unseen objects, both GMAN and GMAN† outperform the related works with a large margin. Significantly, under A3C† baseline, the GMAN† outperforms the related works by 7.36% in SR,

Table 5. Comparisons with the related works for navigation in unseen environments on RoboTHOR simulator.

Method	Unseen Objects				Seen Objects			
	SR↑ (%)	SPL↑ (%)	EPA (%)	DTS↓ (m)	SR↑ (%)	SPL↑ (%)	EPA (%)	DTS↓ (m)
Random	2.12 0.91	1.03 0.35	6.53 0.09	2.26 0.02	2.30 0.26	1.11 0.12	6.58 0.10	2.22 0.08
A3C	9.73 0.06	5.06 0.61	7.69 0.08	2.11 0.01	11.33 0.64	5.39 0.33	9.46 0.09	2.18 0.01
SP [60]	11.37 0.76	5.52 0.37	8.76 0.19	2.08 0.05	10.33 0.64	5.03 0.38	9.76 0.52	2.17 0.02
SAVN [56]	11.00 0.35	4.32 0.30	9.86 0.42	1.98 0.02	13.93 0.38	6.02 0.50	10.79 0.48	**1.97** 0.10
EOTP [35]	11.30 0.62	4.39 0.33	10.56 0.16	1.99 0.01	14.53 0.91	6.25 0.71	11.12 0.17	2.04 0.05
GMAN (ours)	**13.27** 0.32	**5.60** 0.35	10.39 0.14	**1.96** 0.05	**15.07** 0.15	**6.45** 0.38	11.06 0.12	2.02 0.05
A3C†	10.87 0.51	7.26 0.38	8.97 0.08	2.17 0.02	17.23 0.06	11.78 0.05	10.49 0.09	2.11 0.01
SP†	12.93 0.93	7.33 0.45	10.82 0.14	2.11 0.01	18.67 0.46	11.89 0.47	11.00 0.12	2.09 0.01
SAVN†	23.97 0.30	13.02 0.80	12.23 0.23	1.73 0.03	31.90 0.70	17.01 0.78	13.45 0.10	1.82 0.01
EOTP†	21.53 0.45	10.38 0.41	14.32 0.03	1.87 0.03	36.43 1.06	18.94 0.75	14.28 0.09	1.81 0.03
GMAN† (ours)	**27.67** 0.67	**14.29** 0.37	12.47 0.02	**1.68** 0.02	**37.10** 0.61	**19.12** 0.12	13.65 0.04	**1.78** 0.02

3.32% in SPL, and -0.07m in DTS in AI2THOR and 3.70% in SR, 1.27% in SPL, and -0.05m in DTS in RoboTHOR. The results reveal the great advantage of our GMAN on unseen objects. As for the seen objects, since our GMAN is based on the MAML-liked methods with a generative module that generates the features of the target objects, both GMAN and GMAN† can also improve the performance on seen objects, despite that the improvement is less outstanding than that on unseen objects. The results indicate that the generated features bring limited improvements to those well-trained seen objects.

Note that the reported results of SP, SAVN and EOTP is basically consistent with [56,35] while different from [60]. Because our setting (24 types seen objects and 12 types unseen objects) has huge difference with [60] (46 types seen objects and 11 types unseen objects) but is similar to [56,35] (18 types seen objects).

6 Conclusions

In this paper, we propose a generative meta-adversarial network (GMAN) for unseen object navigation. Our method is composed of a feature generator (FG) and an environmental meta discriminator (EMD). The FG synthesizes the object features by learning the mapping from semantic embeddings to features. The EMD adapts the FG with adversarial learning to let FG learn the background information of the navigation scene. Besides, meta-learning is introduced to the adversarial learning for fast adaptation. Experimental results on AI2THOR and RoboTHOR show the effectiveness of our method on unseen object navigation.

Acknowledgement. This work was supported by National Key Research and Development Project of New Generation Artificial Intelligence of China, under Grant 2018AAA0102500, in part by the National Natural Science Foundation of China under Grant 62125207, 62032022, 61902378 and U1936203, in part by Beijing Natural Science Foundation under Grant Z190020, in part by the National Postdoctoral Program for Innovative Talents under Grant BX201700255.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for image classification. CoRR **abs/1503.08677** (2015)
2. Anderson, P., Chang, A., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al.: On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757 (2018)
3. Cao, T., Xu, Q., Yang, Z., Huang, Q.: Task-distribution-aware meta-learning for cold-start CTR prediction. In: MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020. pp. 3514–3522 (2020)
4. Cao, T., Xu, Q., Yang, Z., Huang, Q.: Meta-wrapper: Differentiable wrapping operator for user interest selection in ctr prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
5. Chaplot, D.S., Gandhi, D., Gupta, A., Salakhutdinov, R.R.: Object goal navigation using goal-oriented semantic exploration. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
6. Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural SLAM. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020)
7. Chaplot, D.S., Salakhutdinov, R., Gupta, A., Gupta, S.: Neural topological SLAM for visual navigation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 12872–12881 (2020)
8. Deitke, M., Han, W., Herrasti, A., Kembhavi, A., Kolve, E., Mottaghi, R., Salvador, J., Schwenk, D., VanderBilt, E., Wallingford, M., Weihs, L., Yatskar, M., Farhadi, A.: Robothor: An open simulation-to-real embodied AI platform. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 3161–3171 (2020)
9. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA. pp. 248–255 (2009)
10. Du, H., Yu, X., Zheng, L.: Learning object relation graph and tentative policy for visual navigation. In: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VII. pp. 19–34 (2020)
11. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. Computer **22**(6), 46–57 (1989)
12. Fang, K., Toshev, A., Li, F., Savarese, S.: Scene memory transformer for embodied agents in long-horizon tasks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 538–547. Computer Vision Foundation / IEEE (2019)
13. Felix, R., Kumar, B.G.V., Reid, I.D., Carneiro, G.: Multi-modal cycle-consistent generalized zero-shot learning. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 11210, pp. 21–37. Springer (2018)

14. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 1126–1135 (2017)
15. Frans, K., Ho, J., Chen, X., Abbeel, P., Schulman, J.: Meta learning shared hierarchies. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018)
16. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 2121–2129 (2013)
17. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Transductive multi-view zero-shot learning. IEEE Trans. Pattern Anal. Mach. Intell. **37**(11), 2332–2345 (2015)
18. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 2672–2680 (2014)
19. Gupta, S., Davidson, J., Levine, S., Sukthankar, R., Malik, J.: Cognitive mapping and planning for visual navigation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 7272–7281. IEEE Computer Society (2017)
20. Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.): Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778 (2016)
22. Hochreiter, S., Younger, A.S., Conwell, P.R.: Learning to learn using gradient descent. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) Artificial Neural Networks - ICANN 2001, International Conference Vienna, Austria, August 21-25, 2001 Proceedings. Lecture Notes in Computer Science, vol. 2130, pp. 87–94. Springer (2001)
23. Jayaraman, D., Grauman, K.: Zero-shot recognition with unreliable attributes. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 3464–3472 (2014)
24. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. CoRR **abs/1607.01759** (2016)
25. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 4401–4410 (2019)
26. Kidono, K., Miura, J., Shirai, Y.: Autonomous visual navigation of a mobile robot using a human-guided experience. Robotics Auton. Syst. **40**(2-3), 121–130 (2002)

27. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
28. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings (2017)
29. Kolve, E., Mottaghi, R., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: AI2-THOR: an interactive 3d environment for visual AI. CoRR **abs/1712.05474** (2017)
30. Kumar, A., Gupta, S., Malik, J.: Learning navigation subroutines from egocentric videos. In: Kaelbling, L.P., Kragic, D., Sugiura, K. (eds.) 3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings. Proceedings of Machine Learning Research, vol. 100, pp. 617–626. PMLR (2019)
31. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE Trans. Pattern Anal. Mach. Intell. **36**(3), 453–465 (2014)
32. Li, J., Jing, M., Lu, K., Ding, Z., Zhu, L., Huang, Z.: Leveraging the invariant side of generative zero-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 7402–7411. Computer Vision Foundation / IEEE (2019)
33. Li, J., Wang, X., Tang, S., Shi, H., Wu, F., Zhuang, Y., Wang, W.Y.: Unsupervised reinforcement learning of transferable meta-skills for embodied navigation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 12120–12129 (2020)
34. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
35. Mayo, B., Hazan, T., Tal, A.: Visual navigation with spatial attention. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 16898–16907 (2021)
36. Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., Hadsell, R.: Learning to navigate in complex environments. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings (2017)
37. Mishra, N., Rohaninejad, M., Chen, X., Abbeel, P.: A simple neural attentive meta-learner. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018)
38. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T.P., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. pp. 1928–1937 (2016)
39. Mousavian, A., Toshev, A., Fiser, M., Kosecká, J., Wahid, A., Davidson, J.: Visual representations for semantic target driven navigation. In: International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019. pp. 8846–8852. IEEE (2019)
40. Munkhdalai, T., Yu, H.: Meta networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 2554–2563. PMLR (2017)

41. Narayan, S., Gupta, A., Khan, F.S., Snoek, C.G.M., Shao, L.: Latent embedding feedback and discriminative features for zero-shot classification. In: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXII. pp. 479–495 (2020)
42. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. CoRR **abs/1803.02999** (2018)
43. Oreshkin, B.N., López, P.R., Lacoste, A.: TADAM: task dependent adaptive metric for improved few-shot learning. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. pp. 719–729 (2018)
44. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1532–1543 (2014)
45. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017)
46. Rohrbach, M., Ebert, S., Schiele, B.: Transfer learning in a transductive setting. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 46–54 (2013)
47. Romera-Paredes, B., Torr, P.H.S.: An embarrassingly simple approach to zero-shot learning. In: Bach, F.R., Blei, D.M. (eds.) Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. JMLR Workshop and Conference Proceedings, vol. 37, pp. 2152–2161. JMLR.org (2015)
48. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.P.: Meta-learning with memory-augmented neural networks. In: Balcan, M., Weinberger, K.Q. (eds.) Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR Workshop and Conference Proceedings, vol. 48, pp. 1842–1850. JMLR.org (2016)
49. Savinov, N., Dosovitskiy, A., Koltun, V.: Semi-parametric topological memory for navigation. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018)
50. Savva, M., Malik, J., Parikh, D., Batra, D., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V.: Habitat: A platform for embodied AI research. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 9338–9346. IEEE (2019)
51. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 4077–4087 (2017)
52. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 1199–1208. IEEE Computer Society (2018)

53. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artif. Intell. **99**(1), 21–71 (1998)
54. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 3630–3638 (2016)
55. Wijmans, E., Kadian, A., Morcos, A., Lee, S., Essa, I., Parikh, D., Savva, M., Batra, D.: DD-PPO: learning near-perfect pointgoal navigators from 2.5 billion frames. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020)
56. Wortsman, M., Ehsani, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 6750–6759 (2019)
57. Wu, Y., Wu, Y., Tamar, A., Russell, S.J., Gkioxari, G., Tian, Y.: Bayesian relational memory for semantic visual navigation. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 2769–2779 (2019)
58. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 5542–5551 (2018)
59. Xian, Y., Sharma, S., Schiele, B., Akata, Z.: F-VAEGAN-D2: A feature generating framework for any-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 10275–10284 (2019)
60. Yang, W., Wang, X., Farhadi, A., Gupta, A., Mottaghi, R.: Visual semantic navigation using scene priors. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019 (2019)
61. Ye, X., Yang, Y.: Hierarchical and partially observable goal-driven policy learning with goals relational graph. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 14101–14110 (2021)
62. Zhang, S., Song, X., Bai, Y., Li, W., Chu, Y., Jiang, S.: Hierarchical object-to-zone graph for object navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15130–15140 (October 2021)
63. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017. pp. 3357–3364 (2017)