

Adaptive Multi-Modal Visual Navigation via Task-Driven Transformer with Reinforcement Learning

Libo Gao

*School of Electronic Engineering, Xidian University, Xi'an, China
18241775873@163.com*

Abstract. Aiming at the problems of insufficient fusion of multi-modal visual navigation information, weak target constraint, and easy influence of action decision by partial occlusion in complex scenes, an adaptive navigation method combining task-driven Transformer and reinforcement learning is constructed. The model takes RGB images, depth maps, semantic labels and object vectors as input, forms a unified Token sequence through modal embedding and spatial position encoding, uses task tokens to guide cross-modal attention allocation, and combines local scene memory to update continuous navigation states. The reinforcement learning strategy adds safety constraints on the basis of the action probability output to complete the adjustment decisions of forward, turning, stopping and obstacle avoidance. The experiment constructs 1280 valid trajectories based on 36 indoor navigation scenes, forming 92400 groups of state transition samples. In the test, the model success rate reaches 94.8%, the SPL reaches 0.801, and the collision rate is reduced to 3.7%. In 12 untrained complex scenes, the success rate maintains 91.6%, which indicates that the proposed method can improve the path efficiency, safety and stability of visual navigation.

Keywords: Multi-modal visual navigation, Task-driven Transformer, Reinforcement learning, Adaptive decision making

1. Introduction

Multimodal visual navigation requires agents to use visual, depth and semantic cues for target search, obstacle avoidance and path selection. Single-image features cannot stably represent obstacles, target positions and traversable areas, and unconstrained multimodal fusion easily introduces redundancy [1]. To solve insufficient information utilization, delayed state update and weak decision adaptability, a task-driven Transformer with reinforcement learning is constructed to guide cross-modal attention, update navigation states and output actions for efficiency, safety and generalization evaluation.

2. Multimodal visual navigation tasks and data modeling

2.1. Navigation scene, target state, and multimodal input construction

At each time step, RGB images, depth information, semantic labels, object vectors and local position states are integrated as unified input after size alignment, feature normalization and timestamp matching. The multimodal state vector is defined as follows.

$$s_t = [f_{rgb,t}; f_{dep,t}; f_{sem,t}; g_t; p_t] \quad (1)$$

Where, s_t represents the navigation state vector at time t , $f_{rgb,t}$, $f_{dep,t}$, $f_{sem,t}$ represents RGB, depth, and semantic features respectively, g_t represents the target direction vector, and p_t represents the local position state of the agent. The multi-modal input is organized according to the source of collection, the meaning of features and the purpose of modeling, and the specific composition is shown in Table 1.

Table 1. Multimodal input data and modeling purposes

Input Type	Feature Content	Modeling Use
RGB image	Scene texture, obstacle contours, target appearance	Extract visual spatial features
Depth information	Obstacle distance, traversable space, local geometric structure	Determine feasible areas and collision risks
Semantic labels	Doors, walls, passages, target categories	Enhance scene understanding ability
Target vector	Target direction and relative distance	Guide the task token to focus on the target area
Local position state	Current coordinates, orientation, historical actions	Support state update and action decision-making

2.2. Trajectory samples, action labels, and state transition modeling

Trajectory samples are divided into state-action-next state records for policy training. Action labels include forward, left turn, right turn, stop and obstacle avoidance adjustment, and are generated by position, target distance change and collision detection results [2]. Reward feedback and termination marks are recorded as follows.

$$D = \{(s_t, a_t, r_t, s_{t+1}, d_t)\}_{t=1}^N \quad (2)$$

Where, D represents the set of trajectory samples, s_t represents the current state, a_t represents the execution action, r_t represents the reward feedback, s_{t+1} represents the state at the next moment, d_t represents whether the trajectory terminates, and N represents the total number of samples. This modeling method can integrate visual input, action result and navigation feedback into the same training sample.

3. Task-driven Transformer feature fusion model

3.1. Multimodal input embedding and spatial location encoding

Before entering the Transformer, RGB images, depth maps and semantic labels are converted into uniform-dimensional Tokens. Lightweight convolutional encoders extract local features, and linear mapping projects them into the same feature space to support attention calculation [3]. The modal embedding is calculated as follows.

$$e_{m,t} = \phi_m(x_{m,t}) = W_m x_{m,t} + b_m, m \in \{rgb, dep, sem\} \quad (3)$$

Where, $x_{m,t}$ represents the MTH modal input at time t , $\phi_m(\cdot)$ represents the modal encoding function, W_m and b_m represent the learnable mapping parameters, $e_{m,t}$ represent the mapped modal Token. In order to preserve the relative position relationship of obstacles, target area and traffic space, the two-dimensional spatial position encoding is superimposed into the modal Token:

$$z_{m,t}^0 = e_{m,t} + p_{i,j} + q_m \quad (4)$$

Where, $z_{m,t}^0$ represents the Transformer input Token, $p_{i,j}$ represents the spatial position encoding of row i and column j , and q_m represents the modal type encoding. This processing enables subsequent attention layers to simultaneously identify feature sources and spatial locations. See Figure 1 for the overall structure of the computational relationship between multimodal input embedding, positional encoding, and task-guided fusion.

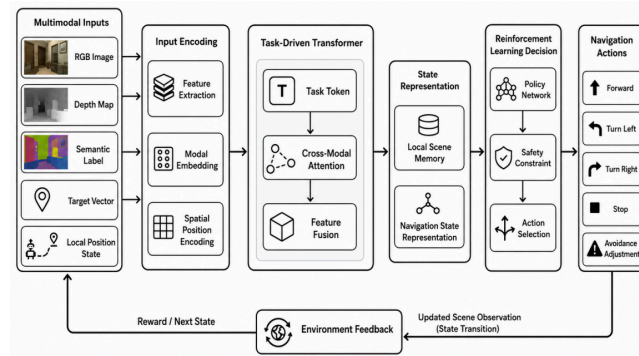


Figure 1. Overall framework diagram of multimodal visual navigation

3.2. Task Token-guided cross-modal attention fusion

The task Token represents the navigation target, direction and distance change, and serves as the query vector in cross-modal attention. RGB, depth and semantic Tokens provide appearance, geometry and category cues, while the attention layer assigns fusion weights to target areas, nearby obstacles and passable spaces. Task-guided attention weights are calculated as follows.

$$\alpha_{m,t} = \text{softmax} \left(\frac{(W_q u_t)(W_k z_{m,t}^0)^T}{\sqrt{d}} \right) \quad (5)$$

Where, $\alpha_{m,t}$ represents the attention weight of the m -th modality at time t , u_t represents the task Token, W_q and W_k represent the query and key mapping matrix, and d represents the feature dimension. Multi-modal tokens are aggregated according to their weights to obtain task-related fusion features:

$$h_t = \sum_{m \in \{\text{rgb}, \text{dep}, \text{sem}\}} \alpha_{m,t} W_v z_{m,t}^0 \quad (6)$$

Where, h_t represents the cross-modal fusion feature and W_v represents the value mapping matrix. This feature can weaken irrelevant background interference and provide task constraints for subsequent navigation state representation.

3.3. Local scene memory and navigation state representation

Cross-modal fusion features mainly describe the current frame and have limited ability to represent passed obstacles, turning actions and target approach trends [4]. A local scene memory unit is introduced to write the current fusion feature, previous action and local position state into the memory vector through gated updating. The navigation state representation is calculated as follows.

$$\begin{aligned} m_t &= \text{GRU}(m_{t-1}, [h_t; a_{t-1}; p_t]) \\ o_t &= \sigma(W_s [h_t; m_t; g_t] + b_s) \end{aligned} \quad (7)$$

Where, m_t represents the local scene memory at time t , m_{t-1} represents the memory state at the last time, a_{t-1} represents the action at the last time, o_t represents the final navigation state representation, $\sigma(\cdot)$ represents the nonlinear activation function, W_s and b_s represent the state mapping parameters. This representation provides a stable state input for the reinforcement learning policy.

4. Reinforcement learning navigation strategy and adaptive decision-making

4.1. Action space, reward function, and policy optimization objective

The policy network takes o_t as input and outputs action probabilities for forward, left turn, right turn, stop and obstacle avoidance adjustment. The reward function combines goal distance reduction, path efficiency, collision penalty and arrival reward, and is defined as follows.

$$r_t = \lambda_1 \Delta d_t - \lambda_2 c_t - \lambda_3 l_t + \lambda_4 q_t \quad (8)$$

Where, r_t represents the reward value at time t , Δd_t represents the target distance reduction, c_t represents the collision penalty term, l_t represents the path length cost, q_t represents the target arrival

reward, and $\lambda_1, \lambda_2, \lambda_3$, and λ_4 represent the weight coefficients. The policy network updates the parameters by maximizing the cumulative discount reward, and the optimization objective is defined as follows.

$$J(\theta) = E_{\pi_\theta} \left[\sum_{t=0}^T \gamma^t r_t \right] \quad (9)$$

Where, $J(\theta)$ represents the strategy optimization objective, π_θ represents the action strategy with parameter θ , γ represents the discount factor, and T represents the length of a single navigation trajectory. This objective unifies the immediate obstacle avoidance feedback with the long-range reach gain into the policy update.

4.2. Adaptive action selection under safety constraints

Safety constraints filter policy actions that violate obstacle distance or goal direction limits. Action selection under safety constraints is defined as follows.

$$a_t^* = \underset{a_t \in A_s}{\operatorname{argmax}} \pi_\theta(a_t | o_t), A_s = \{a_t | D(a_t) \geq D_{\text{safe}}, E(a_t) \leq E_{\text{max}}\} \quad (10)$$

Where, a_t^* represents the final action executed, A_s represents the set of actions satisfying the safety constraints, $\pi_\theta(a_t | o_t)$ represents the action probability output by the policy network, $D(a_t)$ represents the predicted obstacle distance after executing the action, D_{safe} represents the minimum safety distance, $E(a_t)$ represents the cost of the action deviating from the target direction, and E_{max} represents the upper limit of allowable deviation. A closed-loop decision process is formed among the policy network, security constraints, and action execution, as detailed in Figure 2.

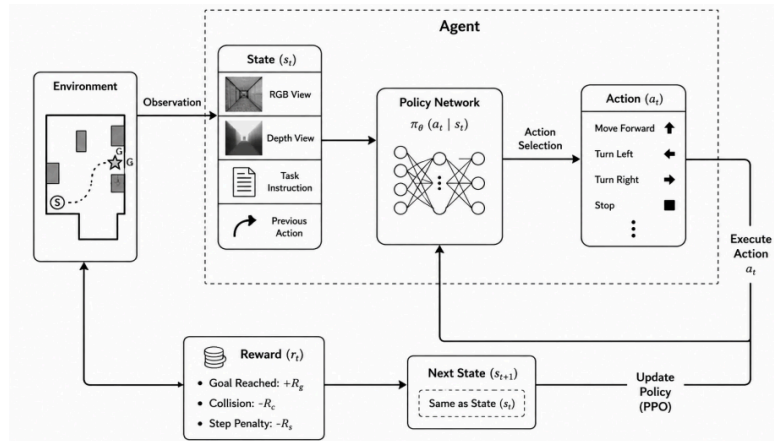


Figure 2. Flow chart of navigation decision in reinforcement learning

5. Experimental design and result analysis

5.1. Experimental platform, data set and evaluation index

The experiment was conducted under a unified software and hardware environment. The multimodal input included RGB images, depth maps, semantic labels and object vectors. The dataset was collected from 36 indoor navigation scenarios, covering corridors, rooms, doorway steering, narrow passages and obstacle detours. A total of 1,280 valid trajectories and 92,400 state transition samples were generated. The experimental platform, data split and evaluation metrics are shown in Table 2.

Table 2. Experimental platform, dataset and parameter settings

Category	Configuration Item	Parameter Setting
Hardware	CPU / GPU	Xeon Silver 4314 ×2 / RTX 4090 24GB
Software	OS / Framework	Ubuntu 22.04 / Python 3.10 / PyTorch 2.2
Scene Data	Indoor Scenes	36 indoor navigation scenes
Trajectory Data	Valid Trajectories	1,280 valid navigation trajectories
Sample Data	State Transition Samples	92,400 groups
Data Split	Train / Val / Test	64,680 / 13,860 / 13,860
Input Size	RGB / Depth / Semantic	224×224 / 224×224 / 224×224
Training	Optimizer / Batch / Epoch	AdamW / 32 / 100
Evaluation	Metrics	Success Rate, SPL, Path Length, Collision Rate, Latency

5.2. Performance comparison of different visual navigation algorithms

To verify the effectiveness of the joint task-driven Transformer and reinforcement learning model, A*, DQN, PPO and ordinary Transformer were selected as comparison methods on 192 complete test trajectories [5]. A* used the same scene raster map, DQN and PPO used the same state input, and ordinary Transformer removed task Token guidance while retaining multimodal feature concatenation. The comparison results are shown in Table 3.

Table 3. Performance comparison of different visual navigation algorithms

Algorithm	Success Rate/%	SPL	Average Path Length/m	Collision Rate/%	Latency/ms
A*	82.3	0.641	18.7	12.5	31.4
DQN	86.5	0.684	17.9	9.8	24.6
PPO	89.1	0.713	17.2	8.1	27.3
Transformer	91.2	0.742	16.6	6.9	29.8
Proposed Method	94.8	0.801	15.4	3.7	32.6

The proposed method achieves the best overall performance, with a success rate of 94.8%, an SPL of 0.801, an average path length of 15.4 m and a collision rate of 3.7%, showing better path efficiency and safety than the comparison methods.

5.3. Ablation experiments and generalization analysis of complex scenes

The ablation experiment was conducted on 192 complete test trajectories. The full model was used as the benchmark, and variants removing the task Token, local scene memory and safety constraints were set together with an RGB-only input version. All groups kept the same training rounds, optimizer, input size and evaluation metrics. For generalization testing, 12 unseen indoor scenes, including narrow passages, target occlusion, continuous corners and dense obstacles, were used to generate 240 trajectories. The results are shown in Figure 3.

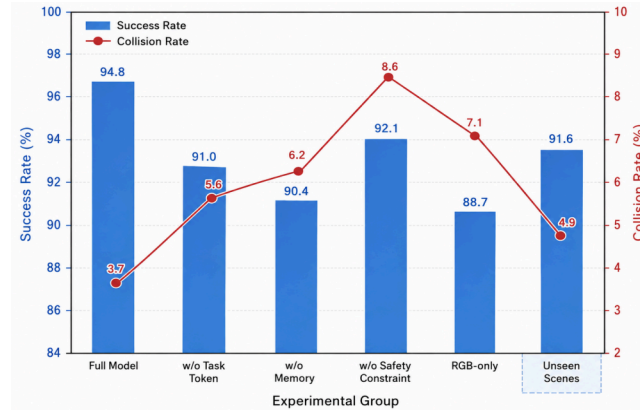


Figure 3. Comparison of ablation experiments and generalization of complex scenes

The full model reaches 94.8% success rate, 0.801 SPL and 3.7% collision rate. Removing the task Token, local scene memory and safety constraints reduces the success rate to 91.0%, 90.4% and 92.1%, while the collision rate without safety constraints rises to 8.6%. The RGB-only version drops to 88.7% success rate and 0.702 SPL. In unseen scenes, the model still maintains 91.6% success rate and 4.9% collision rate.

6. Conclusions

This paper constructs a joint task-driven Transformer and reinforcement learning model for goal-guided fusion and safe action decision in multimodal visual navigation. Task Token guidance enhances responses to target areas and passable spaces, local scene memory improves state stability in corners and occluded scenes, and safety constraints reduce collision risk. Compared with ordinary Transformer, the proposed method increases success rate from 91.2% to 94.8%, improves SPL from 0.742 to 0.801, and reduces collision rate from 6.9% to 3.7%. Removing safety constraints raises collision rate to 8.6%, while the unseen-scene success rate remains 91.6%. Future work will consider dynamic obstacle prediction and lightweight deployment to support online navigation in real scenarios.

References

- [1] Zhou K, Zhang H, Li F. TransNav: spatial sequential transformer network for visual navigation [J]. Journal of Computational Design and Engineering, 2022, 9(5): 1866-1878.
- [2] Wu Q, Wang J, Liang J, et al. Image-goal navigation in complex environments via modular learning [J]. IEEE Robotics and Automation Letters, 2022, 7(3): 6902-6909.
- [3] Li W, Hong R, Shen J, et al. Transformer memory for interactive visual navigation in cluttered environments [J]. IEEE Robotics and Automation Letters, 2023, 8(3): 1731-1738.

- [4] Zeng H, Song X, Jiang S. Multi-object navigation using potential target position policy function [J]. IEEE Transactions on Image Processing, 2023, 32: 2608-2619.
- [5] Chen B, Kang J, Zhong P, et al. Think holistically, act down-to-earth: a semantic navigation strategy with continuous environmental representation and multi-step forward planning [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2024, 34(5): 3860-3875.