Layout-based Causal Inference for Object Navigation

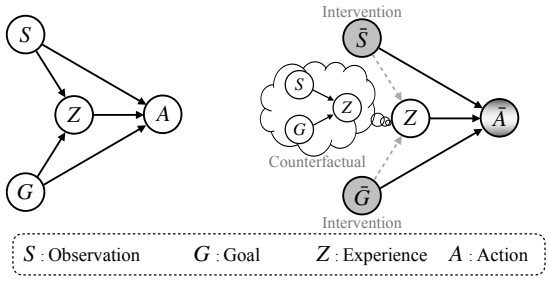
Sixian Zhang[1,2], Xinhang Song[1,2], Weijie Li[1,2], Yubing Bai[1,2], Xinyao Yu[1,2], Shuqiang Jiang[1,2]
[1]Key Lab of Intelligent Information Processing Laboratory of the Chinese Academy of Sciences (CAS),
Institute of Computing Technology, Beijing [2]University of Chinese Academy of Sciences, Beijing
{sixian.zhang, xinhang.song, weijie.li, yubing.bai, xinyao.yu}@vipl.ict.ac.cn
sqjiang@ict.ac.cn

Abstract

Previous works for ObjectNav task attempt to learn the association (e.g. relation graph) between the visual inputs and the goal during training. Such association contains the prior knowledge of navigating in training environments, which is denoted as the experience. The experience performs a positive effect on helping the agent infer the likely location of the goal when the layout gap between the unseen environments of the test and the prior knowledge obtained in training is minor. However, when the layout gap is significant, the experience exerts a negative effect on navigation. Motivated by keeping the positive effect and removing the negative effect of the experience, we propose the layout-based soft Total Direct Effect (L-sTDE) framework based on the causal inference to adjust the prediction of the navigation policy. In particular, we propose to calculate the layout gap which is defined as the KL divergence between the posterior and the prior distribution of the object layout. Then the sTDE is proposed to appropriately control the effect of the experience based on the layout gap. Experimental results on AI2THOR, RoboTHOR, and Habitat demonstrate the effectiveness of our method. The code is available at https://github.com/sx-zhang/Layout-based-sTDE.git.

1. Introduction

The visual object-oriented navigation task (i.e. ObjectNav) [3] requires the agent to navigate to a user-specified goal (e.g. laptop) based on the egocentric visual observations. A typical challenge is navigating in unseen environments, where the goal is invisible most of the time, i.e. the partial observable problem, which typically results in the agent's meaningless actions (e.g. back-tracking or getting lost at dead-ends). Although encouraging the exploration in the unseen environment (until the goal is visible) is an intuitive solution, the lack of environment layout information still limits the efficiency of goal-oriented navigation.



(a) The factual prediction (b) The counterfactual prediction

Figure 1. The proposed causal graph. (a) represents the fact prediction a, i.e. the original prediction of the trained model. (b) refers to the counter-fact prediction $\bar{a}$, i.e. the prediction is only affected by the experience Z. (b) is realized by applying the intervention and counterfactual operations to the original model.

Recently, the learning-based methods attempt to model the prior knowledge of the spatial relationships among the objects, so the agent could infer the likely locations of the goal based on the current observation (which objects are observed currently) and the prior knowledge (the spatial relationships between the goal and the observed objects) learned in the training stage. Some works utilize additional modules to construct the objects graph [15, 59, 60], the region graph [63] and the attention mechanism [32], while others [16, 56] employ a network that implicitly learns the spatial relationships end-to-end. All these methods attempt to establish prior knowledge in training environments, so that the agent would utilize the prior knowledge to associate the real-time observations with the goal, and infer the likely locations of the goal during the inference. The underlying assumption of these methods is that all of the object layouts in unseen environments should be exactly consistent with those in training environments. However, the layout consistency assumption is only partially correct due to the limited training data. Thus, those methods typically suffer from poor generalization [31] in unseen environments.

To reveal the causes of poor generalization, we propose to use the casual graph (i.e. Structural Causal Model, SCM

[38]) to analyze these navigation works. As illustrated in Fig. 1 (a), the navigation model takes the observation S and the goal G as the input, and predicts the action A at each timestamp. The causal links $S \rightarrow Z$ and $G \rightarrow Z$ represent that the observation and the goal are embedded by the pre-constructed modules [15, 32, 59, 60, 63] or the pre-trained network [16, 56]. The embedding vector is defined as the experience Z in the causal graph, which introduces the prior knowledge to influence action prediction ($Z \rightarrow A$). Meanwhile, the real-time observation and the goal also independently affect the prediction without being encoded by the prior knowledge module, which is represented as $S \rightarrow A$ and $G \rightarrow A$, respectively. The causal links $S \rightarrow A$ and $G \rightarrow A$ represent the exploration-based effect (only related to the current episode) on the action prediction, which is different from the experience-based effect $Z \rightarrow A$. Consider two cases of the layout gap between the current environment and the prior knowledge: 1) the layout gap is minor and 2) the layout gap is significant. In the former case, the object layout is consistent in the current environment and the prior knowledge. Thus, the experience Z exerts a positive effect on the prediction of action A. However, the effect of experience Z in the latter case could be negative. If the agent still relies on the "negative" experience to predict actions, it will suffer from poor generalization. Therefore, wisely utilizing the experience is essential to the ObjectNav task.

Motivated by wisely utilizing the learned experience, we propose the soft Total Direct Effect (sTDE) framework based on the Total Direct Effect analysis in causal inference. Our sTDE improves the generalization of the trained model in inference by eliminating the prediction bias brought by the experience. To decouple the effect of experience, we construct the counter-fact prediction $\bar{a}$: the prediction is only affected by the experience Z while ignoring the S and G, as shown in Fig. 1 (b). Then we propose the object layout estimator that calculates whether the effect of the experience is positive, by measuring the layout gap between the current environment and the prior knowledge. Furthermore, our sTDE will remove the counter-fact prediction $\bar{a}$ from the fact prediction a when the layout gap is large.

In this paper, we propose the layout-based soft TDE framework for the ObjectNav task. Specifically, we adopt the Dirichlet-Multinomial distribution [22] to formulate the contextual relationship between objects, which represents the object layout of the environment. Before training, the agent learns prior layout distribution (i.e. the prior parameters of Dirichlet-Multinomial distribution) by randomly exploring the training environments. In the training stage, based on the Bayesian inference, the agent estimates the posterior layout distribution with the prior distribution and newly obtained observations. Then the constantly updated posterior layout is encoded into the navigation model and utilized to learn the environment-adaptive experience. The entire model is trained with RL by maximizing the reward of reaching the goal. In the test stage, our agent will not directly use the trained policy as most previous works do. The agent first calculates the layout gap and the counter-fact prediction. The layout gap is determined by calculating the KL divergence between the posterior and prior distribution of object layouts and serves as a weight to determine whether to remove the counter-fact prediction (i.e. experience effect) from the original prediction. The experimental results on AI2THOR [27], RoboTHOR [12] and Habitat [48] indicate that our layout-based sTDE (L-sTDE) can be a plug-and-play method to boost existing methods to achieve better navigation performances.

2. Related Work

Visual Object Navigation. Target-oriented visual navigation can be categorized [5] into PointGoal [8, 18, 43, 55], AreaGoal [28, 57], AudioGoal [10, 11], ImageGoal [19, 29, 48], LanguageGoal (vision-and-language navigation) [23, 40, 41] and ObjectGoal. We focus on visual ObjectGoal navigation, which sets the object category as the target. Previous map-based methods usually build a metric map [6–8, 44] or topological map [9, 47] to memorize the environment layout. Besides, a popular deep-learning pipeline is to take an end-to-end network [2, 17, 34] to process the visual embedding and goal embedding, then predict the navigation action. Based on that, recent learning-based methods propose various improvement ideas, such as semantic representation [35], prior knowledge [59], object relation graph [15, 60, 63], attention mechanism [32], meta-learning [56], and transformer structure [16]. These well-trained modules provide prior knowledge to guide the agent in unseen environments. In our work, we denote such knowledge learned from training (with whatever improvement designs) as the experience. We try to find out, under different environment layouts, whether the learned experience takes a positive effect or not, and how to better utilize the experience to improve the navigation.

Causal Inference. Causal inference [38, 39] provides analysis tools (e.g. intervention, counter-fact) to reveal the causal effect behind the data-based statistical correlation, which is widely adopted in epidemiology, psychology, or politics research [24, 30, 46]. Recently, causal inference is introduced into the computer vision society for scene graph generation [50], vision and language task [1,36], long-tailed classification [49] and few/zero-shot learning [4, 62]. For the visual navigation task, [37] generates counterfactual observations to improve generalization to new environments for VLN (vision-and-language navigation) task. [54] proposes causal CT (continuous-time) models for visual drone navigation tasks. In our work, we focus on the ObjectNav task and utilize the causal inference to wisely control the effect of experience in the inference.

3. Preliminaries of ObjectNav Task

Task Definition. The ObjectNav task requires the agent to navigate to a user-specified object category from a random starting location in an unseen scene, and the agent is only allowed to utilize the egocentric visual observation (e.g. RGB images) and the semantic embedding of the target object. Formally, at each timestamp t, the agent receives the observation $S = s^t$ and the goal $G = g^t$ and is expected to adopt an action $A = a^t$, $a^t \in \mathcal{A}$. The discrete action space is defined as: $\mathcal{A} = \{MoveAhead, RotateLeft, RotateRight, LookDown, LookUp, Done\}$. When the agent outputs the *Done* action, the episode terminates. An episode is considered **successful** if, within a certain number of steps, the final location of the agent is close to the goal within a distance threshold (e.g. $1m$) and the goal is visible in the egocentric observation.

Navigation Model. Prevailing ObjectNav works can be summarized as a visual-goal encoder and a policy function, as shown in Fig. 2. The visual-goal encoder is constructed to associate the visual observation S and the goal G. We denote such association as the experience Z since the encoder could learn the prior experience during the training.

Previous learning-based methods typically implement the visual-goal encoder as an object relation graph [15, 59, 60], region graph [63], attention module [32] and transformer structure [16]. For instance, the ORG [15] builds an object relation graph, where the nodes are defined as the object categories and the edges are defined as the spatial correlations among different categories. The object relation graph takes the observation and the goal as inputs and encodes them with the learned edges to obtain the experience Z. Previous map-based methods [7, 44] typically construct the semantic map based on the observations, and embed the map with the target. Such embedding is also regarded as the experience Z. Based on the observation, target object, and these learned experiences, the policy module will output the prediction. These navigation models are typically trained with the RL (reinforcement learning) [33, 55, 64].

4. The Proposed Solution

4.1. Layout-based Soft Total Direct Effect

We humans make decisions in a causality-based manner: we have the instinct to assess the effect of the navigation experience [51] and wisely eliminate the cognition bias from the experience. However, the machines are likelihood-based [50] and the predictions depend on the association likelihood between observations, goal, and actions, which is mainly learned from limited training data. Therefore, the predictions are inevitably biased due to the biased experience. Prevailing methods [15, 16, 56, 59, 60, 63] entirely accept the experience learned in the training stage, and rarely consider whether the prior experience is beneficial to the

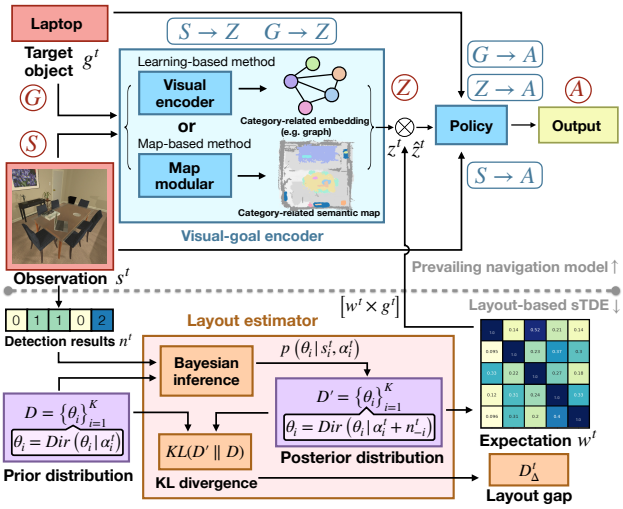


Figure 2. Prevailing navigation framework (top) and our contributions (bottom). Our layout estimator outputs the expectation w^t of posterior layout distribution and the layout gap D_Δ^t. The w^t is proposed to enhance the adaptability of the experience z^t to new environments. The layout gap D_Δ^t serves as a weight to alleviate the experience bias.

current environment. However, as previously discussed, the learned experience has both positive and negative effects, especially in the unseen environments, which may have a different layout from the training environments. Therefore, keeping positive effects and removing negative effects of learned experience is essential to the navigation models.

The causal inference [39, 52, 53] encourages the machine to discover the causality behind the association likelihood, which has been proven to be an effective analysis method in many computer vision tasks [36, 49, 50, 61]. From the perspective of causal inference, the experience Z serves as the mediator between the input (observation S and target G) and the output (action A) as illustrated in Fig. 1 (a). To eliminate the *mediation fallacy*, the causal inference [38] introduces the Total Direct Effect (TDE) analysis to remove the prediction bias brought by the mediator.

Total Direct Effect. The fundamental principle behind TDE analysis [36, 49, 50] is to build a bias prediction $\bar{a}^t$ that is exclusively affected by the mediator, and then remove the bias item from the original prediction a^t. The TDE analysis is formulated as:

$$TDE(a^t) = a^t - \bar{a}^t \tag{1}$$

where $a^t = \pi(A|S = s^t, G = g^t)$ is the factual prediction of a trained model π, which takes the observation $S = s^t$ and target $G = g^t$ as the input, as shown in Fig. 1 (a). The latter is the counterfactual prediction $\bar{a}^t = \pi(A|do(S = \bar{s}^t, G = \bar{g}^t), Z = Z_{s^t,g^t})$ as shown in Fig. 1 (b). The calculation of $\bar{a}^t$ includes two operations

in causal inference: intervention and counterfactual operations. The intervention operation is denoted as $do(\cdot)$ and realized by specifying a certain value for the variable (e.g. $do(S = \bar{s}^t)$). The intervened value (e.g. $\bar{s}^t$, $\bar{g}^t$) is set to a random or zero vector. The intervention operation eliminates the effect of observation S and goal G on the action prediction, while it also changes the value of the causal successor (i.e. Z). However, since our motivation is to model the case only affected by the mediator, which requires the mediator Z to remain the original value. To this end, the counterfactual operation assigns the original value to the variable Z ***as if*** $S = s^t, G = g^t$ had existed. The counterfactual operation is symbolized as $Z = Z_{s^t, g^t}$.

The TDE analysis is easily realized by manipulating the values of several variables (e.g. S, G, Z) and is efficient in removing the effect of the mediator (i.e. experience). However, the experience is not always detrimental. It can also be beneficial when the object layout gap is minor. Therefore, we propose the soft TDE to appropriately maintain the positive effect of experience.

Soft Total Direct Effect. In alternative to previous TDE analysis, our soft TDE modifies the Eq. 1 by adding a trade-off weight, which is formulated as:

$$sTDE(a^t) = a^t - ReLU(D_\Delta^t - \varepsilon) \cdot \bar{a}^t \tag{2}$$

where $ReLU(\cdot)$ is the ReLU activation function, and ε is a threshold hyperparameter. The D_Δ^t is the object layout gap, which is produced by our layout estimator (Sec. 4.2) as shown in Fig. 2. The D_Δ^t evaluates the effect of the experience Z, and preserves the positive experience when the layout gap between the current environment and prior knowledge is minor (i.e. D_Δ^t is less than the threshold ε).

4.2. Layout Estimator

Object Layout. For the ObjectNav task, the main challenge is the partially observable problem, that the agent can only observe the partial area at each time. With continuously observing during the navigation process, more objects will be observed, and the partially known problem can also be alleviated. Considering objects are usually distributed in a reasonable layout, the adjacent relation of objects is essential to help the agent infer the location of the target object based on the observed objects. Therefore, we concentrate on the contextual relationship to represent the object layout. The object layout in a scene is defined as the set $D = \{\theta_i\}_{i=1}^K$, where K is the number of object categories. The $\theta_i = (\theta_{i,j})_{j=1}^K$ represents the context distribution of object category c_i. Each dimension $\theta_{i,j}$ represents the probability $p(c_j|c_i)$ that the object category c_j can be observed around the category c_i in an observation. Specifically, we assume that the context distribution follows the Dirichlet distribution: $\theta_i = (\theta_{i,j})_{j=1}^K \sim Dir(\alpha_i)$, where

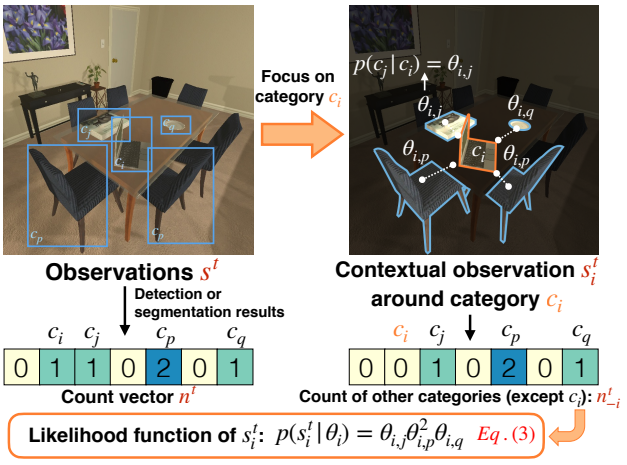


Figure 3. The calculation of the likelihood function $p(s_i^t|\theta_i)$.

$\sum_j \theta_{i,j} = 1, \theta_{i,j} \geq 0$, and $\alpha_i = (\alpha_{i,j})_{j=1}^K, \alpha_{i,j} > 0$ parameterizes the distribution. The Dirichlet distribution is chosen for two reasons: 1) the conjugate prior permits concise calculations; 2) the likelihood-equivalence property [22].

At the beginning of every new episode, the parameter α_i is initialized as the prior initial value α_i^* (α_i^* will be detailed in Sec. 4.3). At each timestamp t, our method computes the posterior object distribution based on the incoming observations. Specifically, the agent receives the visual observation s^t, and the visual observation is then converted into a count vector $n^t \in \mathbb{R}^K$, where each dimension of $n^t = (n_i^t)_{i=1}^K$, $n_i^t \geq 0$ records the number of each object category appearing in the current observation. The count vector n^t is obtained by the object detection or segmentation modules (e.g. Faster R-CNN [45] or Mask R-CNN [20]). As shown in Fig. 3, we define $n_{-i}^t = \left(n_{-i,j}^t\right)_{j=1}^K$, $n_{-i,j}^t \geq 0$ as the count of other categories (except c_i) and define s_i^t as the contextual observation of the category c_i. Conditioned on c_i, assume the probability that s_i^t is observed follows the multinomial distribution $s_i^t \sim Mult\left(n_{-i}^t, \theta_i\right)$. Then the likelihood function can be calculated as:

$$p\left(s_i^t|\theta_i\right) = \prod_{j=1}^K \theta_{i,j}^{n_{-i,j}^t} \tag{3}$$

Meanwhile, according to the previous Dirichlet distribution assumption, the prior distribution of θ_i at timestamp t is $p\left(\theta_i|\alpha_i^t\right) = Dir\left(\theta_i|\alpha_i^t\right)$. Considering the contextual observation s_i^t, the posterior distribution of θ_i is further calculated according to the Bayesian inference as the following:

$$p\left(\theta_i|s_i^t, \alpha_i^t\right) = \frac{p\left(s_i^t|\theta_i\right) p\left(\theta_i|\alpha_i^t\right)}{p\left(s_i^t|\alpha_i^t\right)} = Dir\left(\theta_i|\alpha_i^t + n_{-i}^t\right) \tag{4}$$

The Eq. 4 indicates that the posterior distribution $p\left(\theta_i|s_i^t, \alpha_i^t\right)$ also satisfies the Dirichlet distribution (the calculation process is detailed in the supplements), which is known as the conjugate prior property. The conjugate prior property guarantees the succinct calculation for the posterior distribution of θ_i, i.e. simply combining the count vector n_{-i}^t of context objects around category c_i with the hyperparameter α_i^t of the prior distribution. Similarly, the posterior distributions of other object categories can also be obtained following the Eq. 4.

Furthermore, the posterior distributions at time t serves as the prior distributions at time $t+1$, thus the value of hyperparameter α_i is updated by $\alpha_i^{t+1} = \alpha_i^t + n_{-i}^t$. This update guarantees the knowledge of object layout is continuously adapted to the current navigation environment.

Based on the obtained posterior distribution, the layout gap D_Δ^t is defined as the KL divergence between the posterior and prior object layout over all categories. Formally, the object layout gap D_Δ^t is defined as:

$$D_\Delta^t = tanh(\frac{1}{K}\sum_{i=1}^{K} KL\left(p\left(\theta_i|s^t, \alpha_i^t\right)||p\left(\theta_i|\alpha_i^t\right)\right)) \quad (5)$$

where $tanh(\cdot)$ is the tanh activation function. The layout gap D_Δ^t estimates the gap between the posterior object layout at time t and the prior knowledge before time t, which plays a vital role in the Eq. 2.

Environment-Adaptive Experience. The prevailing methods [15, 16, 59] typically encode the visual observation s^t and the target object g^t into the experience $z^t \in \mathbb{R}^{K\times M}$ during training, where K is the number of object categories and M refers to the feature dimension. The learned experience could be both biased (the layout gap between the training and test environments) and environment-static (without adaptation to the current environment). Therefore, in addition to eliminate the bias in the inference, we also propose to enhance the environmental adaptability of the experience z^t to achieve a better performance. To this end, the dynamically updated knowledge of object layout $D = \{\theta_i\}_{i=1}^K$ is encoded into the navigation framework. Specifically, the expectation of the posterior distribution of the object layout D is formalized as $w^t \in \mathbb{R}^{K\times K}$, $w_{i,j}^t > 0$. For $j = i$, since $p(c_i|c_i)$ always exists, we set $w_{i,j}^t = 1$. When $j \neq i$, $w_{i,j}^t = \mathbb{E}_{p(\theta_i|s_i^t,\alpha_i^t)}(\theta_{i,j})$, where $\mathbb{E}(\cdot)$ denote the expectation calculation. Based on [22], the expectation of the Dirichlet function $Dir(\theta_i|\alpha_i^t + n_{-i}^t)$ is equivalent to the normalization of its parameters. Accordingly, for $j \neq i$,

$$w_{i,j}^t = \mathbb{E}_{p\left(\theta_i|s_i^t,\alpha_i^t\right)}\left(\theta_{i,j}\right) = \frac{\alpha_{i,j}^t + n_{-i,j}^t}{\sum_{j=1}^K(\alpha_{i,j}^t + n_{-i,j}^t)} \quad (6)$$

Since w^t is updated in real-time, we propose to improve the environment-static experience z^t to environment-adaptive experience $\hat{z}^t$ by incorporating the dynamically updated knowledge of object layout, where $\hat{z}^t$ is given by:

$$\hat{z}^t = z^t \odot \left[w^t \times g^t\right] \quad (7)$$

where $\odot$ represents the element-wise multiplication, $\times$ refers to the matrix multiplication, and $g^t \in \mathbb{R}^K$ is a one-hot vector indicating the target category of the goal. The result of $[w^t \times g^t] \in \mathbb{R}^K$ indicates the probability of the target g^t is observed conditioned on each object category, which serves as a weight on the category-specific dimension of z^t.

4.3. Training and Inference

As shown in Fig. 2, our layout estimator outputs (1) the expectation w^t of the posterior layout distribution, and (2) the layout gap D_Δ^t. Both w^t and D_Δ^t are calculated mathematically without learnable parameters, thus our method has the plug-and-play property to boost the existing methods. Specifically, our layout-based sTDE is realized as: **(1) Before training**, to determine the initial value $(\alpha_i^*)_{i=1}^K$, we first employ a random agent equipped with our object layout estimator, whose parameters $(\alpha_i)_{i=1}^K$ are initialized as all one vector. Then the agent randomly explores the training scenes to obtain sufficient observations to update the parameters by Eq. 4. The updated parameters are defined as $(\alpha_i^*)_{i=1}^K$. **(2) In the training stage**, at the beginning of every episode, all parameters $(\alpha_i)_{i=1}^K$ are initialized as the prior parameters $(\alpha_i^*)_{i=1}^K$. The layout estimator calculates the posterior distribution of object layout D (Eq. 4), using the constant observation s^t and the prior distribution. Then the expectation of the posterior distribution w^t is estimated and incorporated in the navigation model π (Eq. 6, 7). The model π is trained with RL. **(3) In the causal-inference stage**, the agent will not directly use the trained policy π, while it first calculates the layout gap D_Δ^t (Eq. 5) and the counter-fact prediction $\bar{a}^t$ (the effect of the experience). Then the agent appropriately removes the counter-fact prediction $\bar{a}^t$ from the fact prediction a^t based on the layout gap D_Δ^t (Eq. 2). For more details and algorithms of training and inference, please refer to the supplements.

Note that the expectation of the posterior distribution of the object layout w^t affects the prediction of the model during both training and inference, while D_Δ^t is utilized for debiasing only in the inference. Furthermore, we want to explain **why the debiasing operation only acts in inference**. As shown in Fig. 1, the prediction is influenced by two effects: the inputs (the observation S and the goal G) and their mediator (the experience E). The mediator (experience) has agnostic effects because it is learned from limited training data. The training data can not cover all room layouts, thus resulting the prediction bias in the inference. Consequently, the debiasing operation (i.e. the sTDE partially removes the effect of experience from prediction) is conducted in inference to decouple the effect from the training data.

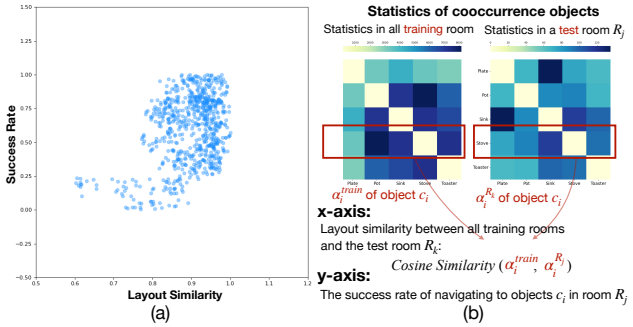


Figure 4. The correlation between the navigation performance on success rate (SR) and the layout similarity.

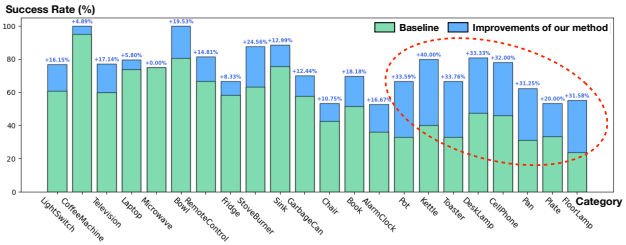


Figure 5. Category-wise improvement of our method compared to the baseline. The categories are arranged from high to low based on their average layout similarity between training and testing scenes.

5. Experiments

5.1. Experimental Setup

We utilize the AI2THOR [27], RoboTHOR [12] simulators to conduct ablation studies and compare our method with the learning-based methods [15, 16, 32, 56, 59, 63, 64]. The experimental settings on above datasets follow the primary works [15,16,63]. Besides, we choose the Gibson [58] dataset in the Habitat simulator [48] to compare our method with the map-based methods and adopt the same settings as [7]. For more details about the settings of the datasets (e.g. the split of training, validation, testing scenes, and the target objects), please refer to the supplements.

Our method and the related works are both trained to maximize the navigation reward via RL. The reward function penalizes the agent by -0.01 for each step and rewards it by 5 if the episode is successful (the successful episode is defined in Sec. 3). All methods are trained with 6M episodes, where in each episode, the starting position of the agent and the goal are randomly selected. Note that, since our method needs prerequisite exploration to determine the initial parameters $(\alpha_i^*)_{i=1}^K$ of the layout distribution. To be fair, we employ the agent to randomly explore the training rooms to learn the initial parameters for 2M episodes, then we train our method with 4M episodes.

For evaluation, we repeatedly run 3 trials and report the results with mean±standard deviation. We choose SR (Success Rate), SPL (Success weighted by Path Length), and DTS (Distance to Goal) metrics for evaluation, which are detailed in the supplements. For all results in the following tables, the ↑ means that the larger value is better, while ↓ indicates the opposite.

5.2. The Compared Methods

We choose the learning-based methods [15, 16, 32, 56, 59, 63, 64] and the map-based method [7] for comparison. **Random**: the agent adopts random actions. **RGB(D)+RL (A3C)** [48, 64]: the agent utilizes the simple embeddings of visual input and the goal, and is trained with A3C [34]. **SP** [59]: SP constructs the scene prior knowledge from the external dataset. The prior knowledge encoded with GCN [26] is provided to the agent as the additional information. **SAVN** [56]: SAVN establishes a sub-network to learn a self-supervised interaction loss. **EOTP** [32]: EOTP proposes an attention module to encode the semantic information about the observed objects. **ORG** [15]: ORG proposes to learn an object relation graph, and builds a memory-augmented tentative policy network (TPN) to produce self-supervision. **VTNet** [16]: VTNet employs the Transformer network to encode the visual information and the goal. **HOZ** [63]: HOZ attempts to abstract the region relation among the objects and proposes to learn the region relation graph to associate the vision and the goal. **SemExp** [7]: SemExp builds an episodic semantic map and trains a goal-oriented policy taking the semantic map as the input.

5.3. Evaluation Results

In this section, we adopt the ORG [15] as the baseline and conduct evaluations on the validation set of AI2THOR.

Correlation between layout and success rate. Our motivation is that the layout gap between training and testing scenes may have an impact on navigation. To verify this, we take the statistic and visualize the results in Fig. 4 (a), where the y-axis represents the success rate of navigating to objects c_i in a testing scene R_j, the x-axis represents the layout similarity of the object c_i between the training scenes and the test scenes, and the scatter points represent all target objects in all test scenes. As shown in Fig. 4 (b), the layout similarity is defined as the cosine similarity of the statistics vectors (the count of cooccurrence objects) obtained in the training scenes and the test scene R_j. Fig. 4 (a) indicates that the navigation performance on SR during inference has a generally positive correlation with the layout similarity, which supports our motivation.

Improvements of our method compared to baseline. The navigation performance of the baseline (green bar) and the improvements brought by our method (blue bar) are shown in Fig. 5. All goals (x-axis) are arranged from high

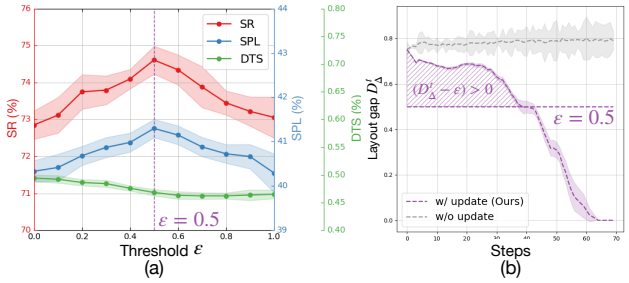


Figure 6. The ablation study of the threshold ε and the visualization of the object layout gap D^t_Δ over the navigation steps.

to low based on their layout similarity. The results indicate that our method can bring more obvious improvement in the categories with low layout similarity (as illustrated by the red dashed circle). We presume that the goals with a larger layout gap are more susceptible to the effects of the experience, thus benefiting more from our de-biased method. The results once again support our motivation.

Ablations on the threshold ε. The threshold ε influences the agent by Eq. 2, which determines whether to remove the prediction bias introduced by the experience. As shown in Fig. 6 (a), the results of different threshold ε basically show the same trend on the SR, SPL metrics: as the value of threshold ε increases, the performances on these metrics initially increase and then decrease. When the value of threshold ε is small, the Eq. 2 tends to consider that the majority of experiences Z are negative and need to be removed. At this point, gradually increasing the threshold ε could retain more experiences and improve performance. However, when the threshold ε is too large (e.g. $\varepsilon = 1$), the Eq. 2 falls into another extreme which considers all experiences are positive and need to be preserved, resulting in the performance degradation. The results on DTS metric indicate that excessively removing the experience (ε is small) will increase the exploration of the agent, thus resulting in worse performance on DTS. Based on the above results, we choose the threshold as $\varepsilon = 0.5$.

Visualization of the object layout gap D^t_Δ. The object layout gap D^t_Δ represents the KL divergence between the posterior and the prior distribution of the layout. The average value of the D^t_Δ over navigation steps in AI2THOR is illustrated in Fig. 6 (b). The results indicate that the layout gap decreases as the number of steps increases (the purple line). However, without updating by our method (the grey line), the layout gap is maintained at a large value. The trend of D^t_Δ demonstrates that the learned knowledge (posterior distribution of the layout) by our method is gradually approaching the real layout of the current scene. It also reveals that the w^t (the expectation of posterior distribution) is environment-adaptive information for guiding the agent.

Additionally, considering the threshold $\varepsilon = 0.5$ and looking back at Fig. 6 (b), we can observe how our layout-based sTDE works in the time (i.e. steps) dimension. According to the Eq. 2, the de-bias operation is activated when $(D^t_\Delta - \varepsilon) > 0$, as shown in the upper purple shadow. At the beginning of an episode, the agent is unfamiliar with the current scene, and the layout gap is large. At this stage, our layout-based sTDE is activated to eliminate the prediction bias caused by the experience. Then with continuous navigating and updating, the prior knowledge of the object layout has been gradually adjusted to fit the current scene. Therefore, at this stage, the updated experience is applicable to the current scene and fully accepted for the navigation.

Table 1. **The ablation studies.** (I) The ablations of several components. The w^t represents encoding the expectation of the posterior distribution of object layout into the model, and D^t_Δ represents debiasing the trained model in the inference with our sTDE. (II) The comparison of different de-bias methods.

ID		Method: Baseline	w^t	D^t_Δ	SR↑ (%)	SPL↑ (%)	DTS↓ (m)
I	1	✓			66.48±0.22	38.42±0.39	0.54±0.01
	2	✓	✓		71.40±0.37	39.11±0.44	0.49±0.02
	3	✓	✓	✓	**74.95**±0.38	**41.68**±0.23	**0.47**±0.02
II	4	w/o De-bias			71.40±0.37	39.11±0.44	0.49±0.02
	5	TDE [53]			69.24±0.12	38.38±0.16	0.52±0.01
	6	*Ours* **sTDE** (zero)			74.95±0.38	**41.68**±0.23	0.47±0.02
	7	*Ours* **sTDE** (random)			**75.03**±0.69	41.48±0.44	**0.46**±0.03

Ablations of different components. We evaluate different components of our method. The ablation study in Tab. 1 (I) indicates the efficacy of each component in our method. Specifically, the expectation w^t of the posterior distribution of the object layout improves the baseline (ORG) performance by 4.92%, 0.69% and -0.05m on SR, SPL and DTS metrics. Moreover, the soft Total Direct Effect (sTDE) further improves the performance and gains 3.55%, 2.57% and -0.02m improvement on SR, SPL and DTS metrics. Overall, our method outperforms the baseline by 8.47%, 3.26% and -0.07m on SR, SPL and DTS metrics.

Comparison with TDE and sTDE. The TDE analysis is effective in removing the mediation fallacy and is applied in many works [36, 49, 50], which regard the mediator (i.e. the experience in our work) as the cause of prediction bias and remove it in the inference by Eq. 1. However, directly applying TDE makes the agent completely ignore the positive effect of the experience. Thus, as shown in Tab. 1 (II), employing TDE (line 5) even degrades the navigation performance (compared with line 4). Compared with TDE, our sTDE (line 6 and 7) adaptively retains the positive effects of the experience and obtains significant improvements. Previous work [50] introduces constructing the counterfactual item (i.e. $\bar{a}^t$), by setting the value of observation S and the goal G to random or zero. We respectively adopt these two ways (line 6 and 7) and find out that their performances

Table 2. The comparisons with the related works (learning-based methods) in AI2THOR and RoboTHOR simulators. The L-sTDE represents our layout-based soft TDE framework based on different baselines. The improvements are shown in blue font.

ID	Method	AI2THOR			RoboTHOR		
		SR↑ (%)	SPL↑ (%)	DTS↓ (*m*)	SR↑ (%)	SPL↑ (%)	DTS↓ (*m*)
1	Random	4.68 ±1.74	2.42 ±1.35	1.36 ±0.01	2.92 ±0.39	1.34 ±0.20	2.40 ±0.02
2	RGB+RL (A3C) [64]	59.06 ±0.19	34.56 ±0.44	0.68 ±0.01	27.48 ±0.46	16.45 ±0.12	2.08 ±0.02
3	SP [59]	62.19 ±0.67	37.60 ±0.35	0.61 ±0.02	26.22 ±0.78	16.90 ±0.33	2.06 ±0.02
4	SAVN [56]	63.27 ±0.11	38.20 ±0.04	0.56 ±0.01	28.53 ±0.77	18.27 ±0.35	1.96 ±0.02
5	EOTP [32]	65.61 ±0.25	38.93 ±0.10	0.55 ±0.01	28.84 ±0.41	18.82 ±0.35	1.95 ±0.02
6	ORG [15]	66.53 ±0.29	39.00 ±0.34	0.54 ±0.01	29.64 ±0.89	19.13 ±0.59	1.95 ±0.03
7	ORG+TPN [15]	68.60 ±0.29	39.40 ±0.17	0.54 ±0.01	30.05 ±0.06	19.05 ±0.08	1.89 ±0.01
8	VTNet [16]	70.10 ±1.00	39.60 ±0.10	0.52 ±0.01	31.62 ±0.74	19.63 ±0.42	1.87 ±0.03
9	HOZ [63]	70.38 ±0.14	39.04 ±0.11	0.48 ±0.02	32.27 ±1.14	20.48 ±0.48	1.85 ±0.01
10	*Ours* **L-sTDE** (EOTP)	**69.62** (4.01↑) ±0.71	**40.02** (1.09↑) ±0.47	**0.52** (0.03↑) ±0.01	**37.37** (8.53↑) ±0.75	**22.00** (3.18↑) ±0.42	**1.81** (0.14↑) ±0.02
11	*Ours* **L-sTDE** (ORG)	**74.85** (8.32↑) ±0.33	**41.56** (2.56↑) ±0.11	**0.47** (0.07↑) ±0.02	**39.27** (9.63↑) ±0.55	**22.81** (3.68↑) ±0.47	**1.77** (0.18↑) ±0.02
12	*Ours* **L-sTDE** (VTNet)	**75.25** (5.15↑) ±0.39	**41.69** (2.09↑) ±0.13	**0.47** (0.05↑)±0.02	**41.53** (9.91↑) ±0.67	**23.87** (4.24↑) ±0.50	**1.72** (0.15↑) ±0.02
13	*Ours* **L-sTDE** (HOZ)	**75.05** (4.67↑) ±0.29	**41.53** (2.49↑) ±0.21	**0.46** (0.02↑)±0.02	**42.13** (9.86↑) ±0.78	**24.54** (4.06↑) ±0.43	**1.67** (0.18↑) ±0.01

are similar, except that using the random value makes the performance more volatile (i.e. larger standard deviation). Therefore, we set the value of the observation and the goal as zeros to construct the counterfactual item. More ablation studies and visualizations are detailed in the supplements.

5.4. Comparisons with the Related Works

In this section, we evaluate all methods on the test datasets of AI2THOR, RoboTHOR, and Gibson. We compare the learning-based methods [15, 16, 32, 56, 59, 63, 64] in AI2THOR and RoboTHOR as shown in Tab. 2. With the egocentric RGB input, all methods are constructed with the Faster R-CNN backbone, which is pre-trained following [15]. To keep fair comparisons, we present VTNet [16] using their reported results with Faster R-CNN. Since other methods [32, 56, 59, 64] are originally proposed with the ResNet [21] pre-trained in ImageNet [14] as the visual backbone. For fair comparisons, following the recommendation of [15], we additionally encode the detection information provided by Faster R-CNN to these methods [32, 56, 59, 64]. Therefore, the reimplemented performances are higher than those results reported in the original papers. As shown in Tab. 2, benefiting from the incorporation of our layout-based sTDE (L-sTDE), the performances of existing methods are improved, especially our L-sTDE based on HOZ and VTNet achieve new SOTA performance. Other learning-based methods employ CLIP [42] as the backbone [25], or adopt pre-training on large-scale datasets (e.g. ProcTHOR [13]), and we compare our L-sTDE with these methods in the supplements.

Since most map-based methods require the RGB-D visual inputs and segmentation information provided by Mask R-CNN [20] to construct the semantic map, we take another comparison with the map-based methods as shown in Tab. 3. The map-based methods employ a parameter-free modular system to build the semantic map, then encode the map into the policy. We regard the encoded map as the experience z^t and modify [7] with our L-sTDE. The results indicate that our method still brings some performance improvements, although the gains are limited compared to that of the learning-based methods. We analyze that, compared with the learning-based methods whose parameters are all trained end-to-end, the map-based methods utilize a modular system, and only a small number of parameters (the policy) need to be trained. Therefore, the map-based methods are less affected by the experience learned from training, and thus benefit less from our method.

Table 3. The comparisons with the related works (map-based methods) in Gibson simulators.

ID	Method	SR↑ (%)	SPL↑ (%)	DTS↓ (*m*)
1	Random	0.04	0.04	3.89
2	RGBD+RL [48]	8.42	2.87	3.23
3	SemExp [7]	65.35	33.52	1.56
4	*Ours* **L-sTDE** (SemExp)	**66.96** (1.61↑)	**34.33** (0.81↑)	**1.53** (0.03↑)

6. Conclusion

We propose the layout-based soft Total Direct Effect framework for ObjectNav task. Our motivation is to keep the positive effect and remove the negative effect of the experience, which is learned as the prior knowledge in training. Specifically, we propose to calculate the layout gap between the current environment and the prior knowledge. Then the layout gap is utilized to assess whether the impact of the learned experience is positive. Furthermore, the soft Total Direct Effect is proposed to appropriately control the effect of the experience on action prediction. The experiment results indicate the effectiveness of our method.

Acknowledgements: This work was supported by Beijing Natural Science Foundation under Grant JQ22012, Z190020, in part by the National Natural Science Foundation of China under Grant 62125207, 62272443, 62032022 and U1936203, in part by the National Postdoctoral Program for Innovative Talents under Grant BX201700255.

References

[1] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10041–10051, 2020. 2

[2] Ziad Al-Halah, Santhosh K. Ramakrishnan, and Kristen Grauman. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 17010–17020. IEEE, 2022. 2

[3] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 1

[4] Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2

[5] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *CoRR*, abs/2006.13171, 2020. 2

[6] Tommaso Campari, Leonardo Lamanna, Paolo Traverso, Luciano Serafini, and Lamberto Ballan. Online learning of reusable abstract models for object goal navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 14850–14859. IEEE, 2022. 2

[7] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Russ R. Salakhutdinov. 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. 2, 3, 6, 8

[8] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. 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. 2

[9] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. 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*, pages 12872–12881. IEEE, 2020. 2

[10] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15516–15525, 2021. 2

[11] Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. 2

[12] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. 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*, pages 3161–3171, 2020. 2, 6

[13] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied AI using procedural generation. 2022. 8

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 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*, pages 248–255, 2009. 8

[15] Heming Du, Xin Yu, and Liang Zheng. 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*, pages 19–34, 2020. 1, 2, 3, 5, 6, 8

[16] Heming Du, Xin Yu, and Liang Zheng. Vtnet: Visual transformer network for object goal navigation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. 1, 2, 3, 5, 6, 8

[17] Kshitij Dwivedi, Gemma Roig, Aniruddha Kembhavi, and Roozbeh Mottaghi. What do navigation agents learn about their environment? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10266–10275. IEEE, 2022. 2

[18] Kuan Fang, Alexander Toshev, Fei-Fei Li, and Silvio Savarese. 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*, pages 538–547. Computer Vision Foundation / IEEE, 2019. 2

[19] Meera Hahn, Devendra Singh Chaplot, Shubham Tulsiani, Mustafa Mukadam, James M. Rehg, and Abhinav Gupta. No rl, no simulation: Learning to navigate without navigating. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 26661–26673, 2021. 2

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988, 2017. 4, 8

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 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, pages 770–778, 2016. 8

[22] David Heckerman, Dan Geiger, and David Maxwell Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, 20(3):197–243, 1995. 2, 4, 5

[23] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez Opazo, and Stephen Gould. VLN BERT: A recurrent vision-and-language BERT for navigation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1643–1653. Computer Vision Foundation / IEEE, 2021. 2

[24] Luke Keele. The statistics of causal inference: A view from political methodology. *Political Analysis*, 23(3):313–335, 2015. 2

[25] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: CLIP embeddings for embodied AI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 14809–14818. IEEE, 2022. 8

[26] Thomas N. Kipf and Max Welling. 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. 6

[27] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474, 2017. 2, 6

[28] Ashish Kumar, Saurabh Gupta, and Jitendra Malik. Learning navigation subroutines from egocentric videos. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 617–626. PMLR, 2019. 2

[29] Obin Kwon, Nuri Kim, Yunho Choi, Hwiyeon Yoo, Jeongho Park, and Songhwai Oh. Visual graph memory with unsupervised representation for visual navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15870–15879, 2021. 2

[30] David P MacKinnon, Amanda J Fairchild, and Matthew S Fritz. Mediation analysis. *Annu. Rev. Psychol.*, 58:593–614, 2007. 2

[31] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. THDA: treasure hunt data augmentation for semantic navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15354–15363, 2021. 1

[32] Bar Mayo, Tamir Hazan, and Ayellet Tal. Visual navigation with spatial attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 16898–16907, 2021. 1, 2, 3, 6, 8

[33] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andy Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. 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. 3

[34] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 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*, pages 1928–1937, 2016. 2, 6

[35] Arsalan Mousavian, Alexander Toshev, Marek Fiser, Jana Kosecká, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 8846–8852. IEEE, 2019. 2

[36] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual VQA: A cause-effect look at language bias. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12700–12710, 2021. 2, 3, 7

[37] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Qinfeng Shi, and Anton van den Hengel. Counterfactual vision-and-language navigation: Unravelling the unseen. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2

[38] Judea Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009. 2, 3

[39] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018. 2, 3

[40] Yuankai Qi, Zizheng Pan, Yicong Hong, Ming-Hsuan Yang, Anton van den Hengel, and Qi Wu. The road to know-where: An object-and-room informed sequential BERT for indoor vision-language navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 1635–1644. IEEE, 2021. 2

[41] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part X*, volume 12355 of *Lecture Notes in Computer Science*, pages 303–317. Springer, 2020. 2

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 8

[43] Santhosh K. Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, pages 400–418, 2020. 2

[44] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18868–18878. IEEE, 2022. 2, 3

[45] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 4

[46] Lorenzo Richiardi, Rino Bellocco, and Daniela Zugna. Mediation analysis in epidemiology: methods, interpretation and bias. *International journal of epidemiology*, 42(5):1511–1519, 2013. 2

[47] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. 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. 2

[48] Manolis Savva, Jitendra Malik, Devi Parikh, Dhruv Batra, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. 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*, pages 9338–9346. IEEE, 2019. 2, 6, 8

[49] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33:1513–1524, 2020. 2, 3, 7

[50] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3713–3722, 2020. 2, 3, 7

[51] Perry W Thorndyke and Barbara Hayes-Roth. Differences in spatial knowledge acquired from maps and navigation. *Cognitive psychology*, 14(4):560–589, 1982. 3

[52] Tyler VanderWeele. *Explanation in causal inference: methods for mediation and interaction*. Oxford University Press, 2015. 3

[53] Tyler J VanderWeele. A three-way decomposition of a total effect into direct, indirect, and interactive effects. *Epidemiology (Cambridge, Mass.)*, 24(2):224, 2013. 3, 7

[54] Charles Vorbach, Ramin M. Hasani, Alexander Amini, Mathias Lechner, and Daniela Rus. Causal navigation by continuous-time neural networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12425–12440, 2021. 2

[55] 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. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 2, 3

[56] 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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6750–6759, 2019. 1, 2, 3, 6, 8

[57] Yi Wu, Yuxin Wu, Aviv Tamar, Stuart J. Russell, Georgia Gkioxari, and Yuandong Tian. 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*, pages 2769–2779, 2019. 2

[58] Fei Xia, Amir Roshan Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9068–9079. IEEE Computer Society, 2018. 6

[59] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 1, 2, 3, 5, 6, 8

[60] Xin Ye and Yezhou Yang. 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*, pages 14101–14110, 2021. 1, 2, 3

[61] Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15404–15414. Computer Vision Foundation / IEEE, 2021. 3

[62] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2

[63] 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 (ICCV)*, pages 15130–15140, October 2021. 1, 2, 3, 6, 8

[64] 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 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3357–3364, 2017. 3, 6, 8