Object-Goal Visual Navigation via
Effective Exploration of Relations among Historical Navigation States

Heming Du[1,3], Lincheng Li[2], Zi Huang[3], Xin Yu[3*]
[1] Australian National University [2] Netease Fuxi AI Lab [3] The University of Queensland

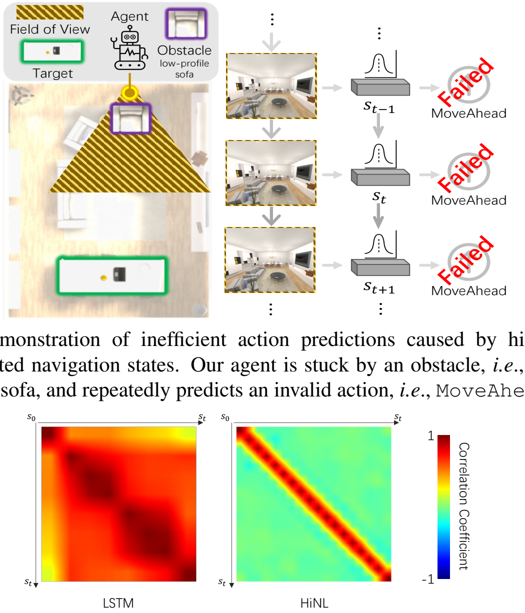
Abstract

*Object-goal visual navigation aims at steering an agent toward an object via a series of moving steps. Previous works mainly focus on learning informative visual representations for navigation, but overlook the impacts of navigation states on the effectiveness and efficiency of navigation. We observe that high relevance among navigation states will cause navigation inefficiency or failure for existing methods. In this paper, we present a History-inspired Navigation Policy Learning (HiNL) framework to estimate navigation states effectively by exploring relationships among historical navigation states. In HiNL, we propose a History-aware State Estimation (HaSE) module to alleviate the impacts of dominant historical states on the current state estimation. Meanwhile, HaSE also encourages an agent to be alert to the current observation changes, thus enabling the agent to make valid actions. Furthermore, we design a History-based State Regularization (HbSR) to explicitly suppress the correlation among navigation states in training. As a result, our agent can update states more effectively while reducing the correlations among navigation states. Experiments on the artificial platform AI2-THOR (*i.e.*, iTHOR and RoboTHOR) demonstrate that HiNL significantly outperforms state-of-the-art methods on both Success Rate and SPL in unseen testing environments.*

1. Introduction

Object-goal visual navigation is to direct an agent to move consecutively toward an object of a specific category. Without knowing the environment map beforehand, at each navigation step, an agent first needs to represent its visual observations, then estimate its navigation states from the visual representations and the preceding states, and at last predict the corresponding action. Therefore, to achieve an effective and efficient navigation system, learning instructive visual representations and navigation states is critical.

Prevailing visual navigation works [13, 14, 50] focus on extracting informative visual representations, while some methods [13, 49] adjust navigation policy during inference. All these approaches commonly employ recurrent neural networks (*e.g.*, LSTM) to estimate navigation states. However, we observe that the navigation states of existing methods [13, 14, 49] exhibit high relevance, as demonstrated in Figure 1b, and the highly-correlated navigation states would lead to inefficient navigation policy (*i.e.*, failure to respond to observation changes rapidly). For instance, as shown in Figure 1a, an agent is stuck by the low-profile sofa and fails to take proper actions to circumvent the obstacle. Hence, we aim to endow an agent with the capability of updating its navigation states effectively while avoiding producing highly-correlated states.



(a) Demonstration of inefficient action predictions caused by highly-correlated navigation states. Our agent is stuck by an obstacle, *i.e.*, low-profile sofa, and repeatedly predicts an invalid action, *i.e.*, `MoveAhead`.

(b) Demonstration of the correlation coefficients among navigation states trained in two manners. Navigation states estimated via LSTM are highly-relevant. In contrast, our HiNL produce low-correlated navigation states.

Figure 1. Motivation of our proposed History-inspired Navigation Learning (HiNL) framework.

In this work, we propose a History-inspired Navigation

[*] Corresponding author

Learning (HiNL) framework to obtain informative navigation states by exploiting the relationships among historical navigation states. HiNL consists of two novel components: (i) a History-aware State Estimation (HaSE) module, and (ii) a History-based State Regularization (HbSR). Here, our HaSE module is designed to generate a state that can be promptly updated according to visual observations. Specifically, HaSE first analyzes the correlations among historical navigation states and then eliminates the influence of dominant historical states on the current state estimation. As a result, an agent is able to predict navigation states which can dynamically react to the current visual observations and then make sensible navigation actions.

Furthermore, existing reinforcement learning-based object-goal navigation systems [13,14,49] often assume the navigation state transition exhibits the first-order Markov property. This would allow the emergence of high correlations among navigation states, leading to inferior navigation policy. To address this issue, we introduce an explicit constraint on the correlation among all the states, namely History-based State Regularization (HbSR). To be specific, HbSR enforces to relevance (*i.e*., correlations) between a state and all its preceding states (except its previous state) to be low. Here, we do not constrain states of two consecutive steps because temporally close states generally have relevance in practice considering the navigation continuity. After training with our HbSR, the correlations among the navigation states become much lower (see Figure 1b). This pheromone also indicates HiNL effectively updates states. Hence, our navigation system can respond to observation changes adaptively.

To demonstrate the superiority of HiNL, we conduct experiments in the widely-adopted artificial environment iTHOR [26] and RoboTHOR [11]. HiNL outperforms the state-of-the-art by a large margin. To be specific, we improve the Success Rate (SR) from 72.2% to 80.1% and Success weighted by Path Length (SPL) from 0.449 to 0.498 in iTHOR. Overall, our major contributions are summarized as follows:

- We propose a History-inspired Navigation Policy (HiNL) framework to effectively estimate navigation states by utilizing historical states.
- We design a History-aware State Estimation (HaSE) to eliminate dominant historical states in the current state estimation. Therefore, the agent reduces the impact of distant navigation states on the state estimation, and thus reacts dynamically to the observation changes.
- We introduce a History-based State Regularization (HbSR) to explicitly constrain the correlations among navigation states. By doing this, the agent can effectively update navigation states with low relevance.

2. Related Works

Visual navigation. Traditional works [3, 4, 32] often leverage an entire environment map for navigation and divide the task into three parts: mapping, localization, and path planning. However, environment maps are generally unavailable in unseen environments. Dissanayake *et al*. [12] adopt simultaneous localization and mapping (SLAM) to infer robot positions. Campari *et al*. [5] learn agent states via a Taskonomy model bank [52], but they need an RGB-D sensor to construct an online map during navigation.

Recently, due to significant advancements in deep learning [16,22,45,51], reinforcement learning-based navigation methods [28, 29, 31, 33, 54] take visual observations as inputs and predict navigation actions. Vision-Language Navigation (VLN) approaches [9, 10, 15, 39, 40] steer an agent to the target based on its visual observations and navigation guidance in natural language. Similar to VLN, point-goal visual navigation methods [44,48] aim at driving an agent to a given point with step-wise directional indications. Moreover, audio-visual navigation methods [7, 8, 18] utilize additional audio signals to move a robot to the target position. Al-Halah *et al*. [1] propose a transfer learning model for multiple navigation tasks by embedding various navigation goals, *e.g*., image, sketch, and audio.

Our work falls in the field of object-goal visual navigation [27,31,36,47,50]. However, existing object-goal navigation methods mainly focus on representing visual features comprehensively while we investigate the impact of navigation states on navigation performance. Wortsman *et al*. [49] exploit word embedding (*i.e*., GloVe embedding [35]) to represent the target category and introduce a meta network that mimics a reward function during inference. Du *et al*. [13] introduce an object relation graph, dubbed ORG, to encode visual observations. They also design a tentative policy for deadlock avoidance and adjust the navigation policy in unseen testing environments. The Hierarchical Object-to-Zone (HOZ) graph [53] offers coarse-to-fine guidance based on real-time updates. Additionally, VT-Net [14] incorporates object and region features with location cues, and EmbCLIP [24] leverages the contrastive language image pretraining encoder for visual navigation tasks.

Correlation Modeling in Reinforcement Learning. Several methods [37, 43] explore correlations in hidden Markov models for inverse reinforcement learning (IRL). For action prediction, Hester *et al*. [20] propose Texplore to model correlations within the transition dynamics via a random forest. Vsovsic *et al*. [42] introduce a Bayesian approach to learn policy from demonstrations of experts by capturing correlations among actions. Alt *et al*. [2] design a Bayesian learning framework to establish temporal and spatial correlations among actions. Furthermore,

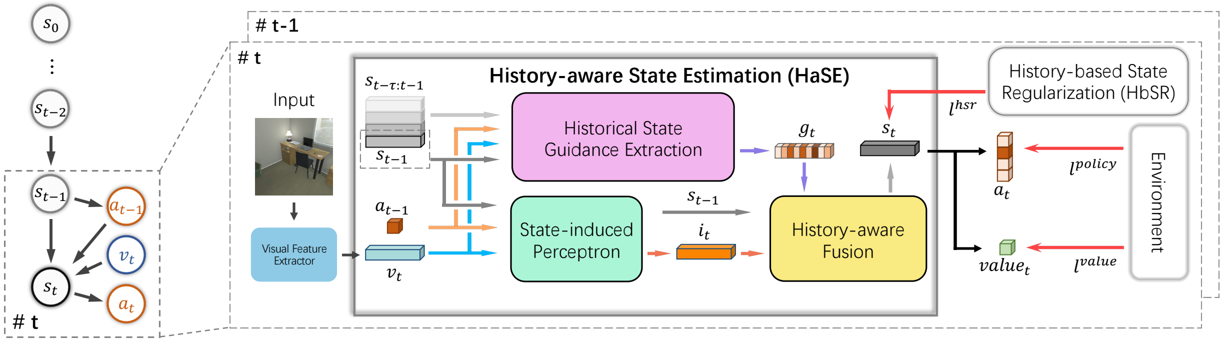


Figure 2. **Our History-inspired Navigation Policy Learning (HiNL) framework.** HiNL takes visual representations as input and outputs navigation actions. HiNL involves two innovative part: a History-aware State Estimation (HaSE) module and a History-based State Regularization (HbSR). HaSE is proposed to estimate navigation states that can reflect current observation changes from the perspective of network design, while HbSR is designed to enforce the informativeness of states from the view of the training objective. Both of them help to achieve effective and efficient navigation policy.

Sermanet [41] propose a self-supervised TCN for learning robotic behaviors and representations from unlabeled multi-viewpoint videos. TCN uses a metric learning loss to create viewpoint-invariant representations that capture relationships between end-effectors.

Different from previous methods that establish correlations among actions, our HiNL aims to eliminate high relevance among navigation states. To the best of our knowledge, our work HiNL is the first attempt to explicitly leverage the correlations among navigation states for object-goal navigation and achieves superior navigation performance.

3. Proposed Methods

In this work, we propose a History-inspired Navigation Policy (HiNL) framework to estimate navigation states effectively. As demonstrated in Figure 2, in each step, the agent first adopts a visual feature extractor to process the current RGB observation. Then, the agent estimates the current navigation state via a History-aware State Estimation (HaSE) module and predicts a navigation action. At last, we introduce a History-based State Regularization (HbSR) to constrain the correlations among navigation states during training, thus achieving low-correlated navigation states and highly effective navigation policy.

3.1. Task Definition and Setup

Following previous works [13, 14, 49, 53], in the object-goal visual navigation task, an RGB monocular camera is the only sensor available to an agent. Prior knowledge of the entire environment, such as the topological map and 3D meshes, is unknown to the agent. An environment is composed of discrete grids, and the agent selects one of six actions, *i.e.*, `MoveAhead`, `RotateLeft`, `RotateRight`, `LookUp`, `LookDown`, `Done`. Specifically, the grid size is 0.25 meters, and the rotation angles of `RotateLeft/RotateRight` and `LookUp/LookDown` are 45° and 30°, respectively.

At the start of each episode, the environment randomly chooses one of the available object categories $T \in \{Sink, \ldots, Microwave\}$ as the target class. Then, the agent is teleported to a random start position $s = \{x, y, \theta_h, \theta_v\}$, where x and y stand for the coordinates, and θ_h and θ_v represent the horizontal angle and the vertical angle for the view of the agent. The agent aims to predict an action distribution based on a learned navigation policy and operates the action with the highest probability for navigation. Then, the agent moves to the next navigation step if it selects a moving action or terminates the episode if it selects the action `Done`. After finishing the episode, we regard the agent as successful when the following three conditions are met simultaneously: (i) the agent selects the ending action `Done` within the maximum step length of an episode; (ii) there is at least one instance of the specified category in the view of the agent; (iii) the distance between the visible target and the agent is less than the threshold, *i.e.*, 1.5 meters. Otherwise, the episode is considered as a failure case.

3.2. Visual Representation Extraction

Similar to previous works [13, 49], we first adopt a deep neural network, *i.e.*, ResNet [19], to extract global features from an observed RGB image. Meanwhile, a detection module, *e.g.*, Fast-RCNN [38] and DETR [6], is introduced to extract object features. Then, the global and object features are merged into a visual representation $\boldsymbol{v_t}$ by graph neural network [13, 53] or transformer [14][1]. For convenience, we adopt $\boldsymbol{v_t}$ to denote the visual feature extracted

[1] In our experiments, we employ both CNN-based [13] and Transformer-based [14] visual representations, demonstrating our framework HiNL is agnostic against various visual representations.

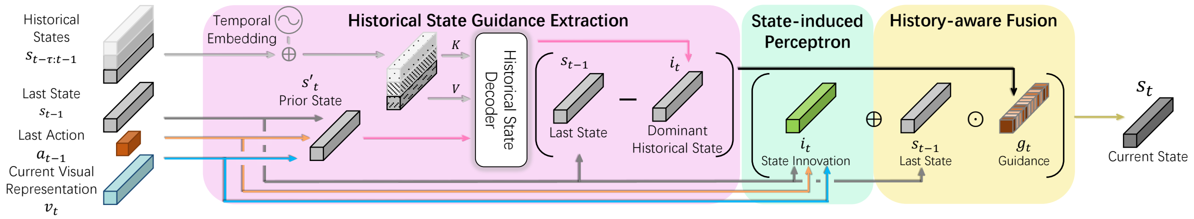


Figure 3. **Illustration of History-aware State Estimation (HaSE).** The agent first adopts a Historical State Guidance Extraction module to estimate a state guidance that indicates the dominant historical states. Then, the agent extracts a state innovation that reflects incremental information of current observations, via a State-induced Perceptron module. At last, the agent generates the current navigation state estimation by fusing the state innovation and guidance in a History-aware Fusion module.

from the RGB observation at time step t.

3.3. History-aware State Estimation

Our History-aware State Estimation (HaSE) estimates the navigation states by eliminating the influence of dominant historical states and focusing on the present observation changes. As demonstrated in Figure 3, after obtaining the visual representation $\boldsymbol{v}_t$, we design a Historical State Guidance Extraction module to predict a state guidance $\boldsymbol{g}_t$ indicating dominant historical states at time step t. Then, we introduce a State-induced Perceptron module to obtain the current state innovation $\boldsymbol{i}_t$, which encodes the new information brought by the current visual observations and the last step action. Next, the history-aware guidance $\boldsymbol{g}_t$ is employed to eliminate the dominant historical states and facilitate the estimation of the current navigation state $\boldsymbol{s}_t$ via our proposed History-aware Fusion module.

Historical State Guidance Extraction. In order to eliminate the dominant historical states in the current state estimation, we extract guidance for the dominant parts by exploring the relationships among historical navigation states. The agent roughly approximates a prior state $\boldsymbol{s}'_t$ based on the last state $\boldsymbol{s}_{t-1}$, action $\boldsymbol{a}_{t-1}$, and the current visual representation $\boldsymbol{v}_t$. The prior state $\boldsymbol{s}'_t$ is encoded with the last navigation state and the current observation, and thus we refer $\boldsymbol{s}'_t$ to the query about the dominant historical states. Let FFN be a feed-forward network, $\boldsymbol{s}'_t$ is formulated as follows,

$$\boldsymbol{s}'_t = \sigma(\mathrm{FFN}(\boldsymbol{s}_{t-1}) + \mathrm{FFN}([\boldsymbol{v}_t; \boldsymbol{a}_{t-1}])), \tag{1}$$

where $[;]$ and σ represent the concatenation operation and sigmoid function, respectively.

After obtaining $\boldsymbol{s}'_t$, the agent memorizes a set of the historical navigation states $\mathbb{S}_{t-\tau:t-1} = \{\boldsymbol{s}_{t-\tau}, \ldots, \boldsymbol{s}_{t-1}\}$ in the last τ steps. In order to explore the temporal relationships among historical navigation states, $\mathbb{S}_{t-\tau:t-1}$ are composed additively with temporal embeddings. Concretely, we adopt absolute temporal embedding [46] based on the timestamp at which a state is estimated. By doing this, the agent perceives not only the temporal orders among the last τ navigation states, but also the length of the current episode. Therefore, the agent can exploit temporal information to eliminate the distant navigation states in extracting the guidance. In addition, we analyze the impacts of different temporal embedding methods (*i.e.*, no/learnable/relative temporal embedding) in Table 2.

Afterward, we pass $\boldsymbol{s}'_t$ and $\mathbb{S}_{t-\tau:t-1}$ through the historical state decoder, *i.e.*, a cross-attention module. We refer to the output $\boldsymbol{z}_t$ as the dominant historical states, and calculate it as follows,

$$\boldsymbol{z}_t = \mathrm{Decoder}_{\mathrm{his}}(\boldsymbol{s}'_t, \mathbb{S}_{t-\tau:t-1}), \tag{2}$$

where $\mathrm{Decoder}_{\mathrm{his}}(\cdot, \cdot)$ indicates the historical state decoder, the first argument for query and the other for key/values. Then, we formulate the guidance $\boldsymbol{g}_t$ as the difference between $\boldsymbol{s}_{t-1}$ and $\boldsymbol{z}_t$ Therefore, we compute $\boldsymbol{g}_t$ as follows,

$$\boldsymbol{g}_t = \sigma(\boldsymbol{s}_{t-1} - \boldsymbol{z}_t). \tag{3}$$

In addition, we set τ as 5 in our experiments and demonstrate the impacts of different τ in Figure 5b.

State-induced Perceptron. Apart from eliminating the dominant historical states, we also aim to encourage the agent to emphasize the observed changes in the current state estimation. Therefore, we extract a state innovation $\boldsymbol{i}_t$ based on $\boldsymbol{v}_t$ to encode the observed changes at the current step. To be specific, we projects $\boldsymbol{v}_t$, $\boldsymbol{s}_{t-1}$ and $\boldsymbol{a}_{t-1}$ to $\boldsymbol{i}_t$. Inspired by LSTM [21], we compute $\boldsymbol{i}_t$ as,

$$\begin{aligned}\boldsymbol{i}_t = &\tanh(\mathrm{FFN}(\boldsymbol{s}_{t-1}) + \mathrm{FFN}([\boldsymbol{v}_t; \boldsymbol{a}_{t-1}])) \odot \\ &\sigma(\mathrm{FFN}(\boldsymbol{s}_{t-1}) + \mathrm{FFN}([\boldsymbol{v}_t; \boldsymbol{a}_{t-1}])),\end{aligned} \tag{4}$$

where $\tanh$ represents hyperbolic tangent function, and $\odot$ stands for the element-wise product.

History-aware Fusion. After obtaining the guidance $\boldsymbol{g}_t$ and state innovation $\boldsymbol{i}_t$, we aim to estimate the current navigation state $\boldsymbol{s}_t$. In order to eliminate the influence of dominant navigation states, the agent first merges $\boldsymbol{s}_{t-1}$ and $\boldsymbol{g}_t$ by multiplication. Then the agent adds $\boldsymbol{i}_t$ to the result of multiplication to incorporate the observation changes. Formally, we formulate the current navigation state $\boldsymbol{s}_t$ estimation as follows,

$$\boldsymbol{s}_t = \tanh(\boldsymbol{s}_{t-1} \odot \boldsymbol{g}_t + \boldsymbol{i}_t). \tag{5}$$

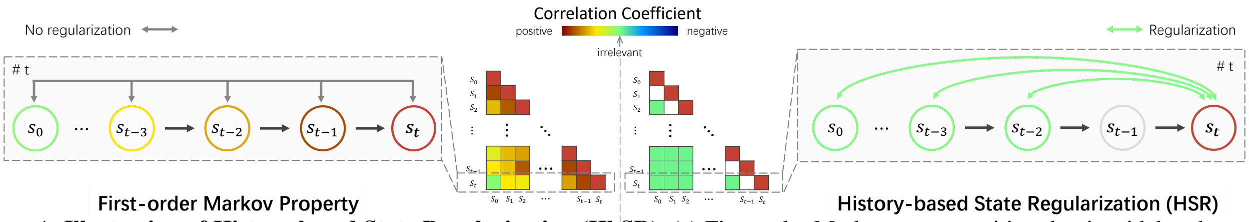


Figure 4. **Illustration of History-based State Regularization (HbSR).** (a) First-order Markov state transition that is widely adopted by LSTM based navigation policy learning. Those methods will produce highly-correlated among states. (b) Our HbSR constrains higher-order relations among states, and leads to low-correlated states. As the states are more informative (*i.e*., less correlated), our navigation policy is more effective and efficiency.

Benefiting from HaSE, the agent can effectively update the navigation states by being alert to the present observation and neglecting the influence of distant states.

3.4. History-based State Regularization

Previous reinforcement learning-based object-goal visual navigation methods are designed based on the first-order Markov assumption, while there are no explicit constraints on the correlations among navigation states. We observe that high relevance among navigation states exhibit in the experiments of previous methods [13, 14] and this is also one of the reasons that prior works yield failure cases. For instance, an agent is stuck in front of an object rather than adopting a rotation action or trapped in a loop. From these failure cases, we can see that the navigation states are highly correlated as the adopted actions are periodically the same. In order to avoid inefficient navigation caused by highly-relevant navigation states, we aim to exert an explicit constraint on the correlations among navigation states.

We introduce a novel History-based State Regularization (HbSR) to reduce the correlations among navigation states from the perspective of the training objective. In order to measure the correlations among navigation states, we adopt the Pearson correlation coefficient [17]. To be specific, after the agent terminates an episode with T steps, we obtain a set of navigation states $\mathbb{S}_{0:T} = \{\boldsymbol{s_0}, ..., \boldsymbol{s_T}\}$. Given two navigation states $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$, we first compute the covariance of the two states as follows,

$$\mathrm{Cov}(\boldsymbol{s_i}, \boldsymbol{s_j}) = \mathbb{E}((\boldsymbol{s_i} - \mathbb{E}(\boldsymbol{s_i}))\,(\boldsymbol{s_j} - \mathbb{E}(\boldsymbol{s_j}))). \quad (6)$$

Then, we compute the correlation coefficient ρ for two states based on the covariance formulated as,

$$\rho(\boldsymbol{s_i}, \boldsymbol{s_j}) = \frac{\mathrm{Cov}(\boldsymbol{s_i}, \boldsymbol{s_j})}{\sqrt{\mathrm{Var}(\boldsymbol{s_i})\;\mathrm{Var}(\boldsymbol{s_j})}}, \quad (7)$$

where $\mathrm{Var}(\boldsymbol{s_i})$ denotes the variance of $\boldsymbol{s_i}$.

The correlation coefficient is expressed in a range $[-1, +1]$. Two states are highly relevant when their correlation is close to either 1 or -1, whereas $\rho = 0$ indicates irrelevant between two states. Therefore, in order to eliminate high relevance among states, we enforce the correlations of states to be close to 0. In particular, since inefficient navigation is generally caused by positive correlations of navigation states, we only apply the proposed regularization on correlations with positive values. According to the first-order Markov property, the present state $\boldsymbol{s_t}$ depends on its immediately preceding state $\boldsymbol{s_{t-1}}$. Thus, we define a regularization-free threshold ξ. Moreover, in HbSR, if a temporal interval between two states is less than the threshold ξ, we do not apply the correlation regularization among consecutive states. This is because navigation has continuity and neighboring states will inherently have relevance while distant states should be irrelevant. Hence, our HbSR l^{hsr} is expressed as,

$$l^{hsr} = \mathop{\mathbb{E}}_{\boldsymbol{s}\in\mathbb{S}}\Big(\sum_{t=\xi+1}^{T}\ \sum_{i=0}^{t-\xi-1} \max(0, \rho(\boldsymbol{s_t}, \boldsymbol{s_i}))\Big). \quad (8)$$

3.5. Training Details

We employ the Asynchronous Advantage Actor-Critic (A3C) Algorithm [30] to train the agent, similar to previous works [13, 14, 49, 53]. We adopt two loss objectives, *i.e*., policy loss l_{policy} and value loss l_{value}, for training the navigation policy and value estimation networks, respectively. Furthermore, since the calculation of correlation coefficients is differentiable [34], we can employ the loss l^{hsr} of HbSR in training. Formally, the total loss L is formulated as the summation of l_{policy}, l_{value} and l^{hsr},

$$l = \lambda_0 l^{policy} + \lambda_1 l^{value} + \lambda_2 l^{hsr}, \quad (9)$$

where λ_0, λ_1 and λ_2 are weighted factors to balance three losses. In our experiments, we set these three weighted factors as 1, 0.5 and 1, respectively.

4. Experiments

To validate the effectiveness and efficiency of the proposed HiNL, we conduct extensive experiments on two artificial environments, *i.e*., iTHOR [26] and RoboTHOR [11].

4.1. Protocols and Experimental details

Dataset. We adopt iTHOR [26], an artificial environment within the AI2-THOR framework, to evaluate our method.

Method	iTHOR [26]				RoboTHOR [11]			
	All		$L > 5$		All		$L > 5$	
	Success	SPL	Success	SPL	Success	SPL	Success	SPL
Random	$8.0_{\pm1.3}$	$3.6_{\pm0.6}$	$0.3_{\pm0.1}$	$0.1_{\pm0.1}$	$4.0_{\pm1.0}$	$1.6_{\pm0.4}$	$0.2_{\pm0.1}$	$0.1_{\pm0.1}$
WE	$33.0_{\pm3.5}$	$14.7_{\pm1.8}$	$21.4_{\pm3.0}$	$11.7_{\pm1.9}$	$7.0_{\pm1.2}$	$2.2_{\pm0.6}$	$4.7_{\pm0.9}$	$1.7_{\pm0.3}$
SP [50]	$35.1_{\pm1.3}$	$15.5_{\pm1.1}$	$22.2_{\pm2.7}$	$11.4_{\pm1.6}$	$10.9_{\pm0.5}$	$4.2_{\pm0.5}$	$7.2_{\pm0.6}$	$3.2_{\pm0.5}$
SAVN [49]	$40.8_{\pm1.2}$	$16.1_{\pm0.5}$	$28.7_{\pm1.5}$	$13.9_{\pm0.5}$	$10.2_{\pm1.0}$	$3.9_{\pm0.9}$	$6.9_{\pm0.8}$	$2.8_{\pm0.6}$
ORG [13]	$65.3_{\pm0.7}$	$37.5_{\pm0.1}$	$54.8_{\pm1.0}$	$36.1_{\pm0.1}$	$45.4_{\pm0.8}$	$21.2_{\pm1.2}$	$39.6_{\pm0.7}$	$18.5_{\pm0.8}$
ORG+TPN [13]	$69.3_{\pm1.2}$	$39.4_{\pm1.0}$	$60.7_{\pm1.3}$	$38.6_{\pm1.1}$	$47.8_{\pm1.2}$	$23.1_{\pm0.9}$	$42.9_{\pm0.9}$	$20.3_{\pm1.0}$
HOZ [53]	$70.6_{\pm1.7}$	$40.0_{\pm1.2}$	$62.8_{\pm1.7}$	$39.2_{\pm0.6}$	$44.3_{\pm1.8}$	$19.8_{\pm1.3}$	$37.2_{\pm1.2}$	$15.4_{\pm0.8}$
TCN [41]	$75.1_{\pm0.8}$	$47.7_{\pm0.9}$	$65.3_{\pm1.4}$	$42.1_{\pm0.8}$	$44.4_{\pm1.6}$	$26.4_{\pm1.5}$	$38.4_{\pm1.1}$	$22.0_{\pm1.2}$
VTNet [14]	$72.2_{\pm1.0}$	$44.9_{\pm0.7}$	$63.4_{\pm1.1}$	$44.0_{\pm0.9}$	$53.2_{\pm1.1}$	$27.5_{\pm1.7}$	$47.0_{\pm0.8}$	$23.3_{\pm0.7}$
HiNL †	$71.2_{\pm0.8}$	$38.9_{\pm0.9}$	$64.1_{\pm0.6}$	$37.0_{\pm0.5}$	$56.1_{\pm1.3}$	$26.4_{\pm0.8}$	$51.8_{\pm0.9}$	$23.9_{\pm0.7}$
HiNL ‡	$\mathbf{80.1_{\pm1.4}}$	$\mathbf{49.8_{\pm1.9}}$	$\mathbf{74.6_{\pm1.7}}$	$\mathbf{47.6_{\pm1.4}}$	$\mathbf{60.6_{\pm1.0}}$	$\mathbf{30.8_{\pm1.2}}$	$\mathbf{56.2_{\pm0.8}}$	$\mathbf{26.6_{\pm0.9}}$

Table 1. Comparison with existing methods. We report the average Success rate (%) and SPL (%) in iTHOR [26] and RoboTHOR [11] as well as their variances in subscripts by repeating experiments five times. $L > 5$ represents the episodes that require at least 5 steps. † and ‡ indicate using CNN-based and Transformer-based visual extractors (*i.e.*, ORG [13] and VTNet [14]), respectively.

The environment includes four types of scenes, *i.e.*, kitchen, living room, bedroom, and bathroom. Each type of scene consists of 30 rooms with different furniture items and placements. Following [13, 14, 53], we use 22 categories as the target classes and ensure that there are at least four targets in each room. We choose the first 20 out of 30 rooms per scene as the training set and equally divide the remaining 40 rooms into the validation and test set.

Furthermore, we conduct experiments in RoboTHOR [11], the other synthetic environment in the AI2-THOR framework. Different from iTHOR, each scene in RoboTHOR is separated by several clapboards. RoboTHOR consists of 89 scenes, while only 75 apartments are published for train/val (60 for training and 15 for validation). Therefore, we select the first 55 of 60 original training scenes as our training dataset and use the rest 5 scenes as the validation set. Then, we adopt the remaining 15 original validation apartments as our test set. We report the test result of the model with the highest success rate on the validation set for both iTHOR and RoboTHOR.

Evaluation metrics. We adopt two evaluation metrics, *i.e.*, success rate and Success weighted by Path Length (SPL), to assess our method performance. **Success rate** is introduced to evaluate the navigation effectiveness and calculated by $\frac{1}{N}\sum_{n=0}^{N} S_n$, where N is the number of episodes and S_n represents a binary success indicator of the n-th episode. Meanwhile, **SPL** measures the efficiency of navigation trajectories. Let L_n be the length of the n-th episode and L_{opt} indicate the length of the optimal path. SPL is formulated as $\frac{1}{N}\sum_{n=0}^{N} S_n \frac{L_n}{max(L_n, L_{opt})}$.

Implementation details. Following [14], we train the VT for 20 epochs with the supervision of expert experience. Then, we train the navigation policy for 2M episodes in total with 32 asynchronous agents. In making a robot learn to approach the target, the penalty is set to $1e^{-3}$ for every step that passes prior to arrival, and thus an agent is encouraged to navigate as quickly as possible. Meanwhile, to encourage the agent complete navigation effectively, we offer a large reward of 5 when the agent finishes a trajectory successfully. We adopt Adam [25] with a learning rate 10^{-4}. Our codes will be publicly related for reproducibility.

4.2. Comparison with Existing Methods

4.2.1 Competing Methods

We compare our method with the following ones: **Random policy.** An agent makes decisions based on a uniform action probability. Thus, the movements and stops of the agent are random. **Scene Prior (SP)** [50] utilizes the scene prior knowledge and category associations for navigation with a graph neural network on the FastText database [23]. **Word Embedding (WE)** makes use of GloVe embedding [35] to signify the target category instead of relying on detection. From trial and error, an agent establishes the connection between object appearances and GloVe embeddings. **Self-adaptive Visual Navigation (SAVN)** [49] presents a meta reinforcement learning method to assist an agent in adapting to unseen environments during inference. **Object Relationship Graph (ORG)** [13] proposes a visual representation learning method to encode relationships among categories via a graph and design a tentative policy network (TPN) to escape from deadlocks during testing. **Hierarchical Object-to-Zone (HOZ)** [53] introduces a graph to guide an agent in a coarse-to-fine manner and a online-learning mechanism to update the graph. **Time-Contrastive Networks (TCN)** [41] presents a self-supervised method, deriving robotic behaviors and representations from unlabeled

	Method	All		$L > 5$	
		Success	SPL	Success	SPL
HaSE	HiNL † w/o HaSE	68.1	37.8	60.3	35.9
	HiNL †	71.2	38.9	64.1	37.0
	HiNL ‡ w/o HaSE	77.9	45.7	71.6	43.5
	HiNL ‡	**80.1**	**49.8**	**74.6**	**47.6**
	HiNL ‡ w/o tem.	78.5	39.8	71.2	39.1
	HiNL ‡ w/ lea. tem.	79.1	45.8	73.1	43.8
	HiNL ‡ w/ rel. tem.	77.3	44.1	69.4	41.6
	HiNL ‡	**80.1**	**49.8**	**74.6**	**47.6**
HbSR	HiNL † w/o HbSR	66.8	38.1	57.8	35.8
	HiNL †	71.2	38.9	64.1	37.0
	HiNL ‡ w/o HbSR	75.4	46.9	67.0	43.8
	HiNL ‡	**80.1**	**49.8**	**74.6**	**47.6**
	HiNL ‡ neg. & pos.	75.4	46.9	67.1	43.8
	HiNL ‡	**80.1**	**49.8**	**74.6**	**47.6**

Table 2. Ablation study on different components of HiNL in iTHOR [26] environment. † and ‡ indicate using CNN-based and Transformer-based visual extractors (*i.e.*, ORG [13] and VTNet [14]), respectively. HiNL w/o tem. indicates HiNL without temporal embedding in HaSE, while HiNL w/ lea. tem. and HiNL w/ rel. tem. represent the adoption of learnable and relative temporal embedding in HiNL, respectively. HiNL neg. & pos. stands for regularizing both positive and negative correlations toward 0.

videos captured across multiple viewpoints, leveraging metric learning loss to establish viewpoint-invariant representations that encapsulate end-effector relationships. **Visual Transformer for Navigation (VTNet)** [14] explores the spatial correlations among objects and observation regions for navigation via a transformer-based network.

4.2.2 Quantitative Results

To demonstrate the superiority of our proposed learning framework, we select two different visual representation extractors: one is based on graph network (*i.e.*, ORG [13]), and the other is based on the transformer (*i.e.*, VT [14]). Table 1 indicates that both HiNL † and HiNL ‡ achieve superior results compared with their original methods (*i.e.*, ORG and VTNet). This demonstrates that given different visual representations, HiNL can improve navigation performance constantly by explicitly exploring the relationships among historical states.

To be specific, HiNL ‡ with the transformer-based extractor surpasses the existing methods by a large margin on both success rate (+7.9%/+7.4%) and SPL (+4.9%/+3.3%) in iTHOR [26]/RoboTHOR [11]. Adopting ORG as the visual representation extractor, HiNL † achieves significant improvement over the original ORG by nearly 10% in success rate in both environments. VTNet and ORG only adopt LSTM to merge well-designed visual representations into navigation states without any explicit constraints on states. The correlations among navigation states are not exploited. This experiment suggests that our HiNL leads to instructive navigation states, and thus significantly improves the effectiveness and efficiency of our navigation system.

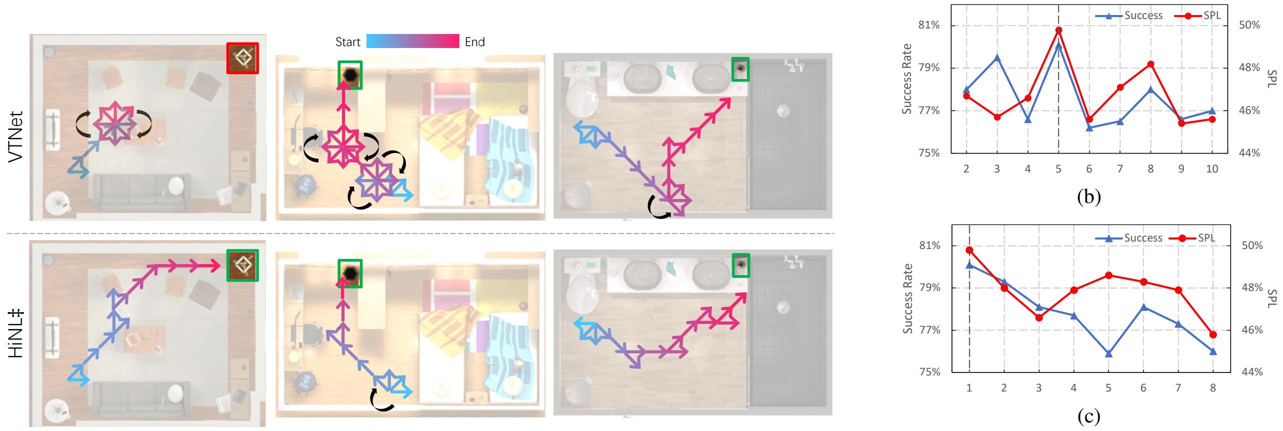
As indicated in Table 1, we observe that HiNL † significantly outperforms ORG+TPN [13]. Although TPN employs an auxiliary network to fine-tune the initial navigation policy network when the agent is stuck in deadlocks during inference, TPN does not guarantee navigation states to be low relevant. Consequently, an agent might be trapped in looping actions, such as moving with rotating, because of the high correlations among navigation states. HiNL solves this issue by reducing the correlations among navigation states, and thus achieves proper actions for navigation. Furthermore, adopting TCN-based loss (*i.e.*,TCN) yields inferior performance with 5% decrease in success rate compared to HiNL. This is because TCN tries to pull temporally neighboring features closer while consecutive navigation states might be very different due to a rotation action. Thus, TCN may overly emphasize the reliance among neighboring states, thus leading to inferior state estimation. In addition, our method achieves more than two times the success rate and SPL than SAVN in both environments. Benefiting from well-designed visual representations and lowering relevance among navigation states, HiNL achieves better navigation performance in testing even without adjusting the navigation policy. This also implies that our navigation states significantly improve the effectiveness and efficiency of navigation in unseen environments.

4.2.3 Case Study

Figure 5a illustrates trajectories of three navigation episodes proceeded by VTNet [14] and HiNL ‡. In the first case, VTNet fails to find the target and is stuck in the environment, as seen in the first column. On the contrary, HiNL ‡navigates toward the object successfully. This implies that using HiNL, our agents can react rapidly to the observed changes. In the second case, VTNet moves forward while rotating and thus leads to inefficient navigation. In contrast, the navigation system trained with HiNL avoids rotating and approaches the target with fewer steps. This demonstrates that our HiNL improves navigation efficiency by reducing the correlations among states. In the final case, VTNet moves forward consistently until it reaches the wall. Benefiting from our HaSE, the agent can get rid of dominant historical states and react rapidly to the observed changes.

4.3. Ablation Study

Impacts of different components. To illustrate the impacts of different components, we conduct experiments on adopting one of the proposed components (*i.e.*, HaSE and HbSR). Furthermore, we report the navigation performance with different visual representation extractors, *i.e.*, ORG and VT. As indicated in Table 2, both HiNL † and



(a)

Figure 5. (a) Visual results in the simulated environments iTHOR. The arrow color changes (*i.e*., from blue to red) represent the navigation progress (*i.e*., from beginning to end). Black arrows indicate rotations. The target objects are highlighted by the green/red boxes, where a green box stands for a success episode and a red box represents a failure case. (b) Impacts of different historical state length $\tau \in [2, 10]$. (c) Results of different regularization-free states $\xi \in [1, 8]$. We set $\tau = 5$ and $\xi = 1$ and highlight them by the black vertical dotted lines in (b) and (c).

HiNL ‡ achieve gains in navigation performance on success rate (+2.9%/+2.2%) and SPL (+0.9%/+4.1%) by introducing HaSE. Furthermore, HiNL † and HiNL ‡ outperform models without HbSR (*i.e*., HiNL †/HiNL ‡ w/o HbSR), respectively. This manifests that both of our proposed components improve the performance of navigation.

Impacts of different temporal embedding in HaSE. Table 2 compares different temporal embedding methods, (*i.e*., w/o temporal embedding, w/ relative temporal embedding and w/ absolute temporal embedding (HiNL)) and highlights the effectiveness of incorporating temporal embedding in historical states. HiNL with absolute temporal embedding significantly improves success rates and SPL on iTHOR compared to no embedding (HiNL ‡ w/o tem.), learnable embedding (HiNL ‡ w/ lea. tem.), and relative embedding (HiNL ‡ w/ rel. tem.). This improvement, especially in navigation efficiency, stems from providing the agent with the current episode length.

Comparison of different τ in HaSE. To illustrate the influence of different historical state lengths, we conduct experiments with $\tau \in [2, 10]$. As demonstrated in Figure 5b, we observe that as historical states become too many, our agent may fail to converge to an optimal policy. On the other hand, estimating navigation states with few historical states do not have sufficient capacity to effectively update states. The agent reaches the peak of success rate and SPL when it adopts 5 historical states in HaSE.

Impacts of different ξ in HbSR. To study the impacts of adopting different regularization-free thresholds ξ, we conduct experiments with $\xi \in [1, 8]$ in iTHOR. As seen in Figure 5c, increasing ξ drops performance clearly. Both success rate (80% to 76%) and SPL (49.8% to 45.8%) decrease when ξ increases from 1 to 8, Since two navigation states generally have high correlations when they are close in temporal, raising ξ indicates ignoring constraints on these correlations. As a result, the effects of HbSR are compromised, and the navigation performance is dropped.

Comparison of regularization on different values. To analyze the impacts of regularization on different values, we demonstrate the comparisons of applying HbSR on (i) both negative and positive correlations and (ii) only positive correlations, as indicated in Table 2. Compared to HiNL trained on regularizing the correlations based on both positive and negative values (HiNL ‡ neg. & pos.), HiNL improves performance by neglecting the negative correlations. This comparison suggests that inefficient navigation is generally caused by positively correlated navigation states.

5. Conclusion

In this paper, we proposed an effective navigation state learning approach for visual navigation, named History-inspired Navigation Policy Learning (HiNL). Our HiNL involves two innovate parts: a history-aware state estimation module (HaSE) and a history-based state regularization (HbSR). HbSR eliminates the negative impacts of dominant historical states and enables current states to reflect newly observed visual information. HbSR significantly reduces the correlations among navigation states, thus leading to more informative states as well as effective actions. Extensive results demonstrate that our HiNL significantly improves navigation performance.

Acknowledgement

This research is funded in part by ARC-Discovery grant (DP220100800 to XY), ARC-DECRA grant (DE230100477 to XY) and ARC Centres of Excellence grant (CE200100025). We thank all anonymous reviewers and ACs for their constructive suggestions.

References

[1] Ziad Al-Halah, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17031–17041, 2022. 2

[2] Bastian Alt, Adrian Šošić, and Heinz Koeppl. Correlation priors for reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[3] Johann Borenstein and Yoram Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on systems, Man, and Cybernetics*, 19(5):1179–1187, 1989. 2

[4] Johann Borenstein and Yoram Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE transactions on robotics and automation*, 7(3):278–288, 1991. 2

[5] Tommaso Campari, Leonardo Lamanna, Paolo Traverso, Luciano Serafini, and Lamberto Ballan. Online learning of reusable abstract models for object goal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14870–14879, 2022. 2

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[7] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15516–15525, 2021. 2

[8] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *European Conference on Computer Vision*, pages 17–36. Springer, 2020. 2

[9] Jinyu Chen, Chen Gao, Erli Meng, Qiong Zhang, and Si Liu. Reinforced structured state-evolution for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15450–15459, 2022. 2

[10] Kevin Chen, Juan Pablo de Vicente, Gabriel Sepulveda, Fei Xia, Alvaro Soto, Marynel Vázquez, and Silvio Savarese. A behavioral approach to visual navigation with graph localization networks. *arXiv preprint arXiv:1903.00445*, 2019. 2

[11] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3164–3174, 2020. 2, 5, 6, 7

[12] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241, 2001. 2

[13] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation. *arXiv preprint arXiv:2007.11018*, 2020. 1, 2, 3, 5, 6, 7

[14] Heming Du, Xin Yu, and Liang Zheng. Vtnet: Visual transformer network for object goal navigation. *arXiv preprint arXiv:2105.09447*, 2021. 1, 2, 3, 5, 6, 7

[15] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 538–547, 2019. 2

[16] Xiaoyu Feng, Heming Du, Yueqi Duan, Yongpan Liu, and Hehe Fan. Seformer: Structure embedding transformer for 3d object detection. *arXiv preprint arXiv:2209.01745*, 2022. 2

[17] David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007. 5

[18] Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9701–9707. IEEE, 2020. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[20] Todd Hester and Peter Stone. Texplore: real-time sample-efficient reinforcement learning for robots. *Machine learning*, 90(3):385–429, 2013. 2

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[22] Meihuizi Jia, Lei Shen, Xin Shen, Lejian Liao, Meng Chen, Xiaodong He, Zhendong Chen, and Jiaqi Li. Mner-qg: An end-to-end mrc framework for multimodal named entity recognition with query grounding. *arXiv preprint arXiv:2211.14739*, 2022. 2

[23] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016. 6

[24] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838, 2022. 2

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[26] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 2, 5, 6, 7

[27] Bar Mayo, Tamir Hazan, and Ayellet Tal. Visual navigation with spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16898–16907, 2021. 2

[28] Lina Mezghan, Sainbayar Sukhbaatar, Thibaut Lavril, Oleksandr Maksymets, Dhruv Batra, Piotr Bojanowski, and Karteek Alahari. Memory-augmented reinforcement learning for image-goal navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3316–3323. IEEE, 2022. 2

[29] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016. 2

[30] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016. 5

[31] Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8846–8852. IEEE, 2019. 2

[32] Giuseppe Oriolo, Marilena Vendittelli, and Giovanni Ulivi. On-line map building and navigation for autonomous mobile robots. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 3, pages 2900–2906. IEEE, 1995. 2

[33] Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. In *International Conference on Learning Representations*, 2018. 2

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[35] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2, 6

[36] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022. 2

[37] Pravesh Ranchod, Benjamin Rosman, and George Konidaris. Nonparametric bayesian reward segmentation for skill discovery using inverse reinforcement learning. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 471–477. IEEE, 2015. 2

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 3

[39] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018. 2

[40] Gabriel Sepulveda, Juan Carlos Niebles, and Alvaro Soto. A deep learning based behavioral approach to indoor autonomous navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4646–4653. IEEE, 2018. 2

[41] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018. 3, 6

[42] Adrian Šošić, Abdelhak M Zoubir, and Heinz Koeppl. A bayesian approach to policy recognition and state representation learning. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1295–1308, 2017. 2

[43] Adrian Šošic, Abdelhak M Zoubir, Elmar Rueckert, Jan Peters, and Heinz Koeppl. Inverse reinforcement learning via nonparametric spatio-temporal subgoal modeling. *Journal of Machine Learning Research*, 19(69):1–45, 2018. 2

[44] Tianqi Tang, Heming Du, Xin Yu, and Yi Yang. Monocular camera-based point-goal navigation by learning depth channel and cross-modality pyramid fusion. 2022. 2

[45] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019. 2

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[47] Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. Less is more: Generating grounded navigation instructions from landmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15428–15438, 2022. 2

[48] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019. 2

[49] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6750–6759, 2019. 1, 2, 3, 5, 6

[50] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018. 1, 2, 6

[51] Xin Yu and Fatih Porikli. Ultra-resolving face images by discriminative generative networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 318–333. Springer, 2016. 2

[52] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy:

Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018. 2

[53] Sixian Zhang, Xinhang Song, Yubing Bai, Weijie Li, Yakui Chu, and Shuqiang Jiang. Hierarchical object-to-zone graph for object navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15130–15140, 2021. 2, 3, 5, 6

[54] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017. 2