NaviFormer: A Spatio-Temporal Context-Aware Transformer for Object Navigation

Wei Xie[1] , Haobo Jiang[2], Yun Zhu[1], Jianjun Qian[1], Jin Xie[3, 4] *

[1]PCA Lab, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[2]Nanyang Technological University, Singapore
[3]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[4]School of Intelligence Science and Technology, Nanjing University, Suzhou, China
{xie.wei, zhu.yun, csjqian}@njust.edu.cn; haobo.jiang@ntu.edu.sg; csjxie@nju.edu.cn

Abstract

Learning discriminative state representations of agents, encompassing the spatial layout and temporal pose trajectory, is essential for effective navigation decisions. However, existing approaches often rely on simplistic plain networks for navigation information fusion, overlooking the complex long-range dependencies across spatio-temporal cues, which leads to suboptimal state perception and potential decision failures. In this paper, we introduce NaviFormer, an effective encoder-decoder navigation transformer, to aggregate discriminative spatio-temporal context information for object navigation. Our navigation encoder not only encodes spatial layouts and temporal agent poses but also innovatively constructs and encodes a passable frontier map, enriching the original state encoding with cues of potential exploration regions. Furthermore, our navigation decoder employs spatio-temporal self-attention and cross-attention mechanisms to model the dependencies among spatial layout encoding, temporal pose encoding, and passable frontier encoding, thereby facilitating comprehensive contextual state feature aggregation. Finally, we leverage these learned spatio-temporal contextual state representations for PPO-based navigation decisions. Extensive experiments on the Gibson, Habitat-Matterport3D (HM3D) and Matterport3D (MP3D) datasets demonstrate the superiority of our approach.

Code — https://github.com/Xie-Nav/NaviFormer

Introduction

Object navigation (ObjectNav), a key branch of embodied AI, serves as a foundational upstream task for various visual navigation-related applications, such as image-goal navigation (Kwon, Park, and Oh 2023) and vision-language navigation (An et al. 2024). In ObjectNav, an agent learns to recognize an unknown 3D environment from egocentric RGB-D images, enabling its navigation model to make informed decisions for locating specified targets. Therefore, how to sufficiently mine the spatio-temporal navigation cues from both the explored spatial layouts and the temporal agent pose trajectory to accurately guide navigation decisions has become crucial to the success of ObjectNav tasks.

*Corresponding author.

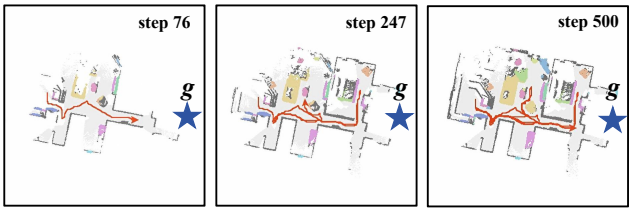

Figure 1: The red line represents the agent trajectory in a failure episode of CNN-based method (Chaplot et al. 2020b). The blue five-pointed star represents the position of the target object (i. e., toilet.). Despite the agent having conducted a more extensive exploration of the environment, the target object remains undiscovered.

Existing object navigation methods can be primarily divided into two categories: end-to-end methods and modular methods. End-to-end methods (Dang et al. 2023; Du, Yu, and Zheng 2020; Zhang et al. 2021) directly process egocentric RGB images to search for the target object by learning a mapping relationship between the current state and the optimal short-range action. While effective, the learned navigation models would inevitably stick into sub-optimal decisions when there is a significant discrepancy between the unknown scene layout and the prior knowledge, causing frequent collides with surrounding objects or obstacles. By contrast, the modular methods exploit the accumulated scene information in BEV (bird's-eye view) maps to determine long-term goals and search for targets. For example, (Chaplot et al. 2020b; Zhang et al. 2024, 2023a; Luo et al. 2022; Ramakrishnan et al. 2022; Yu, Kasaei, and Cao 2023c,b) employ a plain network such as CNNs to fuse the accumulated navigation cues, encompassing the explored spatial layouts and the temporal pose trajectory, forming current agent state representations. Nevertheless, due to the large spatial layouts in different scenes and the inherent local bias of CNN, trival navigation information fusion based on CNN is likely to cause the navigation model to ignore the long-range spatio-temporal dependencies between spatial layouts and temporal pose trajectories, leading to state representation misperception and potential decision failures, as demonstrated in Fig. 1.

In this paper, we propose a novel navigation trans-

former (NaviFormer) to sufficiently aggregate discriminative spatio-temporal context information for reliable object navigation. Specifically, our NaviFormer primarily consists of a passable frontier map-enhanced navigation encoder and a spatio-temporal navigation decoder. In our navigation encoder, in addition to encoding spatial layouts and temporal agent poses, we also generate an effective passable frontier map to identify the most promising regions for exploration. As such, we can encode this map to enhance original feature representations with rich exploration guidance. With the encoded spatial and temporal features above, our spatio-temporal navigation decoder then aims to learn the spatio-temporal contextual association across the spatial layouts, the temporal pose trajectories and passable frontiers for reliable navigation information fusion and discriminative state representations. Then, the self-attention module is employed to strengthen the spatial and temporal attributes of the spatial layouts and the temporal pose trajectory. Furthermore, based on the cross-attention mechanism, the spatial frontier decoder utilizes the current spatial layout features and the frontier features to obtain a critical frontier weight. The temporal pose decoder employs the temporal pose trajectory features and current spatial layout features to derive a target-related weight. Finally, a spatio-temporal contextual state representation combined with critical frontier weight and the target-related weight is considered to guide the navigation model to make reasonable decisions. Our NaviFormer is evaluated in Gibson, Habitat-Matterport3D (HM3D), and Matterport3D (MP3D), demonstrating superior performance in the ObjectNav task compared to other state-of-the-art methods.

Our main contributions are summarized in the following three points:

- We construct a novel navigation transformer, NaviFormer, to effectively aggregate spatio-temporal contextual navigation cues for discriminative agent state representations and reliable decisions.
- We develop a passable frontier map-enhanced navigation encoder for encoding spatio-temporal features of spatial layouts, temporal agent poses, and the passable frontier map. Notably, we innovatively build and integrate an effective passable frontier map to identify the most promising regions for exploration.
- We design a navigation decoder that leverages spatial layouts, temporal pose trajectorys, and passable frontiers to establish better spatio-temporal contextual state representations.
- Extensive experiments on Gibson, HM3D, and Matterport3D show our method's superiority, surpassing previous approaches in key metrics.

Related Work

End-to-end Method ObjectNav task (Ramakrishnan et al. 2022; Dang et al. 2023; Du, Yu, and Zheng 2020; Zhang et al. 2021) has seen significant advancements over the past decade. The end-to-end method (Hu et al. 2024a; Zhou et al. 2023a; Hu et al. 2024b) aims to establish a mapping relationship between egocentric images and short-term decisions. In addition to image features, the object features obtained by object detector (Ren et al. 2016; Redmon and Farhadi 2018; Zhu et al. 2024) are the excellent salient features, they can help navigation model learn scene representation, so most works (Zhang et al. 2023b; Li et al. 2021) use object features as the model input. The agent can effectively deduce the target position by leveraging the contextual object relations derived from the object features. Du et al. (Du, Yu, and Zheng 2020) and Zhang et al. (Zhang et al. 2021) use the object relationship to establish graphs. There are many similar methods (Ye and Yang 2021; Dang et al. 2022; Hu et al. 2024b; Li et al. 2021) like these. Some methods (Savinov, Dosovitskiy, and Koltun 2018; Zhu et al. 2019) use memory structures to preserve image features and object features, and the transformer (Wu et al. 2023, 2024; Dai et al. 2024) or graph (Savinov, Dosovitskiy, and Koltun 2018; Wu et al. 2019) structure are used to extract the important information from the memories. The end-to-end method, limited by its mechanism, outputs only short-term decisions and lacks long-term planning for navigation.

Modular Method Compared with the end-to-end method, the modular method (Chaplot et al. 2020b; Ramakrishnan et al. 2022) based on RL (Jiang, Xie, and Yang 2021; Jiang et al. 2022; Liu et al. 2023) is more popular. Point exploration (Chaplot et al. 2020a,b; Zhang et al. 2024) involves a navigation model that outputs specific position points within the scene as the subsequent set of long-term navigation goals. To simplify the exploration process, Luo et al. (Luo et al. 2022) ask the agent explore the four corners of the local map in a clockwise direction, but repeatedly traversing these corners can diminish the efficiency of navigation. Zhang et al. (Zhang et al. 2023a) selected one of the four corners as a long-term goal, which improved the exploration efficiency. Exploring frontiers (Yu, Kasaei, and Cao 2023c,b) is also a wise exploration strategy, because the agent must pass through these frontiers when exploring the scene. Compared to these methods, which overlook long-term spatio-temporal dependencies due to their plain networks, our method explicitly constructs spatio-temporal contextual state representations to mitigate suboptimal state recognition. With the development of large language models (Achiam et al. 2023) and visual language models (Radford et al. 2021), some works attempt to directly use LLM or VLM to guide the agent to explore the scene (Yokoyama et al. 2024; Yu, Kasaei, and Cao 2023a). However, slow response speed and communication interference in LLM- and VLM-assisted navigation can delay decision-making. Hence, we propose a smart, lightweight navigation model to tackle these challenges in ObjectNav.

Method

Task Definition

For the ObjectNav task, the agent is required to search for a specified target object (e.g., bed, chair) and navigate to it in an unseen environment. The initial position of the agent in the scene is set randomly. The agent receives the egocentric RGB-D observation $\boldsymbol{o}_t$, target object category $\boldsymbol{g}$, and pose $\boldsymbol{p}_t$ (e.g., coordinate $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, orientation $\boldsymbol{\theta}_t$) at timestamp t. The agent transmits these scene informations $(\boldsymbol{o}_t, \boldsymbol{g}, \boldsymbol{p}_t, \boldsymbol{\theta}_t)$

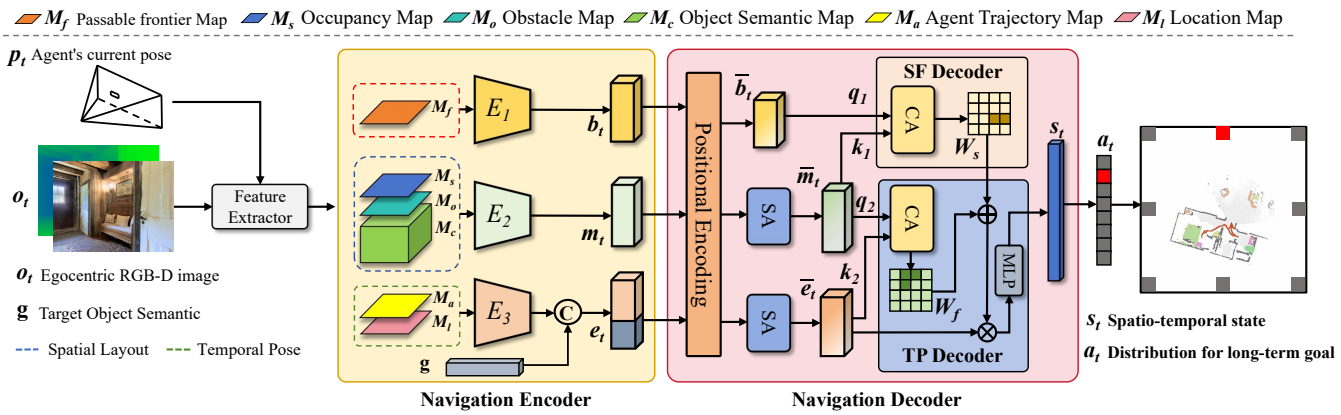

Figure 2: Framework of encoder-decoder navigation transformer (NaviFormer) for object navigation. The passable frontier map is constructed from the obstacle map and the outer edges of the occupancy map. The passable frontier map, spatial layouts, and the temporal pose trajectory are fed into the navigation encoder to output three different features. For the navigation decoder, the spatial frontier decoder (SF decoder) and the temporal pose decoder (TP decoder) use these three features to produce the critical frontier weights and the target-related weights. These weights enhance current state representation. Finally, the MLP processes the enhanced state representation and outputs the action distribution of the long-term goal.

to the navigation model, which outputs decisions to guide the agent to move. The discrete actions that the agent can perform include: $move\_forward$, $turn\_left$, $turn\_right$, and $stop$. The current episode ends when the agent automatically executes $stop$ action. After the episode ends, if the agent is within $1m$ of the target instance, the ObjectNav task is considered successful. The maximum time step for the agent to interact with the environment is 500.

Overview

Our navigation framework is shown in Fig. 2. The RGB-D images collected by the agent are merged into the BEV maps $\boldsymbol{M} \in \mathbb{R}^{C\times H\times W}$ through mathematical transformations. H and W represent the height and width of $\boldsymbol{M}$. The outer edges on the occupancy map $\boldsymbol{M}_s$ are subtracted from the obstacle map $\boldsymbol{M}_o$ to obtain a passable frontier map $\boldsymbol{M}_f$. $\boldsymbol{M}_f$, concatenated with the spatial layouts (i. e. $\boldsymbol{M}_s$, $\boldsymbol{M}_o$ and object semantic map $\boldsymbol{M}_c$) and the temporal pose trajectories (i. e., agent trajectory $\boldsymbol{M}_a$ and current agent position $\boldsymbol{M}_t$), is input into the navigation encoder to yield three distinct features: frontier features, spatial layout features, and temporal trajectory features. For navigation decoder consisting of the two subdecoders, spatial frontier decoder utilizes the spatial layout feature and frontier feature to derive a critical frontier weight. Temporal pose decoder leverages temporal pose features and spatial layout features to produce a target-related weight. These weights are then employed to establish spatio-temporal contextual state representations. Ultimately, MLP extracts the important spatio-temporal context information from state representations, and output the action distribution of the long-term goal.

Navigation Encoder

Passable Frontier Map Especially in the initial stage of ObjectNav tasks, identifying the regions that may contain a large amount of unknown critical scene information is extremely important based on the real-time explored spatial layout. The frontiers (Ramakrishnan et al. 2022; Yu, Kasaei, and Cao 2023c,b) are good exploration landmarks between explored free-spaces and unexplored regions, which can help the agent efficiently explore unknown environment and receive valuable scene information quickly. Therefore, we construct a passable frontier map based on the frontiers as prior prompt to explicitly help the navigation model recognize valuable scene cue in the explored spatial layouts over time. The construction process of the passable frontier map is shown in Fig. 3.

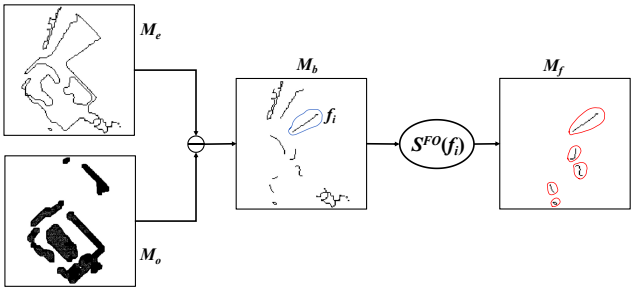

Figure 3: The occupancy map is transformed into the outer edge map. A candidate frontier map is then obtained by subtracting the obstacle map from the outer edge map. A scoring function assesses all frontiers in the candidate frontier map.

The contour extraction (Bradski 2000) is first utilized to delineate the outer edge map $\boldsymbol{M}_e \in \mathbb{R}^{H\times W}$ of the occupancy map $\boldsymbol{M}_s \in \mathbb{R}^{H\times W}$. By subtracting the obstacle map $\boldsymbol{M}_o$ from $\boldsymbol{M}_e$, the candidate frontier map $\boldsymbol{M}_b \in \mathbb{R}^{H\times W}$ is obtained. To improve the working efficiency of the model, the previous method (Yu, Kasaei, and Cao 2023b) used the largest outer edge of $\boldsymbol{M}_s$ to calculate the candidate frontiers,

which might overlook other valuable frontiers. In contrast, since our model treats the frontier as the prior prompt, even though $\boldsymbol{M}_b$ incorporates most of the frontiers, the computational complexity of the model does not increase as the number of frontiers grows. After getting $\boldsymbol{M}_b$, each frontier must be evaluated within $\boldsymbol{M}_b$ according to a scoring function S^{FO}. Frontiers that are farther from the agent's current position and cover a larger number of pixels are assigned higher scores, reflecting that they may contain rich scene information. However, frontiers that are excessively distant from the agent are overlooked, as the agents are currently unable to reach them. All frontiers within $\boldsymbol{M}_b$ get corresponding scores according to the following scoring function S^{FO}:

$$S^{FO}\left(f_i\right) = A_{f_i} + \lambda \cdot dist\left(\boldsymbol{p}_t, \boldsymbol{c}_{f_i}\right) \tag{1}$$

$$dist\left(\boldsymbol{p}_t, \boldsymbol{c}_{f_i}\right) = \begin{cases} \left\|\boldsymbol{p}_t - \boldsymbol{c}_{f_i}\right\|_2 & dist\left(\boldsymbol{p}_t, \boldsymbol{c}_{f_i}\right) < D \\ 0 & otherwise \end{cases} \tag{2}$$

Where f_i is the frontier that needs to be scored. A_{f_i} is the number of pixels occupied by f_i. $dist\left(\boldsymbol{p}_t, \boldsymbol{c}_{f_i}\right)$ represents the Euclidean distance between the center of f_i and the agent position $\boldsymbol{p}_t$. If the distance from the agent to f_i surpasses the predefined threshold D, the distance score of f_i is effectively nullified. λ is a constant. In $\boldsymbol{M}_b$, frontiers with lower scores are discarded, while those with higher scores are retained and incorporated into the passable frontier map $\boldsymbol{M}_f \in \mathbb{R}^{H \times W}$. As shown in Fig. 3, the red circle represents these retained frontiers. $\boldsymbol{M}_f$ includes most of the frontiers within the explored spatial layouts, leading to the fact that $\boldsymbol{M}_f$ can monitor the valuable scene information of the entire explored spatial layouts.

Encoder In order to utilize the characteristic that the passable frontier map can capture the valuable scene information based on the explored spatial layouts, we design a navigation encoder that not only encodes spatial layouts and temporal pose trajectories but also encodes the passable frontier map to better refine feature extraction with cues of potential exploration regions. The structure of navigation encoder is shown in Fig. 2.

The encoder contains three different sub-encoders: passable frontier encoder E_1, spatial layout encoder E_2 and temporal pose encoder E_3. These encoders adopt multi-layer convolutional network structure. The passable frontier map $\boldsymbol{M}_f$ is encoded into frontier feature $\boldsymbol{b}_t \in \mathbb{R}^{B_s \times H_2 \times W_2}$ through E_1. As mentioned in the Task Definition, the occupancy map $\boldsymbol{M}_s \in \mathbb{R}^{H \times W}$, the obstacle map $\boldsymbol{M}_o \in \mathbb{R}^{H \times W}$ and the object semantic map $\boldsymbol{M}_c \in \mathbb{R}^{C_o \times H \times W}$ can be regarded as the spatial layouts and encoded by E_2 to obtain the spatial layout feature $\boldsymbol{m}_t \in \mathbb{R}^{K_s \times H_2 \times W_2}$. The agent trajectory map $\boldsymbol{M}_a \in \mathbb{R}^{H \times W}$ and the current location map $\boldsymbol{M}_l \in \mathbb{R}^{H \times W}$ are encoded by E_3 to get the temporal trajectory feature. The target semantic feature $\boldsymbol{g} \in \mathbb{R}^{1 \times N}$ is replicated and reshaped and then concated with the temporal trajectory feature feature to form the agent temporal pose feature $\boldsymbol{e}_t \in \mathbb{R}^{A_s \times H_2 \times W_2}$.

Navigation Decoder

To establish better spatio-temporal context state representations, we build the navigation decoder, which consists of feature preprocessing, the spatial frontier (SF) decoder and the temporal pose (TP) decoder. Its structure is visually represented and detailed in Fig. 2.

Feature Preprocessing The feature preprocessing of the spatial layout feature $\boldsymbol{m}_t$, the frontier feature $\boldsymbol{b}_t$ and agent temporal pose features $\boldsymbol{e}_t$ is to prevent the time series information in $\boldsymbol{e}_t$ and spatial structure information in $\boldsymbol{b}_t$ and $\boldsymbol{m}_t$ from being destroyed during subsequent decoding.

$\boldsymbol{m}_t$, $\boldsymbol{b}_t$ and $\boldsymbol{e}_t$ are first preprocessed with positional encoding. The self-attention modules are used to further strengthen $\boldsymbol{m}_t$ and $\boldsymbol{e}_t$. As a result, the transformed representations $\bar{\boldsymbol{m}}_t \in \mathbb{R}^{K_s \times H_2 \times W_2}$ and $\bar{\boldsymbol{e}}_t \in \mathbb{R}^{A_s \times H_2 \times W_2}$ are obtained. Considering that the frontier information contained in $\boldsymbol{b}_t$ is usually smaller than $\boldsymbol{m}_t$ and $\boldsymbol{e}_t$, $\boldsymbol{b}_t$ does not need to be enhanced by the self-attention mechanism. We demonstrate in the supplementary material whether the self-attention module is needed and how many layers of self-attention modules make the navigation model optimal.

Spatial Frontier Decoder To utilize the explored spatial layout information to evaluate which frontiers may contain unknown valuable scene information, the spatial frontier decoder is designed.

The working process of the spatial frontier encoder is shown in Fig. 2. A cross-attention mechanism is employed to perform feature fusion on $\boldsymbol{b}_t$ and $\bar{\boldsymbol{m}}_t$. $\boldsymbol{b}_t$ is the query $\boldsymbol{q}_1$, $\bar{\boldsymbol{m}}_t$ is the key $\boldsymbol{k}_1$, the correlation between the two is calculated as follows:

$$\boldsymbol{W}_f = \mathrm{softmax}\left(\frac{\boldsymbol{b}_t \boldsymbol{w}_{q_1} \left(\bar{\boldsymbol{m}}_t \boldsymbol{w}_{k_1}\right)^T}{\sqrt{d}}\right) \tag{3}$$

Where $\boldsymbol{w}_{q_1}$ and $\boldsymbol{w}_{k_1}$ are learnable parameters. $\boldsymbol{W}_f \in \mathbb{R}^{hd_1 \times H_2 \times H_2}$ is the critical frontier weight. hd_1 is the number of heads in the cross-attention. The pixel value corresponding to the position in $\boldsymbol{W}_f$ represents the significance of the different frontiers in the passable frontier map $\boldsymbol{M}_f$. The valuable frontiers are assigned larger weights and are likely to contain very useful unknown scene information. Therefore, the critical frontier weight $\boldsymbol{W}_f$ indirectly marks more important frontiers in the passable frontier map $\boldsymbol{M}_f$. The navigation model can determine which frontiers have potential for exploration based on $\boldsymbol{W}_f$, and select those significant frontiers to assist the agent in exploring the scene, ensuring that the agent quickly obtain useful scene information in a limited number of steps.

Temporal Pose Decoder As the known scene layout continues to expand, the agent needs to concentrate on the target-related regions to locate the approximate target location. To ensure that the agent continues to focus on target-related regions after obtaining a large amount of valuable scene information based on $\boldsymbol{W}_f$, the temporal pose decoder is constructed. Fig. 2 shows the architecture of the temporal pose decoder.

The cross-attention mechanism is utilized to process spatial layout feature $\bar{\boldsymbol{m}}_t$ and agent temporal pose feature $\bar{\boldsymbol{e}}_t$. $\bar{\boldsymbol{m}}_t$ is query $\boldsymbol{q}_2$, $\bar{\boldsymbol{e}}_t$ is key $\boldsymbol{k}_2$. $\bar{\boldsymbol{m}}_t$ and $\bar{\boldsymbol{e}}_t$ first perform the

following correlation calculation:

$$\boldsymbol{W}_s = \text{softmax}\left(\frac{\bar{\boldsymbol{m}}_t \boldsymbol{w}_{q_2} \left(\bar{\boldsymbol{e}}_t \boldsymbol{w}_{k_2}\right)^T}{\sqrt{d_2}}\right) \tag{4}$$

Where $\boldsymbol{w}_{q_2}$ and $\boldsymbol{w}_{k_2}$ are learnable parameters. Since $\bar{\boldsymbol{e}}_t$ not only records the regions that the agent has passed through, but also contains the target semantic information, the calculation process in Eq. 4 can infer which regions in the scene are close to the agent and related to the target. The result of the inference is put into the target-related weight $\boldsymbol{W}_s \in \mathbb{R}^{hd_2 \times H_2 \times H_2}$. The larger the pixel values in $\boldsymbol{W}_s$, the higher the likelihood that the target object is present in the region corresponding to these pixels. Therefore, $\boldsymbol{W}_s$ marks the target-related region, which provides important clues for the agent to find the target.

After obtaining $\boldsymbol{W}_f$ and $\boldsymbol{W}_s$, to build better long-range spatio-temporal dependencies between the explored spatial layouts and temporal pose trajectory of the agent, $\boldsymbol{W}_f$ and $\boldsymbol{W}_s$ weight agent temporal pose feature $\bar{\boldsymbol{e}}_t$ to deduce more intuitive spatio-temporal contextual state representation:

$$\boldsymbol{s}_t = \bar{\boldsymbol{e}}_t + \text{softmax}\left(\boldsymbol{W}_f + \boldsymbol{W}_s\right) \bar{\boldsymbol{e}}_t \boldsymbol{w}_{v_2} \tag{5}$$

Where $\boldsymbol{w}_{v_2}$ is the learnable parameter. Since state $\bar{\boldsymbol{e}}_t$ receives the prompts provided by $\boldsymbol{W}_f$ and $\boldsymbol{W}_s$, $\boldsymbol{s}_t \in \mathbb{R}^{A_s \times H_2 \times W_2}$ contains both potentially valuable scene information and candidate target-related information.

The MLP processes $\boldsymbol{s}_t$ and output action distribution $\boldsymbol{a}_t \in \mathbb{R}^{1\times 8}$. As shown in Fig. 2, the agent select one of the eight candidate edge points on the map as a long-term goal (red point) based on $\boldsymbol{a}_t$. The map-edge exploration allows the agent to freely choose to explore the scene or search for target objects without being restricted by the spatial layout.

Navigation Policy

After the navigation model outputs the long-term goal $\boldsymbol{a}_t$, the Fast Marching Method (Sethian 1999) is used to plan the shortest path from the agent current position to the $\boldsymbol{a}_t$. The local policy calculates the discrete actions that the agent can perform based on the shortest path. The shortest path is continuously updated according to the changes of $\boldsymbol{a}_t$. The navigation model NaviFormer is trained with PPO (Schulman et al. 2017). And success reward and exploration reward based on the size of the explored spatial layouts are employed can ensures that the navigation decoder can provide optimal target-related weight $\boldsymbol{W}_s$ and critical frontier weight $\boldsymbol{W}_f$ to construct ideal spatio-temporal context state representations during training. The brute force untrap mode method (Luo et al. 2022) is adopted to help the agents avoid obstacles.

Experiments

Experimental Setup

Datasets and Evaluation Metric Our NaviFormer is evaluated on three common datasets: Gibson, Habitat-Matterport3D (HM3D) and Matterport3D (MP3D). For Gibson, We select 25 train/5 val scenes based on Gibson tiny split, 1000 val episodes are used to demonstrate the model performance. For HM3D, 80 train/20 val scenes are selected, and 2000 val episodes are used. The 6 object categories are chosen as target categories on Gibson and HM3D. For MP3D, 56 train/11 val scenes are employed, and 2195 val episodes are utilized. The 21 object categories are chosen as target categories on MP3D.

Following previous works (Zhang et al. 2024; Yu, Kasaei, and Cao 2023b; Ramakrishnan et al. 2022), we adopt three standard metrics to evaluate our method. Success rate (SR): the ratio of success episodes to total episodes, Success weighted by Path Length (SPL): the ratio of the optimal paths to the agent actual paths, which measures the navigation efficiency of the agent. Distance to Goal (DTS): the distance from the agent to the success threshold of the target object at the end of the episode.

Implementation Details For the agent interacting with Gibson, HM3D and MP3D, 3D indoor simulator Habitat platform (Savva et al. 2019) is employed to drive the three 3D sence datasets. The size of the egocentric RGB-D images received by the agent is $(4, 480, 640)$. We follow the previous method (Yu, Kasaei, and Cao 2023c) to perform semantic segmentation using RedNet (Jiang et al. 2018) and Mask-RCNN (He et al. 2017). The width and height of the BEV map are $(480, 480)$. The agent rotates 30° and moves $0.25m$ forward at each time step.

Comparisons with the State-of-the-art

The experimental results on Gibson, HM3D, and MP3D (Tab. 1) show that our method outperforms recent state-of-the-art methods across most metrics. As the key metric for navigation tasks, our method surpasses SGM (Zhang et al. 2024) in SR by $4.6\%/1.5\%/2.4\%$ on Gibson/HM3D/MP3D val, demonstrating significant advantages in the ObjectNav task. For the efficiency metric SPL, our NaviFormer surpasses all the state-of-the-art methods on HM3D and MP3D, and reaches the average on Gibson, which shows that our method can still maintain good efficiency when facing different types of scenes. Our method has similar results on DTS, our method outperforms all the state-of-the-art methods on Gibson and HM3D.

Notably, as shown in Tab. 2, navigation methods based on large language models (LLMs) or vision-language models (VLMs) fail to outperform ours in SR. This suggests that while LMs provide useful decision-making guidance, there remains a gap between their scene understanding and real-world distributions. Additionally, our method's average inference speed (FPS) is 2–3 times faster than these large model-based methods.

Ablation Study

In order to validate whether each submodule of our NaviFormer is reasonable, we design the multiple experimental schemes to compare with our method.

Passable Frontier Map We set four experiments to examine the effects of the passable frontier map $\boldsymbol{M}_f \in \mathbb{R}^{H\times W}$ on navigation model performance.

$\boldsymbol{\Phi_1}$: **CNN**. Multi-layer convolutional network encodes the

Method	Presented at	Gibson			HM3D			MP3D		
		SR (%)↑	SPL(%)↑	DTS(m)↓	SR(%)↑	SPL(%)↑	DTS(m)↓	SR(%)↑	SPL(%)↑	DTS(m)↓
DD-PPO (Wijmans et al. 2020)	ICLR'20	15.0	10.7	3.24	27.8	14.5	6.49	8.0	1.8	7.93
ANS(Chaplot et al. 2020a)	ICLR'20	67.1	34.9	1.66	27.3	9.2	5.80	-	-	-
SemExp(Chaplot et al. 2020b)	NeurIPS'20	71.7	39.6	1.39	-	-	-	28.3	10.9	6.06
L2M(Georgakis et al. 2022)	ICLR'22	-	-	-	-	-	-	32.1	11.0	5.12
PONI(Ramakrishnan et al. 2022)	CVPR'22	73.6	41.0	1.25	-	-	-	31.8	12.1	5.10
Stubborn(Luo et al. 2022)	IROS'22	-	-	-	-	-	-	31.2	13.5	5.01
3D-aware(Zhang et al. 2023a)	CVPR'23	74.5	42.1	1.16	52.4	24.5	4.25	34.0	14.6	**4.74**
Peanut(Zhai and Wang 2023)	ICCV'23	77.3	42.5	1.27	58.8	29.3	3.51	35.7	14.6	4.88
FSE(Yu, Kasaei, and Cao 2023b)	ICRA'23	71.5	36.0	1.35	53.8	24.6	3.75	-	-	-
SGM(Zhang et al. 2024)	CVPR'24	78.0	**44.0**	1.11	59.8	29.4	3.47	37.7	14.7	4.93
NaviFormer (Ours)	AAAI'25	**82.6**	40.9	**0.76**	**61.3**	**29.6**	**3.40**	**40.1**	**15.1**	5.19

Table 1: Comparisons with the state-of-the-art methods on Gibson/HM3D/MP3D val. Due to the different settings or the lack of some important metrics, some methods (Peanut, SGM) are re-implemented according to our experimental setup.

Method	Presented at	Gibson			HM3D			MP3D			
		SR(%)↑	SPL(%)↑	DTS(m)↓	SR(%)↑	SPL(%)↑	DTS(m)↓	SR(%)↑	SPL(%)↑	DTS(m)↓	FPS↑
ESC(Zhou et al. 2023b)	ICML'23	-	-	-	39.2	22.3	-	28.7	14.2	-	3
L3mvn(Yu, Kasaei, and Cao 2023c)	IROS'23	76.9	38.8	1.01	54.2	25.5	3.93	-	-	-	4
PixNav(Cai et al. 2024)	ICRA'24	-	-	-	37.9	20.5	-	-	-	-	2
NaviFormer (Ours)	AAAI'25	**82.6**	**40.9**	**0.76**	**61.3**	**29.6**	**3.40**	**40.1**	**15.1**	5.19	**8**

Table 2: Comparisons with the Large-Model-based zero-shot methods on Gibson/HM3D/MP3D val.

Method	Gibson			HM3D		
	SR(%)	SPL(%)	DTS(m)	SR(%)	SPL(%)	DTS(m)
$\boldsymbol{\Phi_1}$	78.8	**44.6**	**1.056**	59.1	**29.3**	**3.493**
$\boldsymbol{\Phi_2}$	**79.7**	43.5	1.160	**59.8**	27.5	3.934
$\boldsymbol{\Phi_3}$	78.9	41.5	1.239	57.4	27.7	3.874
$\boldsymbol{\Phi_4}$	79.5	42.5	1.186	58.7	28.3	3.881

Table 3: Ablation experiments of the passable frontier map on Gibson/HM3D val.

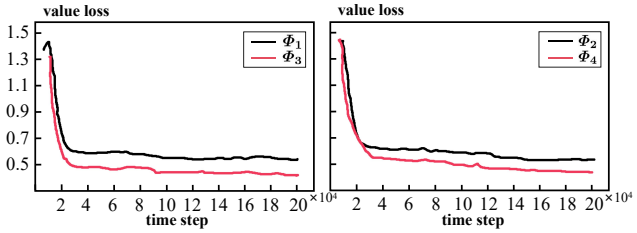

Figure 4: The value loss of different methods throughout training process.

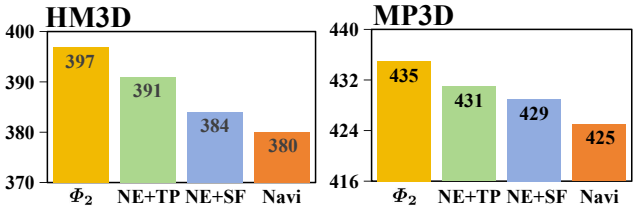

Figure 5: The number of exploration failures of different variants on HM3D and MP3D.

spatial layout and temporal pose trajectory.

$\boldsymbol{\Phi_2}$: **CNN** + $\boldsymbol{M}_f$. Multi-layer convolutional network encodes the spatial layout, temporal pose trajectory and passable frontier map $\boldsymbol{M}_f$.

$\boldsymbol{\Phi_3}$: **ViT**. Multi-layer self-attention blocks encodes the spatial layout, temporal pose trajectory.

$\boldsymbol{\Phi_4}$: **ViT** + $\boldsymbol{M}_f$. Multi-layer self-attention blocks encodes the spatial layout, temporal pose trajectory and passable frontier map $\boldsymbol{M}_f$.

The results of these variants and methods in Gibson/HM3D are shown in Tab. 3. From the results in the table, it can be seen that both the CNN-based method $\boldsymbol{\Phi_2}$ and the transformer-based method $\boldsymbol{\Phi_4}$ have a higher SR when they use the passable frontier map as input, indicating that the passable frontier map indeed assists the agent in exploring more useful scene information. For the value loss curves of the two methods in Fig. 4, we can see that the convergence results of $\boldsymbol{\Phi_3}$/$\boldsymbol{\Phi_4}$ is better than that of $\boldsymbol{\Phi_1}$/$\boldsymbol{\Phi_2}$, but the performance of transformer-based methods is worse than that of simple CNN-based methods. These phenomena indicate that although transformer have stronger learning capabilities than CNN, for object navigation tasks, transformer-based structures cannot better implicitly represent the long-range dependencies between spatial layouts and temporal agent poses.

Navigation Encoder In Tab. 4, we can see that the performance of NE-only method is not as good as that of CNN-based method $\boldsymbol{\Phi_2}$. This is because the lack of decoder, resulting in multiple features forming information silos and thus failing to model the connection between patial layouts and temporal agent poses.

Besides, we can also see that the performance of the

Figure 6: Successful episode with NaviFormer in HM3D val. Each figure represents explored spatial layout from the perspective of BEV. The upper left corner of each figure is the egocentric RGB image. The different colors in the Each figure represent the various object categories perceived by the agent. The red line represents the agent trajectory and the arrow is the current orientation of the agent.

NE	SF	TP	Gibson			HM3D		
			SR(%)	SPL(%)	DTS(m)	SR(%)	SPL(%)	DTS(m)
	Φ_2		79.7	43.5	1.160	59.8	27.5	3.934
✓			78.4	**43.6**	1.233	57.9	26.3	3.874
✓	✓		81.4	40.3	1.146	60.7	28.3	3.765
✓		✓	80.6	42.6	1.047	60.4	28.9	3.648
✓	✓	✓	**82.6**	40.9	**0.763**	**61.3**	**29.6**	**3.404**

Table 4: Ablation experiments of different sub-modules in NaviFormer. NE is global sence feature encoder, SF is spatial frontier decoder, TP is temporal pose decoder.

model has been significantly improved when the spatial frontier decoder (SF) or the temporal pose decoder (TP) are used for feature fusion. The SR of NE+SF and NE+TP are 3.0%/2.8% and 2.2%/2.5% higher than NE-only method in Gibson/HM3D. In summary, our encoder and SF decoder, along with the TP decoder, exhibit complementary effects. When used together, the model attains its peak efficacy and demonstrates superior performance metrics.

Navigation Decoder In Tab. 4, compared with the baseline $\boldsymbol{\Phi_2}$, the SR of NE+SF and NE+TP are 1.7%/0.9% and 0.9%/0.6% higher on Gibson/HM3D. The results support that using SF and TP to explicitly capture the important scene region is more reliable than using $\boldsymbol{\Phi_2}$ to implicitly learn the spatio-temporal relationship between patial layouts and temporal agent poses.

We also observe that the SR of NE+TP is lower than that of NE+SF, which shows that exploring the frontier is more conducive to locating targets than directly exploring the target-related region. However, the SPL of NE+SF is lower than that of NE+TP, excessive concentration on to the frontier feature compromise the efficiency of the navigation model. Our viewpoint is further validated in Fig. 5, which presents the number of exploration failures for $\boldsymbol{\Phi_2}$, NE+SF, NE+TP, and NaviFormer on HM3D/MP3D val. NE-based methods (+SF, +TP, SF+TP) outperform $\boldsymbol{\Phi_2}$ with fewer failures. NE+SF achieves fewer failures than NE+TP, highlighting the importance of frontier-based exploration over direct target search. NaviFormer (NE+SF+TP) integrates the strengths of the spatial frontier decoder and temporal pose decoder, balancing frontier exploration and target localization, thereby further enhancing SR, SPL, and DTS to a competitive level.

Qualitative Analysis

To demonstrate how NaviFormer guides the agent in searching for the target object, we test our method on the failure episode mentioned in the introduction, and the performance of the model is shown in Fig. 6. The agent needs to navigate near the toilet in an unseen environment. At the initial stage of the task, the agent explores the more spacious regions in the scene. As the time step increases, most regions in the scene are explored, the agent begins to infer the approximate location of the target based on the known spatial layout and continuously visits the explored scene locations, which takes a lot of time for the agent. After a period of searching, the agent locates the target-related region (i. e., the blue circle in the fourth figure) at the end of the episode and gradually approaches the target object. When the agent detects the target via segmentation model (blue circle in the last figure), it swiftly navigates to the target.

Conclusion

In this paper, we proposed a novel NaviFormer model to solve the ignorance of the complex long-range dependencies between the spatial layouts and the temporal agent pose trajectory. We first bulid the passable frontier maps to assist NaviFormer recognizing valuable scene cue. Navigation encoder then encodes the spatial layouts, the temporal agent poses and the novel passable frontier maps respectively. Based on these state encoding, navigation decoder builds better long-range spatio-temporal contextual state representations by the spatial frontier decoder and the temporal pose decoder. Extensive experiments on Gibson, HM3D and MP3D demonstrate the superiority of our method.

Acknowledgments

This work was supported by the National Science Fund of China (Grant Nos. 62276144, 62176124).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

An, D.; Wang, H.; Wang, W.; Wang, Z.; Huang, Y.; He, K.; and Wang, L. 2024. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bradski, G. 2000. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11): 120–123.

Cai, W.; Huang, S.; Cheng, G.; Long, Y.; Gao, P.; Sun, C.; and Dong, H. 2024. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *Proceedings of International Conference on Robotics and Automation*, 5228–5234.

Chaplot, D. S.; Gandhi, D.; Gupta, S.; Gupta, A.; and Salakhutdinov, R. 2020a. LEARNING TO EXPLORE USING ACTIVE NEURAL SLAM. In *Proceedings of the International Conference on Learning Representations*.

Chaplot, D. S.; Gandhi, D. P.; Gupta, A.; and Salakhutdinov, R. R. 2020b. Object goal navigation using goal-oriented semantic exploration. In *Proceedings of the Advances in Neural Information Processing Systems*, 4247–4258.

Dai, F.; Zhu, Y.; Shen, Y.; Xie, J.; and Qian, J. 2024. Dense Voxel Representation Network for Implicit Scene Completion. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.

Dang, R.; Shi, Z.; Wang, L.; He, Z.; Liu, C.; and Chen, Q. 2022. Unbiased directed object attention graph for object navigation. In *Proceedings of the ACM International Conference on Multimedia*, 3617–3627.

Dang, R.; Wang, L.; He, Z.; Su, S.; Tang, J.; Liu, C.; and Chen, Q. 2023. Search for or navigate to? dual adaptive thinking for object navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8250–8259.

Du, H.; Yu, X.; and Zheng, L. 2020. Learning object relation graph and tentative policy for visual navigation. In *Proceedings of the European Conference on Computer Vision*, 19–34.

Georgakis, G.; Bucher, B.; Schmeckpeper, K.; Singh, S.; and Daniilidis, K. 2022. Learning to Map for Active Semantic Goal Navigation. In *Proceedings of the International Conference on Learning Representations*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2961–2969.

Hu, X.; Lin, Y.; Fan, H.; Wang, S.; Wu, Z.; and Lv, K. 2024a. Building Category Graphs Representation with Spatial and Temporal Attention for Visual Navigation. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(7): 1–22.

Hu, X.; Lin, Y.; Wang, S.; Wu, Z.; and Lv, K. 2024b. Agent-centric relation graph for object visual navigation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(2): 1295–1309.

Jiang, H.; Li, G.; Xie, J.; and Yang, J. 2022. Action Candidate Driven Clipped Double Q-Learning for Discrete and Continuous Action Tasks. *IEEE Transactions on Neural Networks and Learning Systems*.

Jiang, H.; Xie, J.; and Yang, J. 2021. Action candidate based clipped double q-learning for discrete and continuous action tasks. In *Proceedings of the AAAI conference on artificial intelligence*, 7979–7986.

Jiang, J.; Zheng, L.; Luo, F.; and Zhang, Z. 2018. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint arXiv:1806.01054*.

Kwon, O.; Park, J.; and Oh, S. 2023. Renderable neural radiance map for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9099–9108.

Li, W.; Song, X.; Bai, Y.; Zhang, S.; and Jiang, S. 2021. Ion: Instance-level object navigation. In *Proceedings of the ACM International Conference on Multimedia*, 4343–4352.

Liu, S.; Zhou, Y.; Song, J.; Zheng, T.; Chen, K.; Zhu, T.; Feng, Z.; and Song, M. 2023. Contrastive Identity-Aware Learning for Multi-Agent Value Decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11595–11603.

Luo, H.; Yue, A.; Hong, Z.-W.; and Agrawal, P. 2022. Stubborn: A strong baseline for indoor object navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3287–3293.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of International Conference on Machine Learning*, 8748–8763.

Ramakrishnan, S. K.; Chaplot, D. S.; Al-Halah, Z.; Malik, J.; and Grauman, K. 2022. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18890–18900.

Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1137–1149.

Savinov, N.; Dosovitskiy, A.; and Koltun, V. 2018. Semi-parametric topological memory for navigation. In *Proceedings of the International Conference on Learning Representations*.

Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; et al. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9339–9347.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sethian, J. A. 1999. Fast marching methods. *SIAM review*, 41(2): 199–235.

Wijmans, E.; Kadian, A.; Morcos, A.; Lee, S.; Essa, I.; Parikh, D.; Savva, M.; and Batra, D. 2020. DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames. In *Proceedings of the International Conference on Learning Representations*.

Wu, Y.; Song, H.; Liu, B.; Zhang, K.; and Liu, D. 2023. Co-salient object detection with uncertainty-aware group exchange-masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19639–19648.

Wu, Y.; Wu, Y.; Tamar, A.; Russell, S.; Gkioxari, G.; and Tian, Y. 2019. Bayesian relational memory for semantic visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2769–2779.

Wu, Y.; Zhang, K.; Qian, J.; Xie, J.; and Yang, J. 2024. Text2LiDAR: Text-guided LiDAR Point Cloud Generation via Equirectangular Transformer. In *Proceedings of the European Conference on Computer Vision*, 291–310.

Ye, X.; and Yang, Y. 2021. Hierarchical and partially observable goal-driven policy learning with goals relational graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14101–14110.

Yokoyama, N.; Ha, S.; Batra, D.; Wang, J.; and Bucher, B. 2024. VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation. In *Proceedings of International Conference on Robotics and Automation*, 42–48.

Yu, B.; Kasaei, H.; and Cao, M. 2023a. Co-NavGPT: Multi-robot cooperative visual semantic navigation using large language models. *arXiv preprint arXiv:2310.07937*.

Yu, B.; Kasaei, H.; and Cao, M. 2023b. Frontier semantic exploration for visual target navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 4099–4105.

Yu, B.; Kasaei, H.; and Cao, M. 2023c. L3mvn: Leveraging large language models for visual target navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3554–3560.

Zhai, A. J.; and Wang, S. 2023. Peanut: Predicting and navigating to unseen targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10926–10935.

Zhang, J.; Dai, L.; Meng, F.; Fan, Q.; Chen, X.; Xu, K.; and Wang, H. 2023a. 3d-aware object goal navigation via simultaneous exploration and identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6672–6682.

Zhang, S.; Song, X.; Bai, Y.; Li, W.; Chu, Y.; and Jiang, S. 2021. Hierarchical object-to-zone graph for object navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15130–15140.

Zhang, S.; Song, X.; Li, W.; Bai, Y.; Yu, X.; and Jiang, S. 2023b. Layout-based causal inference for object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10792–10802.

Zhang, S.; Yu, X.; Song, X.; Wang, X.; and Jiang, S. 2024. Imagine Before Go: Self-Supervised Generative Map for Object Goal Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16414–16425.

Zhou, K.; Guo, C.; Guo, W.; and Zhang, H. 2023a. Learning heterogeneous relation graph and value regularization policy for visual navigation. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhou, K.; Zheng, K.; Pryor, C.; Shen, Y.; Jin, H.; Getoor, L.; and Wang, X. E. 2023b. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *Proceedings of the International Conference on Machine Learning*, 42829–42842.

Zhu, G.; Lin, Z.; Yang, G.; and Zhang, C. 2019. Episodic reinforcement learning with associative memory. In *Proceedings of the International Conference on Learning Representations*.

Zhu, Y.; Hui, L.; Shen, Y.; and Xie, J. 2024. SPGroup3D: Superpoint Grouping Network for Indoor 3D Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7811–7819.