Adaptive Strategy Weighting with Fault Tolerant Localization for Object Navigation

1st Yanwei Zheng 2nd Shaopu Feng 3rd Bowen Huang 4th Changrui Li 5th Xiao Zhang 6th Dongxiao Yu*

2025 IEEE International Conference on Multimedia and Expo (ICME) | 979-8-3315-9495-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICME59968.2025.11210064

***Abstract*—End-to-end navigation models commonly incorporate multiple sub-modules, each designed for distinct purposes such as searching, obstacle avoidance, and target localization. However, agents equipped with these modules may still struggle to apply the appropriate strategies at the right locations and stages. For instance, agent might incorrectly rely on the search or localization module for obstacle avoidance, reducing adaptability in dynamic environments. Additionally, existing methods assume the recognition for target object is always correct, neglecting the unavoidable misclassification caused by visually similar objects. To apply appropriate strategy for a given situation, we introduce Adaptive Strategy Feature Fusion (ASFF). It heuristically assigns appropriate weights to different sub-modules based on current observation and memory state, enabling flexible integration with arbitrary sub-module combinations. To improve localization in the presence of misclassification, we propose Fault Tolerant Target Memory Aggregator (FTTMA), a module that uses clustering-based sparse self-attention and target cross-attention to minimize interference from misclassified object, providing accurate target orientation to the agent. Experiments on the AI2THOR and RoboTHOR datasets, including both typical and zero-shot navigation tasks, demonstrate that our model outperforms the state-of-the-art (SOTA) methods in both success rate and navigation efficiency.**

***Index Terms*—Object Navigation, Fault Tolerant, Sparse Attention, Zero-Shot**

I. Introduction

Object navigation requires an agent to search for and approach a target object in an unfamiliar environment, using egocentric RGB views. With the advancements in computer vision and natural language processing, end-to-end methods [1]–[6] based on reinforcement learning [7] have been widely used in navigation.

Existing navigation models typically employ multi-module architectures [6], [8], [9], which incorporate various sub-modules to perform different navigation strategies like searching, localization and collision avoidance (shown in Figure 1). Examples include search-purpose modules [3], [5], [10] that explore unknown environments by leveraging object semantics correlations, as well as collision avoidance modules which construct obstacle maps [6], [11] through trial and error. Recent research has also introduced target memory aggregator modules [6], [8], [9] to improve target localization. These aggregators maintain a target memory buffer that stores detection records, including the target's object detection results and the agent's relative position throughout the episode.

All authors are affiliated with the School of Computer Science and Technology, Shandong University, Shandong Province, China. The email for corresponding author is 202315157@mail.sdu.edu.cn.

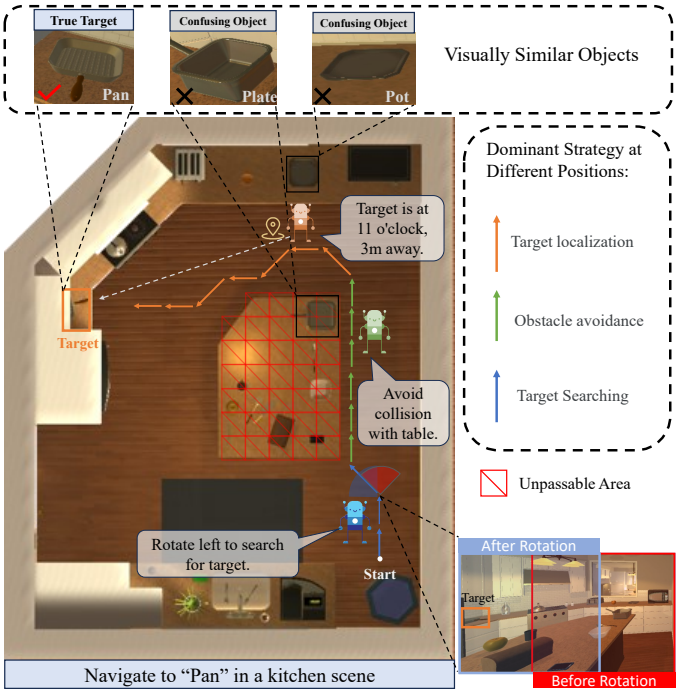

Fig. 1. An ideal navigation example. An excellent and comprehensive navigation agent should be able to leverage different abilities based on current observations and historical memory, adopting flexible navigation behaviors to adapt to dynamic environments. Additionally, for objects that are visually similar to the target in the scene, the agent should distinguish them through multi-angle and positional observations, thereby accurately locating the target.

Their integration significantly enhances localization accuracy and improves overall navigation performance. However, the increased incorporation of sub-modules also heightens the unpredictability of training, leading to mismatches between the sub-modules and their intended strategy, which in turn reduces the agent's adaptability to dynamic environments. For instance, when the agent is in a corner, it may rely on the search-purpose module to memorize the layout of the training scene thus avoiding collisions. Alternatively, when target localization is required in the later stage of navigation, the collision avoidance module might become overly activated unexpectedly. In addition, previous studies assume that the agent can always accurately recognize target object using object detectors [12], [13]. But in practice, limitations such as observation distance [14], viewing angle, image resolution, and insufficient training data often result in misclassification between visually similar objects. For example, in the kitchen

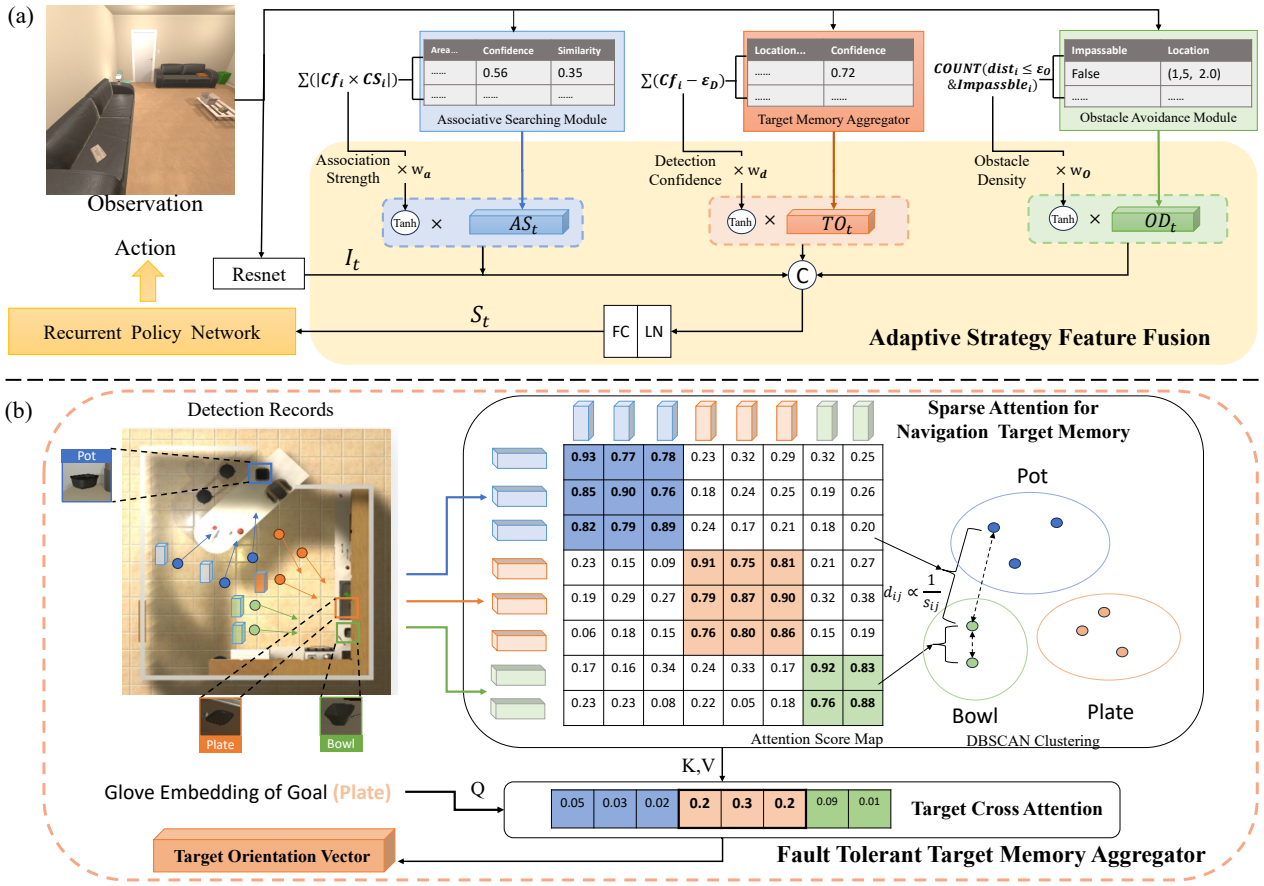

Fig. 2. **(a)**: Model overview and illustration of Adaptive Strategy Feature Fusion (ASFF). LN: LayerNorm, FC: FullyConnected, Cf_i: Detection confidence of i-th object, $\varepsilon_D, \varepsilon_O$: Threshold parameter. After obtaining the associative searching vector AS_t, target orientation vector TO_t, obstacle distribution vector OD_t from corresponding modules, ASFF assigns appropriate weights to fuse them based on three heuristic indicators: Association Strength, Detection Confidence and Obstacle Density. The resulting fused representation S_t is used by the policy network to determine the action a_t. **(b)**:Fault Tolerant Target memory aggregator (FTTMA). The input to FTTMA are the object detection records of target object, which may include misclassified ones from visually similar objects. In the sparse attention layer, attention scores are initially calculated for all record pairs. During clustering, the distance between two records is negatively correlated to their score. Based on the clustering results, only the attention computations within the same cluster are retained, while inter-cluster ones are masked. In the subsequent cross-attention layer, the target's glove embedding is used as queries to extract the final target orientation vector.

scenes, pots, bowls and pans are frequently confused with one another (as shown in Figure 1). These misclassified records stored in the memory buffer prevent the aggregator from guiding the agent toward the true direction of the target object, resulting in inefficient or even failed navigation.

In order to enhance the adaptability across different locations and stages of navigation, we introduce the Adaptive Strategy Feature Fusion (ASFF), a feature fusion method applicable to various modules and their combinations. It dynamically distributes the agent's attention weights to three core strategy modules—searching, localization, and obstacle avoidance—based on the current observation and memory state.

To improve localization accuracy in the presence of misclassified detections, we propose the Fault Tolerant Target Memory Aggregator (FTTMA), which consists of a sparse self-attention [15] layer and a target cross-attention layer. The self-attention layer utilizes a sparse attention mechanism based on DBSCAN [16], enabling detection records associated with the same object in the memory buffer to converge and be refined, while records corresponding to different objects are excluded from mutual attention calculations, thus reducing the interference from misclassified objects. In the following cross-attention layer, the semantic embedding of the target is used as queries to extract the final orientation feature, which contains information specific to the target. To summarize, our contributions are as follows:

1)We propose Adaptive Strategy Feature Fusion (ASFF), an efficient and versatile feature fusion method for object navigation.

2)We propose the Fault Tolerant Target Memory Aggregator (FTTMA), which employs a clustering-based sparse attention mechanism to accurately localize the target, even in the presence of visually similar objects.

3)We conducted extensive experiments on the AI2THOR and RoboTHOR datasets, evaluating both typical and zero-shot navigation tasks, demonstrating effectiveness of our method.

II. METHOD

A. Problem Setting

The agent is initialized to a random reachable position in the scene with random pitch and yaw angles: $s = (x, y, \theta, \beta)$ and random target g. According to the RGB image o_t and target g, the agent learns a navigation strategy $\pi(a_t|o_t, g)$, where $a_t \in A = \{MoveAhead, RotateLeft, RotateRight, LookDown, LookUp, Done\}$. The episode is considered successful if the agent outputs $Done$ while being within 1.0 meter of the visible target.

B. Overview

The overall structure of our model is illustrated in Figure 2(a). Upon receiving the observation, the agent processes it using ResNet18 [17] to generate the global image embedding, I_t. Leveraging object detection features, the associative search module VAG (proposed in [5]) computes the associative searching vector, AS_t. Meanwhile, our proposed fault tolerant target memory aggregator (detailed in Figure 2(b)) outputs the target orientation vector, TO_t. The obstacle avoidance module GM (proposed in [11]) generates the obstacle distribution vector, OD_t. These features are then fused using our adaptive strategy feature fusion, resulting in the fused representation S_t, which is passed into a recurrent policy network to determine the action a_t.

C. Adaptive Strategy Feature Fusion for Object Navigation

Current navigation models typically incorporate multiple modules, each with distinct capabilities, to enable flexible navigation strategies. However, the capabilities required by the agent vary depending on the location and stage within an episode. The existing feature fusion methods, which simply concatenate features from different modules, fail to account for this variability, resulting in insufficient adaptability to dynamic environments. Therefore, we propose three importance indicators that reflect the demand for the core navigation strategies—searching, localization, and obstacle avoidance—based on the current observation and memory state. Leveraging these indicators, we introduce the Adaptive Strategy Feature Fusion (ASFF) for object navigation, as shown in Figure 2(a).

Throughout most of the navigation process, particularly when the target has not been found, the agent's primary strategy is to search for the target. In existing studies, various modules (such as VAG [5], VTNet [10], ORG [3], etc.) based on the spatial and semantic relationships between observed objects and target objects have been proposed for target searching. The input for these modules typically consists of the detection results of observed objects. Based on the objects semantic correlation, we propose the indicator **Association Strength** to reflect the importance of these module according to current observation:

$$AssociationStrength = \sum_{i=1}^{\widehat{No}} (|Conf_i \cdot CS(g, i)|), \tag{1}$$

where $\widehat{No}$ is the number of detected objects in the current frame, $Conf_i$ is the detection confidence of the i-th detected object, and $CS(g, i)$ is the cosine similarity between the glove embeddings of the target and i-th object type.

After discovering the target, accurately localizing the target becomes crucial for efficiently planning the path to complete the navigation. This strategy is typically handled by memory aggregators (such as TAMSA [9], NTWA [6], and our proposed FTTMA). The input to the aggregators typically consists of target detection records from various locations throughout an episode. Generally, the more frequently a target object is detected with higher confidence, the more accurate and valuable the information provided by the aggregator. Therefore, the **Detection Confidence** is proposed to reflect the importance of the aggregator:

$$DetectionConfidence = \sum_{i=1}^{\widehat{Nr}} (Conf_i - \varepsilon_D) \tag{2}$$

where the confidence score of each object is ensured to be greater than the threshold parameter ε_D, and $\widehat{Nr}$ represents the number of records stored in the target memory.

During navigation, the agent may collide with walls and furniture at corners or narrow corridors, even fall into deadlock. In this context, the obstacle avoidance module becomes essential. Based on historical trial-and-error experience, the obstacle avoidance module uses explicit [11] or implicit maps [6] to depict the obstacles distribution and reduce collisions. Since obstacles at a distance have minimal impact on current navigation decision, we propose **Obstacle Density** indicator:

$$\begin{aligned} ObstacleDensity =& COUNT(dist_i \leq \varepsilon_O \& Impassble_i) \\ & i = 1, 2, \cdots, \widehat{Np} \end{aligned} \tag{3}$$

where $dist_i$ is the distance of the i-th recorded location points relative to the current position, ε_O is the distance threshold and $\widehat{Np}$ is the total number of location points recorded by the module, $Impassable_i$ is a Boolean value indicating that the i-th map block or record is marked as impassable when true.

With above indicators that reflect the importance of each module, we also introduce three learnable parameters w_a, w_d, w_o for each branch to enhance the adaptability. The final feature fusion formula is as follows:

$$p_a = tanh(AssociationStrength \cdot w_a), \tag{4}$$

$$p_d = tanh(DetectionConfidence \cdot w_d), \tag{5}$$

$$p_o = tanh(ObstacleDensity \cdot w_o), \tag{6}$$

$$S_t = FC(LN([p_a \cdot AS_t, p_d \cdot TO_t, p_o \cdot OD_t, I_t])), \tag{7}$$

where I_t is the image embedding extracted by ResNet18. The state representation S_t is used as the input of the policy network to produce the navigation action a_t.

D. Fault Tolerant Target Memory Aggregator

During navigation, due to the inevitable limitations of observation conditions, the agent may misclassify visually similar objects when attempting to detect the target. These erroneous

detection records are stored and accumulated in the target memory buffer, hindering the agent in accurately localizing the target when using existing target memory aggregators [6], [9]. To address this issue, we propose Fault Tolerant Target Memory Aggregator (FTTMA). Its main contribution is to differentiate visually similar objects through clustering and perform sparse attention computation based on the clustering results: attention is computed only between records within the same cluster, while interactions between records from different clusters are masked, effectively eliminating the impact of misclassified records on target localization. As shown in Figure 2(b), FTTMA consists of a sparse self-attention layer based on DBSCAN and the following target cross attention-layer. Its input is detection records $\{D_1, D_2, ...D_n\}$ stored in the target memory buffer. Each record D_i consists the coordinates of the bounding box $b_i \in \mathbb{R}^{1\times4}$, confidence value $c_i \in \mathbb{R}$, agent's relative position and pose $l_i \in \mathbb{R}^{1\times4}$, object visual appearance feature $v_i \in \mathbb{R}^{1\times256}$ extracted from the penultimate layer of the DETR [13] decoder.

In the sparse attention layer, we designed two sets of queries and keys that focus on both object location and appearance similarity respectively. The formula for queries, keys and values of the sparse attention are as follows:

$$Q_i^l = L_2Norm([[b_i, l_i]W_l^Q, 1]), Q_i^v = L_2Norm(v_iW_v^Q) \tag{8}$$

$$K_i^l = L_2Norm([[b_i, l_i]W_l^K, 1]), K_i^v = L_2Norm(v_iW_v^K) \tag{9}$$

$$V^i = [b_i, l_i, v_i]W^V \quad i = 1, 2, ..n, \tag{10}$$

where $L_2Norm(x) = \frac{x}{\|x\|_2}$, the additional scalar 1 is concatenated to ensure the uniqueness mapping after the normalization. To adaptively balance the focus on object position and visual appearance similarity, we introduce a learnable parameter α when calculating the attention score:

$$A_{ij} = (1 - \sigma(\alpha))Q_i^l(K_j^l)^T + \sigma(\alpha)Q_i^v(K_j^v)^T, \tag{11}$$

where $\sigma(\cdot)$ is the sigmoid function.

To implement the DBSCAN clustering, we set the distance between record pairs D_i and D_j as follows:

$$\begin{aligned} d_{ij} =&(1 - \sigma(\alpha))(||(Q_i^l - K_j^l)||_2 + ||(Q_j^l - K_i^l)||_2) \\ &+ \sigma(\alpha)(||(Q_i^v - K_j^v)||_2 + ||Q_j^v - K_i^v||_2) \end{aligned} \tag{12}$$

Based on the clustering results, we retain the attention computations only between records belonging to same cluster by a masking approach:

$$\widehat{A}_{ij} = \begin{cases} A_{ij}, \text{records i and j belong to the same cluster} \\ -\inf, \text{records i and j belong to different clusters} \end{cases} \tag{13}$$

Using the masked attention matrix $\widehat{A}$, the attended result of the sparse attention $\{\widehat{D}_1, \widehat{D}_2, \cdots, \widehat{D}_n\}$ computed as follows:

$$\widehat{D} = softmax(\frac{\widehat{A}}{\sqrt{d_k}})V. \tag{14}$$

Then, in the cross attention layer, our design is similar to existing Non-local Target Memory Aggregation (NTWA) [6]:

$$Q_i = PW_i^Q, K_i = [\widehat{D}, c]W_i^K, V^c = \widehat{D}W^V \tag{15}$$

$$h_i = softmax(\frac{Q_iK_i^T}{\sqrt{d_k}}), i = 1, 2, ..\widehat{Nh} \tag{16}$$

$$TO_t = cat(h_1, h_2, \cdots h_{\widehat{Nh}})W^OV^c \tag{17}$$

where P is the target's glove [18] embedding , and $\widehat{Nh}$ is the number of heads for multi-head attention. Eventually, the target orientation vector TO_t includes only distinctive clues about the target, enabling localization with greater accuracy.

III. Experiment

A. Experimental Setting

1) Dataset: AI2Thor and RoboTHOR [19]datasets are selected for evaluation. AI2Thor includes 4 types of room: kitchen, living room, bedroom, and bathroom, each consists of 30 floorplans, of which 20 rooms are used for training, 5 rooms for validation, 5 rooms for testing. RoboTHOR consists of 75 scenes, 60 of which are used for training and 15 for testing.

2) Evaluation Metrics: Success rate (SR), success weighted by path length (SPL) [20] metrics are used to evaluate our method. The formula of SR is $SR = \frac{1}{K}\sum_{i=1}^{K} Suc_i$, where K is the number of episodes, and Suc_i indicates whether the i-th episode is successful. SPL indicates the efficiency of the agent, its formula is $SPL = \frac{1}{K}\sum_{i=1}^{K} Suc_i\frac{L_i^*}{\max(L_i, L_i^*)}$, where L_i is the length of the path actually traveled by the agent. L_i^* is the optimal path length.

3) Implementation Details: Our model is trained by 16 workers on 2 RTX 3090 Nvidia GPU for 2.5M episodes. By evaluating on the validation set, the ε_D and ε_O is set as 0.4 and 0.75 repectively. The capacity of target memory buffer is 30. During testing, the max episode length is set to 100. We show results for all targets (ALL) and a subset of targets ($L \geq 5$) whose optimal trajectory length is longer than 5.

TABLE I
COMPARISONS WITH THE STATE-OF-THE-ART METHODS ON THE AI2-THOR/ROBOTHOR DATASETS

Method	ALL (%)		$L \geqslant 5$ (%)	
	SR	SPL	SR	SPL
SP [2](ICLR 2019)	61.8/23.0	37.4/15.2	51.4/19.3	33.2/13.7
SAVN [21](CVPR 2019)	63.2/25.2	36.6/16.2	51.5/20.2	34.2/14.6
ORG [3](ECCV 2020)	67.3/29.1	36.3/18.5	58.1/22.0	35.3/17.4
SA [4](CVPR 2021)	66.1/26.7	38.2/17.1	54.1/20.9	33.8/15.7
HOZ [22](ICCV 2021)	68.2/30.2	36.8/18.4	60.1/23.3	36.2/17.6
VTNet [10](ICLR 2020)	71.8/33.7	45.2/20.0	64.1/29.8	44.1/19.0
DOA [5](MM 2022)	76.5/36.6	43.4/20.7	71.2/31.3	44.6/20.0
DAT [9](ICCV 2023)	80.6/39.3	45.7/25.4	73.7/34.9	46.2/20.1
IOM [6](MM 2023)	81.3/41.6	47.1/28.1	77.3/36.8	48.1/22.2
MT [8](ICML 2023)	82.2/41.5	48.4/29.0	76.4/37.2	48.5/22.7
Ours	**83.9/42.7**	**49.8/29.7**	**77.8/38.5**	**50.0/24.3**

TABLE II
ZERO-SHOT NAVIGATION PERFORMANCE COMPARISON ON THE AI2-THOR DATASETS.

Method	Seen/Unseen split	Unseen Classes			
		ALL (%)		$L \geq 5$ (%)	
		SR↑	SPL↑	SR↑	SPL↑
ZER [23]	18/4	31.0	14.8	25.7	13.9
ZSON [24]	18/4	57.3	20.7	46.4	21.5
MT-ZS	18/4	68.3	27.1	57.4	29.2
Ours-ZS	18/4	**69.8**	**29.3**	**59.0**	**30.9**
ZER	14/8	24.4	10.3	13.9	8.7
ZSON	14/8	52.3	17.9	33.1	14.0
MT-ZS	14/8	62.1	23.8	46.1	23.2
Ours-ZS	14/8	**63.7**	**26.2**	**47.9**	**24.9**

B. Comparisons to the State-of-the-art

As shown in Table I that our method outperforms the previous state-of-the-art (SOTA) method (MT) by 1.7/1.2 and 1.4/0.7 for all test points in SR and SPL (AI2-Thor/RoboTHOR,%). Additionally, under the $L \geq 5$ conditions, our method achieves improvements of 1.5/1.3 and 1.5/1.6 in SR and SPL (AI2-Thor/RoboTHOR,%). Furthermore, our approach demonstrates superiority over other methods employing memory aggregator like DAT [9] and IOM [6].

C. Zero-Shot Performance

As shown in Table II, we also conducted target-agnostic zero-shot object navigation experiments [23]–[25], where unseen objects were neither used as targets nor detectable during training. There are two settings for the number of seen and unseen objects: 18/4 and 14/8, which are similar to the experimental settings of MT-ZS [8]. We also compared our method with the ZER [23] and ZSON [24] models, which are specifically designed for zero-shot navigation. The experimental results demonstrate that our method achieves competitive performance under challenging zero-shot conditions.

D. Universality of Adaptive Strategy Feature Fusion

TABLE III
EXPERIMENTAL RESULTS OF ADAPTIVE STRATEGY FEATURE FUSION APPLYING TO DIFFERENT MODULE COMBINATIONS ON AI2-THOR

Using ASFF	Module Category			ALL (%)		$L \geq 5$ (%)	
	Search	Aggregator	Obstacle	SR	SPL	SR	SPL
✓	VTNet	FTTMA	GM	81.5	47.0	76.9	47.8
	VTNet	FTTMA	GM	80.7	45.8	74.2	45.6
✓	VAG	TAMSA	GM	81.2	47.5	74.6	48.4
	VAG	TAMSA	GM	80.2	46.5	73.7	47.1
✓	VAG	FTTMA	IOM	83.1	49.4	76.9	49.7
	VAG	FTTMA	IOM	81.4	46.8	75.2	46.8

To evaluate the effectiveness of the ASFF on different module combinations, we conducted a series of experiments on AI2Thor [19]. As shown in Table III, we replaced the original search module, target memory aggregator, and obstacle avoidance module with VTNet [10], the target-aware multi-scale aggregator (TAMSA) from DAT [9], and the implicit obstacle map from IOM [6]. These models were tested both with and without ASFF. The experimental results demonstrate that methods incorporating ASFF achieve significant improvements in SR and SPL, highlighting its flexibility and adaptability to various module combinations.

E. Ablation Experiments

TABLE IV
ABLATION EXPERIMENT RESULTS ON AI2-THOR

FTTMA	ASFF		ALL (%)		$L \geq 5$ (%)	
	Dynamic	Learnable	SR	SPL	SR	SPL
			80.3	45.8	73.6	45.9
✓			81.7	46.6	75.8	46.5
	✓		81.1	46.8	74.5	46.3
	✓	✓	81.4	47.7	75.0	48.2
✓	✓	✓	**83.9**	**49.8**	**77.8**	**50.0**

To evaluate the effectiveness of FTTMA and ASFF, we conducted ablation experiments on AI2-Thor, as shown in Table IV.

1) Baseline: Our baseline model uses TAMSA [9] to replace FTTMA. The associative searching vector, target orientation vector, and obstacle distribution vector are directly concatenated and fed into the policy network.

2) Fault Tolerant Target Memory Aggregator: After applying FTTMA to replace TAMSA, the model outperformed the baseline in SR and SPL by 1.4/2.2 and 1.2/0.6 (ALL/L≥5,%), respectively. This indicates that FTTMA enhances the agent's ability to distinguish and locate target objects, particularly in scenes involving long navigation paths.

3) Adaptive Strategy Feature Fusion: To analyze the effectiveness of ASFF, its ablation experiments were divided based on whether the model used the importance indicators of the three branches ("Dynamic") and whether it employed three learnable multipliers ("Learnable") as shown in Table IV. When using only the indicators, the model achieved improvements of 0.8/0.9 and 1.0/0.4 in SR and SPL (ALL/L≥5, %) compared with baseline. After introducing the learnable parameters, the model showed further progress in SPL (0.9/1.9, %), compared to model using only the indicators. It suggest that combining importance indicators and learnable parameters in ASFF enhances adaptability across various situations.

4) Qualitative Analysis: Figure 3 illustrates the testing results of our model compared to baseline in selected scenes. In a kitchen scene, our method successfully located and navigated to the target object, whereas the baseline was misled by visually similar object, resulting in navigation failure. In a living room, our method demonstrated a more effective integration of search, obstacle avoidance, and target locating capabilities, achieving the goal with higher efficiency.

IV. CONCLUSION

In this article, we propose a Adaptive Strategy Feature Fusion (ASFF) for object navigation to enhance the adaptability to dynamic environments. The Fault Tolerant Target

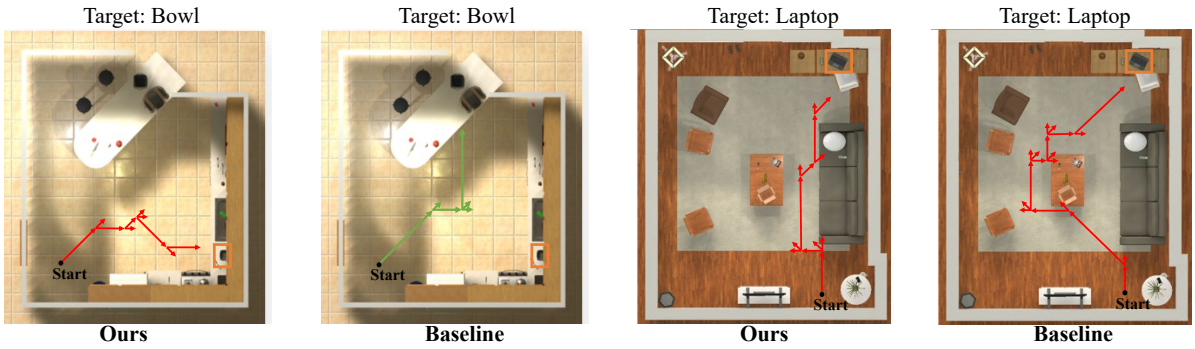

Fig. 3. Visualization of the testing process. The target is selected by the orange box. Red and green trajectories represent success and failure, respectively.

Memory Aggregator (FTTMA) is also introduced, which adopts a sparse mechanism to distinguish and refine target memory precisely and robustly. Experiments on AI2THOR and RoboTHOR demonstrate that our work achieves state-of-the-art performance. In future work, we plan to further investigate the design of feature fusion for improved navigation performance.

V. Acknowledgment

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2022YFF0712100, the National Natural Science Foundation of China under Grant 62202273, Major Basic Research Program of Shandong Provincial Natural Science Foundation under Grant ZR2022ZD02, Joint Key Funds of National Natural Science Foundation of China under Grant U23A20302, and the Key Technology Research and Industrialization Demonstration Projects of Qingdao under Grant 23-1-2-qljh-8-gx.

References

[1] Yuke Zhu, Roozbeh Mottaghi, et al., "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 3357–3364.

[2] Wei Yang, Xiaolong Wang, et al., "Visual semantic navigation using scene priors," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019, OpenReview.net.

[3] H Du, X Yu, et al., "Learning object relation graph and tentative policy for visual navigation," in *European Conference on Computer Vision*, 2020.

[4] Bar Mayo, Tamir Hazan, et al., "Visual navigation with spatial attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16898–16907.

[5] Ronghao Dang, Zhuofan Shi, et al., "Unbiased directed object attention graph for object navigation," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 3617–3627.

[6] Wei Xie, Haobo Jiang, et al., "Implicit obstacle map-driven indoor navigation model for robust obstacle avoidance," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 6785–6793.

[7] Volodymyr Mnih, Adria Puigdomenech Badia, et al., "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[8] Ronghao Dang, Lu Chen, et al., "Multiple thinking achieving meta-ability decoupling for object navigation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 6855–6872.

[9] Ronghao Dang, Liuyi Wang, et al., "Search for or navigate to? dual adaptive thinking for object navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8250–8259.

[10] Heming Du, Xin Yu, et al., "Vtnet: Visual transformer network for object goal navigation," in *International Conference on Learning Representations*, 2020.

[11] Haokuan Luo, Albert Yue, et al., "Stubborn: A strong baseline for indoor object navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 3287–3293.

[12] Shaoqing Ren, Kaiming He, et al., "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[13] Nicolas Carion, Francisco Massa, et al., "End-to-end object detection with transformers," in *European Conference on Computer Vision*, 2020, pp. 213–229.

[14] Tun Wang et al., "Light field depth estimation: A comprehensive survey from principles to future," *High-Confidence Computing*, vol. 4, no. 1, pp. 100187, 2024.

[15] Aurko Roy, Mohammad Saffar, et al., "Efficient content-based sparse attention with routing transformers," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 53–68, 2021.

[16] Martin Ester, Hans-Peter Kriegel, et al., "Density-based spatial clustering of applications with noise," in *Int. Conf. knowledge discovery and data mining*, 1996, vol. 240.

[17] Kaiming He, Xiangyu Zhang, et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[18] Jeffrey Pennington, Richard Socher, et al., "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[19] Eric Kolve, Roozbeh Mottaghi, et al., "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[20] Peter Anderson, Angel Chang, et al., "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[21] Mitchell Wortsman, Kiana Ehsani, et al., "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6750–6759.

[22] Sixian Zhang, Xinhang Song, et al., "Hierarchical object-to-zone graph for object navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15130–15140.

[23] Apoorv Khandelwal et al., "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14829–14838.

[24] Qianfan Zhao, Lu Zhang, et al., "Zero-shot object goal visual navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2025–2031.

[25] Dingbang Li, Wenzhou Chen, et al., "Tina: Think, interaction, and action framework for zero-shot vision language navigation," in *IEEE International Conference on Multimedia and Expo, ICME 2024, Niagara Falls, ON, Canada, July 15-19, 2024*. 2024, pp. 1–6, IEEE.