2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

ST-DiffPlanner: A Safety-Enhanced Topology-Aware Diffusion Planner for Global Path Planning

Jiaquan Yan[1], Fang Zhao[1], Huiyu Yuan[1], Yushi Chen[1], Long Wang[1], Dan Luo[2], Haiyong Luo*[3]

***Abstract*— In complex environments, traditional path planning methods rely on manually defined models, requiring tedious adjustments under varying scenarios or constraints. They also suffer from unstable time overhead and exponentially increasing computational costs as environmental complexity grows. Deep learning-enhanced methods, while optimizing decisions via neural networks, remain constrained by explicit search/sampling frameworks—this leads to unstable real-time performance and failure to capture real-world trajectory distributions. In contrast, diffusion-based planning directly learns trajectory distributions from data, offering predictable inference latency via fixed inversion steps and inherent support for multimodal solutions. However, its lack of explicit safety constraints often leads to trajectory safety issues, resulting in planning failures. To address these limitations, this paper proposes ST-DiffPlanner, a global path planner following the pipeline of "topology cognition—direction focusing—trajectory generation". It introduces three targeted optimizations: (1) leveraging topological awareness to constrain the diffusion model to focus on collision-free regions; (2) optimizing inference-phase projection to ensure trajectory continuity and safe distances from obstacles; (3) designing a topology anchor-based safety loss to enhance model safety and training stability. Experimental results demonstrate that ST-DiffPlanner exhibits strong generalization across multiple scenarios and modalities, accurately capturing environmental features and learning task-compliant trajectory characteristics. Our method achieves an average trajectory generation success rate of 96.9%, significantly outperforming baseline methods. Moreover, validation in both simulated and real-world robot platforms confirms its applicability across different systems.**

I. INTRODUCTION

In industrial logistics, household services, emergency rescue, and other scenarios, robots often operate in narrow passages and maze-like structures—such as shelf aisles and underground parking lots—with limited safety margins. Planning systems in these scenarios must simultaneously meet multiple critical requirements: strict safety constraints, stable real-time performance, and cross-scenario generalization.

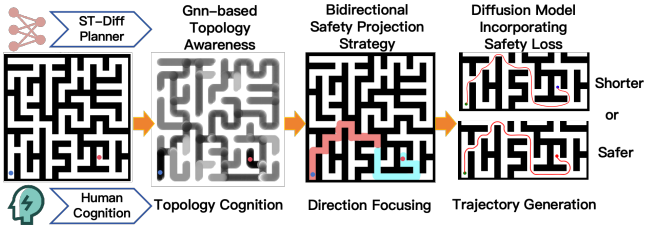


Fig. 1. The core idea of ST-DiffPlanner: Inspired by human cognitive logic for trajectory planning in complex environments.

Traditional methods based on A* or RRT* [1], [2] exhibit reasonable optimality or convergence in low-dimensional static environments but rely on manually defined cost functions, heuristics, and collision models. When task requirements or constraints change, they require tedious parameter retuning and constraint restructuring; in complex maze-like scenarios, search scales escalate exponentially with environmental complexity, directly causing fluctuations in real-time performance and a significant drop in success rates.

Learning-enhanced approaches enhance the learned heuristics [3], [4] or sampling biases [5], [6] of A*/RRT* via neural networks to improve efficiency, yet remain constrained by explicit search/sampling frameworks. Their performance is limited by manual designs, with time overhead fluctuating with environmental complexity and map scale.

In recent years, diffusion-based generation has emerged as a promising paradigm for path planning [7]–[12]. These methods directly learn trajectory distributions from data, achieve predictable time costs via fixed inversion steps, and inherently support multimodal solutions—making them suitable for rapidly generating diverse feasible paths across scenarios. Unlike most existing works, which are confined to scenario-specific planning [7], [9], [10] or focus solely on local planning [11], [12], DiPPeR [13] stands out by enabling cross-scenario global planning in complex mazes through visual occupancy features. Nevertheless, DiPPeR and similar diffusion-based methods suffer from a critical flaw: the generated trajectory space lacks explicit safety constraints, often leading to obstacle penetration. While some studies [14], [15] attempt to enhance safety via projection or optimization guidance during denoising, their effectiveness is limited in dense maze-like environments. Additionally, certain training-phase safety losses [16] only work for sparse obstacles and fail to converge stably in complex settings. Thus, enhancing the safety of diffusion-based global planning in intricate mazes remains a pressing and unresolved challenge.

*This work was supported in part by the National Natural Science Foundation of China under Grant 62261042, the Key Research Projects of the Joint Research Fund for Beijing Natural Science Foundation and the Fengtai Rail Transit Frontier Research Joint Fund under Grant L221003, Beijing Natural Science Foundation under Grant 4232035 and 4254084, the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA28040500, Yibin high-level Introduction talents project under Grant 2024YG01 and the China Postdoctoral Science Foundation under Grant 2024M750200(Corresponding author: Haiyong Luo).

[1]Jiaquan Yan, Fang Zhao, Huiyu Yuan, Yushi Chen, Long Wang are with Beijing University of Posts and Telecommunications, Beijing, China {yanjiaquan, zfsse, 2024110771, chenyushi, 92536077}@bupt.edu.cn

[2]Dan Luo is with Beijing Forestry University, Beijing, China dluo_work@bjfu.edu.cn

[3]Haiyong Luo is with Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China yhluo@ict.ac.cn

To address the limitations of existing path planning methods mentioned above, this paper proposes a novel diffusion-based trajectory planning approach, Safety-Enhanced Topology-Aware Diffusion Planner (ST-DiffPlanner). Inspired by human path planning cognition in complex environments, as illustrated in Fig. 1, the design of ST-DiffPlanner follows three key steps that mirror such cognitive processes: spatial perception, direction focusing, and trajectory generation.

First, humans rapidly perceive topological spaces to identify potential feasible regions. Correspondingly, we introduce a GNN-based topological awareness module to estimate plannable areas, reducing the model's focus on low-value regions (e.g., dead ends or passages that clearly do not lead to the destination) and thereby improving global reachability and cross-scenario robustness.

Second, humans efficiently determine planning directions via bidirectional search, a strategy validated by multiple studies [17]. Drawing on this, we design a topology-aware bidirectional safety projection strategy to focus trajectory generation directions, ensuring coherence and controllability in narrow, tortuous passages.

Finally, humans generate task-compliant trajectories based on the aforementioned knowledge. In line with this, we construct a differentiable safety field and corresponding safety loss using topology anchors; the violation weighting mechanism significantly reduces collision rates while stabilizing training, enabling the model to learn trajectory characteristics alongside safety awareness.

The main contributions of this paper are as follows:

- We propose a GNN-integrated topology-aware diffusion framework, where GNN endows the diffusion model with global spatial understanding to address its topological cognition deficiency.
- We design a topology anchor-based differentiable safety loss to enhance the model's ability to recognize safety boundaries during training.
- We develop a topology-aware bidirectional safety projection strategy to ensure the safety of generated trajectories during inference.
- Experiments validate that ST-DiffPlanner exhibits excellent generalization across multiple scenarios and modalities, with a best average trajectory success rate of 96.9% (far exceeding baselines). Its platform invariance is verified on both simulated and real-world platforms.

II. RELATED WORK

Trajectory planning, a core module of robotic autonomous navigation, is primarily categorized into three branches: Classical Path Planning, Learning-Enhanced Path Planning, and Diffusion-Based Path Planning.

A. Classical Path Planning

Classical methods such as A* and RRT* have significant drawbacks in adaptability and efficiency: they overly rely on manually predefined strategies and require extensive invalid exploration in complex environments (e.g., dead ends, narrow passages). Although existing improvements [1], [2] attempt to mitigate these issues by designing specific rules, their inherent limitations persist. Furthermore, different trajectory planning tasks impose diverse requirements, such as right-lane driving, generating safer trajectories, or producing shorter trajectories. To address these demands, classical methods often require function redesign, yet manual design may fail to capture task-specific complexities.

B. Learning-Enhanced Path Planning

Learning-enhanced path planning methods leverage neural networks to enhance the exploration efficiency of traditional approaches. For instance, NIRRT* [5] and [6] improve efficiency by limiting search ranges, while N-A* [3] and ViT-A* [4] utilize models for intelligent decision-making. However, these methods are still confined to traditional frameworks: in maze scenarios with numerous obstacles, they still exhibit large variations in time overhead.

C. Diffusion-Based Path Planning

Diffusion models have emerged as a research hotspot in recent years due to their unique advantages: they can directly learn the distribution characteristics of ground-truth trajectories to adapt to diverse demands, and their fixed diffusion steps ensure stable time overhead. Nevertheless, a critical flaw of diffusion models is that the generated trajectory space lacks explicit safety constraints, leading to unguaranteed safety—particularly in complex scenarios.

Some existing methods [7]–[12] build unified frameworks for multi-task motion planning, which work for devices like manipulators [8], [9] and cover Maze2D navigation tasks [7], [10]. However, these methods are often trained for specific scenarios and tasks, leading to poor generalization in general path planning. Methods such as those proposed in [11], [12] utilize obstacle environments as conditional inputs for navigation. However, their emphasis on local planning renders them susceptible to local optima in the absence of global path guidance. Notably, DiPPeR [13] achieves generalization and enables global planning in complex mazes. However, due to diffusion models' inherent limits, it has safety defects and cannot guarantee trajectory safety.

Constrained diffusion planners [14], [15] attempt to enhance safety via projection or optimization strategies during denoising—such as safety optimization based on Control Barrier Functions [14] and trajectory projection optimization [15]. But in global planning for complex mazes, they still cannot simultaneously ensure trajectory safety and continuity. [16] introduces safety loss during training, but it is only valid for simple scenarios and unstable in maze convergence.

To address these issues, this paper proposes a universal global diffusion planner that enhances the safety of global trajectory planning during both training and inference phases via a GNN-based topological awareness mechanism.

III. PRELIMINARIES

A. Diffusion Model

Diffusion models [18] are generative models based on "progressive noising-reverse denoising". They recover target-

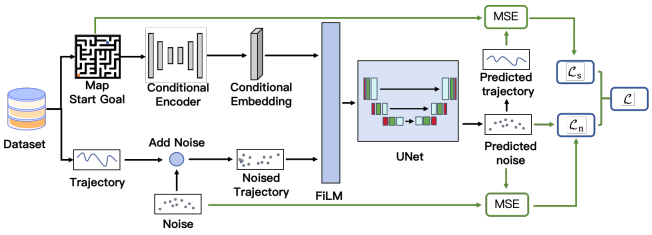


Fig. 2. The Training Pipeline. ST-DiffPlanner's training builds on DiPPeR's image-conditioned diffusion pipeline, adding a GNN-based topological perception encoder (integrated into the conditional encoder) and computing safety loss from observations after trajectory generation.

distribution samples from random noise via a Markov chain $\{x_t\}_{t=1}^{T}$, with samples denoted as x_0.

Forward noising process. The original sample x_0 is progressively transformed into Gaussian noise through $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$, where $\beta_t \in (0,1)$ denotes the noise coefficient that increases with the step t. The noise addition formula for x_t is:

$$\mathcal{X}_t(x_0, \epsilon) = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, \tag{1}$$

where $\epsilon \sim N(0, I)$ is the noise term, and $\alpha_t = \prod_{i=1}^{t}(1-\beta_i)$.

The denoising network $\epsilon_\theta(X_t, t)$ is optimized to minimize the mean squared error (MSE) between the true noise ϵ and the network-predicted noise, with the objective function:

$$\min_\theta \mathbb{E}_{t\sim[1,T],p(x_0),N(\epsilon;0,I)}\left[\|\epsilon - \epsilon_\theta(X_t(x_0,\epsilon),t)\|^2\right]. \tag{2}$$

Reverse denoising process. Denoising is performed iteratively using the trained network, following:

$$x_{t-1} = D(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t \cdot \epsilon, \tag{3}$$

where $\overline{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, and σ_t is the noise standard deviation of the reverse process.

B. Project Diffusion Model

Traditional diffusion models often struggle to satisfy predefined constraints. To address this limitation, PDM (Projected Diffusion Model) [15] introduces an iterative projection strategy into the standard diffusion framework, thereby integrating constraints into the reverse denoising process.

The projection operation is implemented in the form of a constrained optimization problem:

$$P_C(x) = \arg\min_{y\in C}\|y - x\|_2^2, \tag{4}$$

where C is the constraint-satisfying space, and $\|y-x\|_2^2$ is the distance from x to the nearest feasible y. With projection, the reverse step becomes $x_{t-1} = P_C(D(x_t, t))$, where P_C denotes projection onto constraint set C.

IV. METHOD

Our method draws inspiration from baseline studies [13], [15], [16]. We reproduced the DiPPeR framework and integrated a topology-aware conditional input based on Graph Neural Networks (GNNs). During training, a differentiable safety loss is designed to ensure the safety of generated trajectories, and the strategy from [15] is adopted to stabilize training loss. In the inference phase, this study incorporates the projection strategy from [16] and redesigns a trajectory planning-specific projection mechanism tailored to complex maze environments, further enhancing trajectory safety.

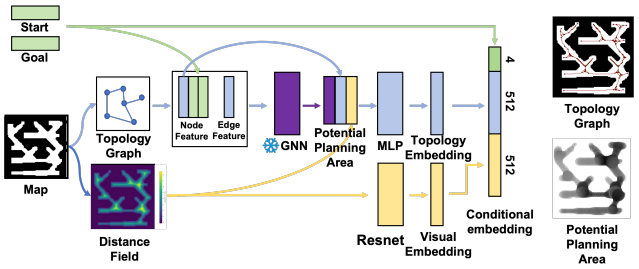
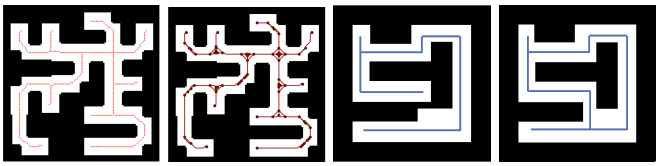


Fig. 3. The conditional encoder module includes two parts: ResNet for visual encoding of distance fields, GNN (Gated-GCN) for topological encoding of topological graphs. These, combined with start and goal coordinates, form the conditional encoding. On the right are examples of topological graphs and potential planning areas (circles centered at topological nodes, with radii from distance field values and colors indicating GNN-output interest scores).



(a) Dense topological graph (b) Sparse topological graph (c) Observation example 1 (d) Observation example 2

Fig. 4. Generation of topological graphs and observation examples. (a) Original dense topological graph generated via morphological skeletonization; (b) Sparse topology nodes obtained after keypoint extraction. (c)(d) show observation examples with similar image features but distinctly different topological structures (blue lines represent topological graphs).

A. Framework

Our diffusion training framework is illustrated in Fig. 2. Inputs (occupancy map, start/end point coordinates) are first encoded by a conditional encoder, with the embedding guiding the diffusion model's generation. For trajectory processing, ground-truth trajectories are noised to obtain noisy trajectories, which are fused with conditional embedding via Feature-wise Linear Modulation (FiLM) [19]. The fused feature is fed into the U-Net module, outputting predicted noise $\epsilon_\theta(\mathcal{X}_t(O, x_0, \epsilon), t)$. MSE loss is computed by comparing predicted noise and ground-truth noise. After denoising, the predicted trajectory is used with environmental information to calculate safety loss. All data samples X_t^k are 2D trajectories, where $k \in (0, K)$ denotes the trajectory sequence time step and $t \in (0, T)$ represents the diffusion process time step.

B. Conditional Encoder

As shown in Fig. 3, inputs for conditional feature extraction include start/end point coordinates and the observed

image O. From O, two representations are derived: the distance field image D and the topological graph G. Feature embeddings are extracted from D and G via a Visual Encoder and a Topology Encoder, respectively.

Topological Graph Generation. Morphological skeletonization [20] is applied to extract a single-pixel-wide skeleton structure while preserving the topological properties of the original observation. Edges are then constructed between adjacent skeleton points to form a dense topological graph. Subsequently, key points are filtered: only those with a degree not equal to 2 are retained, and additional key points are uniformly sampled along long edges to generate a sparse topological graph, as illustrated in Fig. 4a and Fig. 4b.

Visual Encoder. A ResNet-18 visual encoder with spatial softmax pooling is trained to convert D into a latent embedding d while preserving spatial information. It is trained with framework components except the topology encoder.

Topology Encoder. The Gated Graph Convolutional Network (Gated-GCN) serves as the Topology Encoder, taking node features (including coordinates and start/end information) and edge features (corresponding to connection distances) from G as input. It outputs edge interest scores, which are converted into node interest scores for the dense topological graph. Node coordinates, interest scores, and their corresponding values in the distance field D form intermediate results (topology circles), which are mapped to the latent embedding g via a fully connected MLP layer. Here, the distance field value of each node roughly indicates the size of its surrounding safe region, aiding subsequent safety judgments.

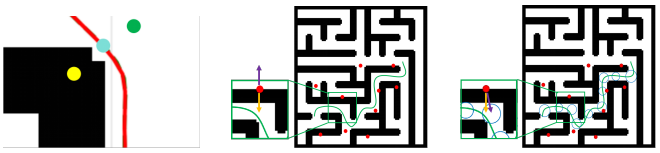
The Gated-GCN is pre-trained using ground-truth trajectories. During pre-training, the task is formulated as an edge classification problem: edges lying on the ground-truth trajectory are labeled 1, and all other edges are labeled 0. The model outputs a normalized probability difference, which serves as the interest score.

Why Use Topology Features? Visual features can capture global patterns but have weak ability to distinguish topological structures. As shown in Fig. 4c and Fig. 4d, minor pixel changes in similar images may lead to significantly different topological structures. In contrast, graph-convolved topological features can effectively distinguish such differences, making them more suitable for path planning scenarios.

Why Not Use the Shortest Path on the Topological Graph? The topological graph is an abstract representation of space, and its shortest path does not necessarily correspond to the optimal path in physical space. Instead, the Topology Encoder identifies multiple candidate paths by outputting interest scores for trajectory segments, which preserves data-driven learning capabilities while aligning with the diffusion model's goal of learning latent properties of ground-truth trajectory distributions.

C. Diffusion Training

The training process of diffusion follows the basic workflow of Denoising Diffusion Probabilistic Models (DDPMs). A noisy trajectory is generated from x_0 through diffusion



(a) $\mathcal{L}_n$ of yellow and green dots is similar, but safety loss is different.

(b) The gradient directions of $\mathcal{L}_{sp}$ and $\mathcal{L}_n$ are often in conflict.

(c) The gradient directions of $\mathcal{L}_{sc}$ and $\mathcal{L}_n$ are roughly aligned.

Fig. 5. Schematic of safety loss. (a) The yellow and green dots are two trajectory points, and the blue dot is the ground-truth reference point. (b) and (c) Red dots: input noisy trajectory. Purple arrows: gradient direction of safety loss. Yellow arrows: gradient direction of MSE loss. Green line: ground-truth trajectory. Blue circles: topology circles. The radius value is the value of the distance field corresponding to the topological point.

according to eq. (1) with noise ϵ. The network predicts noise $\hat{\epsilon}$, and the MSE loss $\mathcal{L}_n$ is calculated between the predicted noise $\hat{\epsilon}$ and the true Gaussian noise ϵ, $\mathcal{L}_n$ is defined as:

$$\mathcal{L}_n = \text{MSE}(\epsilon_t, \epsilon_\theta(X_t(O, x_0, \epsilon), t)), \tag{5}$$

where ϵ_t is the noise corresponding to x_t, and t is the diffusion step. As shown in Fig. 5a, while MSE loss based on ground-truth trajectories enables the model to learn trajectory generation patterns in specific environments, it fails to endow the model with an understanding of "safety". Thus, a safety loss is needed to enhance the model's safety awareness.

Existing safety losses (e.g., $\mathcal{L}_{sp} = \text{MSE}(\hat{x}_t^k, x_{t,sp}^k)$, where $x_{t,sp}^k$ is nearest safe point) are only effective for sparse obstacles. In complex scenarios like mazes, as shown in Fig. 5b and Fig. 5c, their gradients easily conflict with $\mathcal{L}_n$, leading to failure in training convergence.

Inspired by the "safety corridor" concept in trajectory optimization, we design a safety corridor loss $\mathcal{L}_{sc}$ derived from ground-truth trajectories, aligning its convergence direction with that of MSE loss:

$$\mathcal{L}_{sc} = \text{MSE}(\max(|l_k - \hat{x}_t^k|, 0) + \max(|\hat{x}_t^k - u_k|, 0), 0), \tag{6}$$

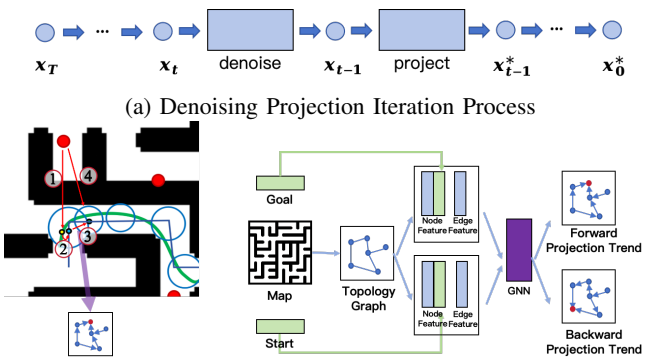
where u_k and l_k denote the upper/lower bounds of the k-th point in the safety corridor, and $\hat{x}_t^k$ is the k-th point of the denoised trajectory at diffusion step t.

However, safety corridors lack a direct connection with image inputs, making it difficult for the model to learn their mapping relationship and increasing training complexity. To address this, we use topology circles as safety corridors to link the safety loss with high-interest topology circle information in topology embeddings, designing a topology anchor-based safety loss $\mathcal{L}_s$:

$$\mathcal{L}_s = \text{MSE}(\max(||\hat{x}_t^k - s_k||_2 - r_k, 0)), \tag{7}$$

where s_k and r_k are the center coordinate and safety radius of topology circle, which is obtained by projecting the ground-truth trajectory point onto the dense topological graph.

Notably, the value of $\mathcal{L}_s$ fluctuates significantly with diffusion step t: when t is small, $\mathcal{L}_s$ is too small to be effective; when t is large, $\mathcal{L}_s$ becomes excessively large, causing loss oscillations and unstable training. To resolve



(a) Denoising Projection Iteration Process

(b) Projection

(c) Projection Trend Estimation Module

Fig. 6. Schematic of projection. (a) Projection steps are integrated into the diffusion denoising process. (b) Projection workflow: 1) Identify the reference point; 2) Project to the nearest topological node; 3) Find the adjacent point based on the projection trend; 4) Perform safety projection. (c) The projection trend estimation module (forward/reverse) takes the end/start point as the target, respectively, and outputs the projection trend.

this, we adopt the method from [16]: x_{t-1} is obtained via the forward process, and the average safety loss μ_s of x_{t-1} is precomputed based on the full dataset to reweight $\mathcal{L}_s$ accordingly.

The final training loss is $\mathcal{L} = \mathcal{L}_n + \lambda \frac{\mathcal{L}_s}{\mu_s}$, where λ is the weight coefficient to adjust the contribution of the safety constraint loss to the total loss.

D. Inference and Projection

In the inference phase, we adopt PDM's method [15] by integrating the projection strategy into the denoising iteration. As shown in Fig. 6a, the workflow initiates with a Gaussian noisy trajectory. Starting from diffusion step T, the model predicts noise to denoise the trajectory via eq. (3), followed by projecting the denoised result. This denoising-projection cycle repeats until step 0, yielding the final trajectory.

The original PDM code employs two trajectory projection methods, both with limitations (verified experimentally in Fig. 7): point projection maps each point to the nearest obstacle-free region. While effective for resolving local collisions, it fails to guide denoising toward correct regions in maze environments when the overall trajectory trend is erroneous; line projection preserves continuity by projecting the current point into the previous point's safe region but tends to trap trajectories in local areas within mazes, preventing target reaching.

To address these, we propose a topology-aware guided bidirectional line projection strategy, using GNN to estimate topological graph trends for optimized projection. As shown in Fig. 6c, the GNN input includes directed topological graph G's features with target-point information. It outputs labels for G's edges: label 0 for "forward" and label 1 for "reverse". Specifically, edge labels are determined by vertex topological distances to the target: edge (u,v) is labeled 0 if $dist(u, target) > dist(v, target)$, 1 otherwise.

Algorithm 1 Topology-guided Trajectory Projection

Input: Guide topology nodes $P = \{p_n \mid n = 1..N\}$ with safety radii r_k;
Trajectory points after the denoising step x_t;
Forward Projection Trend t_f;
Backward Projection Trend t_b;

for $k = 2$ *to* $K/2$ **do**
  $S = \{p_n \in P \mid \text{dist}(x_t^{k-1}, p_n) < r_n\}$;
  $p^k_{\text{near}} = \arg\min_{p_n \in S} \text{dist}(x_t^{k-1}, p_n)$;
    ▷ Find nearest topology point
  $p^k_{\text{adj}} = \text{adj}(p^k_{\text{near}}, t_f)$; ▷ Get adjacent point by t_f
  $x_t^{k*} = \text{P}(x_t^k, (p^k_{\text{adj}}, r^k_{\text{adj}}))$; ▷ Project to safe area
end
for $k = K-1$ *to* $K/2+1$ **do**
  $S = \{p_n \in P \mid \text{dist}(x_t^{k+1}, p_n) < r_k\}$;
  $p^k_{\text{near}} = \arg\min_{p_n \in S} \text{dist}(x_t^{k+1}, p_n)$;
    ▷ Find nearest topology point
  $p^k_{\text{adj}} = \text{adj}(p^k_{\text{near}}, t_b)$; ▷ Get adjacent point by t_b
  $x_t^{k*} = \text{P}(x_t^k, (p^k_{\text{adj}}, r^k_{\text{adj}}))$; ▷ Project to safe area
end

The projection process is implemented as in algorithm 1: forward projection iterates over points from $k = 2$ to $K/2$, first mapping x_t^{k-1} to the nearest topology node (as shown in Fig. 6b), then to an adjacent node p_{adj} via GNN-based forward trend estimation (with adjacency index m tied to GNN label probability differences), before projecting to the corresponding topology circle (p_{adj}, r_{adj}). The projection formula is defined as:

$$P(x, (p_{adj}, r_{adj})) = \text{argmin}(\max(||x_t^k - p_{adj}||_2 - r_{adj}, 0)). \tag{8}$$

The denoising step formula for each iteration is as follows:

$$x^*_{t-1} = P(D(x_t, t), (p_{\text{adj}}, r_{\text{adj}})). \tag{9}$$

Reverse projection follows an analogous process, iterating from $k = K-1$ to $k = K/2+1$; the distinction is that target point fed into the GNN is replaced with the start point.

V. EXPERIMENTS

A. Data Generation

Three datasets were generated for evaluation. Dataset A and Dataset B use Kruskal's MST Algorithm [21] to create dense-obstacle mazes (following DiPPeR's approach), challenging the safety of diffusion-based trajectory generation. To validate the model's ability to learn distinct preferences, two ground-truth trajectory types were designed: Dataset A prioritizes safety, while Dataset B focuses on shorter paths. Dataset C comprises realistic, diverse maps with randomly varying channel sizes/quantities, better reflecting real-world scenarios. All ground-truth trajectories are generated via A* pathfinding with B-spline optimization for smoothness. Each dataset includes 10,000 training maps, 200 validation maps, and 100 test maps (100×100 pixels), with 100 trajectories per map.

B. Comparative Experiments

All experiments were conducted on an NVIDIA RTX A6000. Our model was trained on 10,000×100 samples for 200 epochs, with 100 diffusion iterations and a trajectory horizon of 180. A separate model was trained for each dataset, tested on their respective test set (100 maps, 10 trajectories per map).

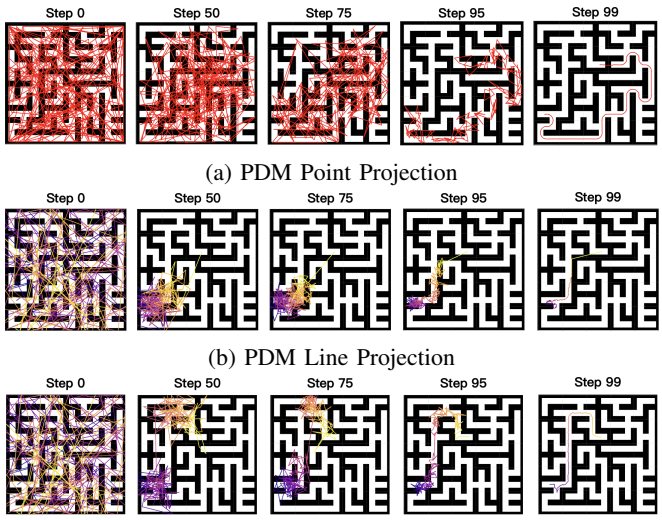
We compared against baselines: DiPPeR [13] (reproduced to match the original paper, as it is not open-source), learning-enhanced methods (N-A* [3], NIRRT* [6]), and traditional methods (RRT*, A*). Since PDM [15] is not designed for global path planning in generalized scenarios, we incorporate its projection component into our framework (corresponding to the subsequent schemes STDP-P and STDP-L) to conduct a comparison of projection strategies.

Our method has five variants: a) no projection; b) non-projection with 10 simultaneous trajectory samples; c) PDM point projection; d) PDM line projection; e) our proposed topology-guided projection. To balance performance and efficiency, a tiered combination scheme was designed: lightweight methods are prioritized, and subsequent methods are activated only if prior ones fail. The latter improves success rates but increases computational overhead, thus being triggered solely when safety risks exist. Details are provided in table I.

TABLE I Design of different combination schemes

Scheme	Description
STDP-lite	Only use variant a
STDP-lite10	Use variant a first; if failed, use variant b
STDP-P	Use variant a first; if failed, use variant c
STDP-L	Use variant a first; if failed, use variant d
STDP-T	Use variant a first; if failed, use variant e
STDP-PT	Use STDP-L first; if failed, use variant e
STDP-Full	Use STDP-lite10 first; if failed, use variant c; if failed again, use variant e

Success Rate and Time Overhead. We compared the success rates and time overheads of the diffusion method across the three types of datasets used in our training, with the experimental results presented in table II. The values in parentheses denote the individual time overheads of Variants a-e. Across all datasets, our method achieves significantly higher success rates than the baseline methods. For diffusion-based global trajectory planning, the final point is hard-constrained to the goal, so failures stem solely from trajectory unsafety, making success rate a direct indicator of safety performance. Specifically: STDP-lite incurs a time cost comparable to that of the baselines, while delivering substantial improvements in safety; a comparison between STDP-L and STDP-T confirms that our proposed projection method outperforms the line projection of PDM; and a comparison among STDP-P, STDP-T, and STDP-PT reveals the complementarity between PDM's point projection and our method—the former resolves local collision issues, whereas the latter corrects planning trend errors. As illustrated in Fig. 7, when the overall inference direction is deviated, only our method can guide the diffusion process to generate safe

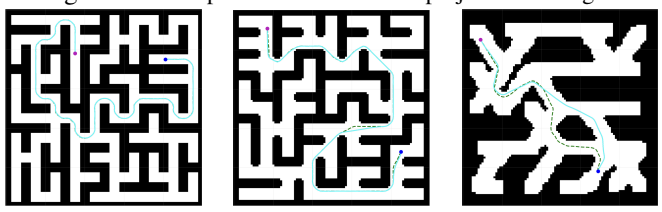


(a) PDM Point Projection

(b) PDM Line Projection

(c) Topology-aware Guided Bidirectional Line Projection

Fig. 7. Reverse processes of different projection strategies



(a) Dataset A (b) Dataset B (c) Dataset C

Fig. 8. Experimental results. Blue lines represent generated trajectories, and green lines denote ground-truth trajectories.

trajectories. STDP-Full achieves the best performance with moderate temporal overhead, without significant increases. This is attributed to the design of combination schemes: high-overhead methods only come into play in scenarios where the model struggles to find solutions, thus avoiding substantial increases in overall temporal overhead. The additional time overhead of our method compared to the baseline stems from the topology-aware module and projection process we designed. The average additional computational overhead incurred in forward propagation is 36.67 ms.

Trajectory Characteristics Learning. To verify the diffusion method's Trajectory Characteristics Learning capability, we evaluated it on Datasets A and B, both with distinct trajectory characteristics. Specifically, we calculated the MSE between successful trajectories (planned by different methods across tasks) and ground-truth trajectories; this MSE quantified each algorithm's ability to learn trajectory characteristics. As shown in table III, our method outperforms traditional and learning-enhanced methods in learning ground-truth trajectory Characteristics. Figs. 8a and 8b confirm the method can learn distinct trajectory characteristics (safety-prioritized/shortest path). Fig. 8c demonstrates its generalization: it generates shorter, smoother paths than A*-searched ground-truth trajectories. Notably, STDP-lite has lower MSE than other variants, as its MSE only includes safe trajectories. In contrast, other STDP methods have higher success rates—their MSE calculation covers more

TABLE II Success Rate and Time Performance of Diffusion methods

Method	Dataset A		Dataset B		Dataset C		Avg SR(%)↑
	SR(%)↑	Time(s)↓	SR(%)↑	Time(s)↓	SR(%)↑	Time(s)↓	
DiPPeR	91.2	**0.75**	84.60	**0.76**	81.71	**0.80**	85.84
STDP-lite (variant a)	96.10	0.80 (0.80)	85.18	0.81 (0.81)	83.32	0.81 (0.81)	88.20
STDP-lite10 (variant b)	98.75	0.87 (1.73)	93.38	1.07 (1.75)	90.11	1.11 (1.75)	94.08
STDP-P (variant c)	97.25	0.84 (0.87)	91.52	0.94 (0.88)	92.40	0.96 (0.88)	93.72
STDP-L (variant d)	96.42	1.29 (12.52)	86.42	2.65 (12.38)	84.59	2.91 (12.55)	89.14
STDP-T (variant e)	96.66	1.44 (16.40)	88.87	3.21 (16.19)	86.40	3.54 (16.34)	89.95
STDP-PT	97.63	1.29	93.22	2.32	93.38	2.20	94.74
STDP-Full	**99.06**	1.05	**96.34**	1.86	**95.30**	2.09	**96.90**

TABLE III MSE Performance of all methods (in pixel^2)

Category	Method	Dataset A	Dataset B
Traditional	RRT*	2.33	2.17
Learning-Enhanced	N-A*	3.15	3.37
	NIRRT*	3.99	2.41
Diffusion	DiPPeR	0.023	1.07
	STDP-lite	**0.015**	**0.98**
	STDP-lite10	**0.015**	1.50
	STDP-P	0.017	1.46
	STDP-L	0.025	1.09
	STDP-T	0.050	1.42
	STDP-PT	0.043	1.72
	STDP-Full	0.025	1.84

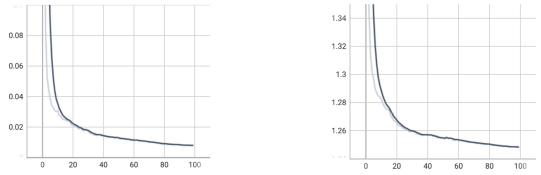
TABLE IV Average Trajectory Generation Time (in seconds) of different methods across maps with various resolutions.(Resolution: A.280×280; B.360×360; C.600×600; D.760×760.)

Method	Map A	Map B	Map C	Map D
A*	2.21	6.70	16.9	35.76
RRT*	**0.29**	1.13	5.04	5.91
N-A*	1.3	3.85	6.1	13.49
NIRRT*	0.31	-	-	-
DiPPeR	0.79	**0.82**	**0.81**	**0.82**
STDP	0.83	0.85	0.86	0.91

challenging scenarios, directly raising the average MSE.

Stable Time Overhead. To verify time overhead stability, we conducted a comparative experiment on the MRPB benchmark [22] (containing multi-scale complex scenarios), where 10 trajectory generations were performed for each map. As shown in table IV: traditional and learning-enhanced methods exhibit significantly increased overhead with larger map scales (and NIRRT* fails in some maps); in contrast, our method remains stable across scales—map scale only affects the feature encoding stage, which accounts for an extremely low proportion of total overhead. This result confirms the core advantage of the diffusion method: stable time overhead. The fundamental reason is that map scale only impacts the conditional encoding module, whose proportion in the total overhead of the diffusion method is minimal.

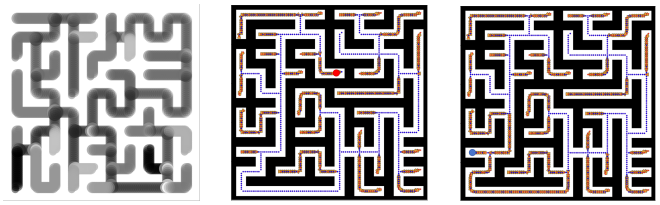
Safety Loss. In training with small-sample data on maze scenarios, we compared the convergence performance between the topology anchor-based safety loss and the nearest safe point-based safety loss. As shown in Fig. 9, the experimental results demonstrate that the safety loss adopted in this paper achieves smooth convergence.



(a) The topology anchor-based safety loss converges rapidly and stably

(b) The nearest safety point-based safety loss struggles to converge to a low value.

Fig. 9. Training convergence of different safety losses. The x-axis represents the number of training iterations, and the y-axis represents the loss value.



(a) Result of topology encoder network

(b) Result of forward Trend Estimation network

(c) Result of backward Trend Estimation network

Fig. 10. Topology-Aware GNN Results. (a) Darker colors indicate higher regional planning attention. (b) and (c) Red/blue dots: destination/starting point; yellow arrows: projected trend direction.

C. GNN Topology-Aware Network

This paper employs Gated-GCN to train two models: a topology encoder network for topological feature encoding, and a Trend Estimation network for guiding projection. The topology encoder focuses on identifying all potential planning channels, prioritizing recall with a recall-accuracy weight ratio of 9:1 for optimization. Conversely, the Trend Estimation network uses a standard 1:1 weight. GNN-related outputs are shown in Fig. 10.

D. Simulation and Real-World Experiments

Performance validation was conducted in both simulated and real-world scenarios, using hardware based on the YUHESEN FW-MINI chassis—equipped with an LSLIDAR C32 and Realsense D435i for environmental perception.

To validate its capability as a global planner, ST-DiffPlanner was integrated into the Autonomous Exploration Development Environment (AEDE) framework, with the model deployed on a remote workstation. Specifically, LIWO

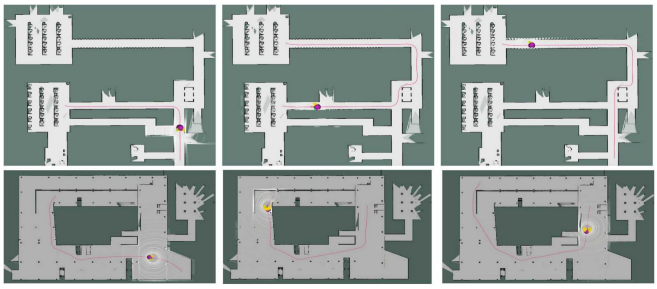


Fig. 11. Simulation Environments Deployment.

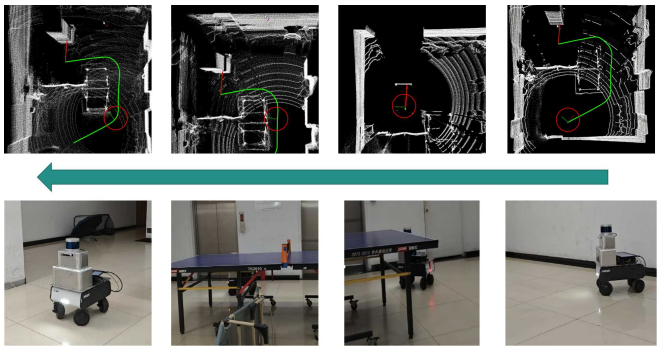


Fig. 12. Real-World Deployment.

[23] provides localization, while the AEDE framework handles obstacle detection and local planning via the Dynamic Window Approach (DWA); ST-DiffPlanner, as the global planner with a mapping resolution of 0.05 m, generates local waypoints to guide DWA. Additionally, AEDE includes a Gazebo simulation environment.

Effectiveness and platform invariance of the method were verified in both simulated scenarios and real-world scenarios, achieving an average planning time of 0.94 s, with experimental results presented in Fig. 11 and Fig. 12.

VI. CONCLUSION

This paper presents ST-DiffPlanner, a topology-aware diffusion path planner that enhances planning safety. Compared with the existing SOTA diffusion-based global planners, our method offers superior performance: on one hand, STDP-lite maintains the same time overhead as the SOTA but outperforms it in success rate by 2.36%; on the other hand, although STDP-Full sacrifices a small amount of time overhead, it substantially boosts the success rate by 11.06% relative to the SOTA. It retains diffusion's strengths in learning trajectory characteristics and maintaining stable overhead in large-scale scenarios. Effectiveness and platform invariance are verified in both simulation and real-world environments, with successful tests demonstrating platform-independent multimodal learning ability. Future work will extend this method to multi-agent collaborative planning for more complex tasks.

REFERENCES

[1] Q. Chai, Y. Wang, Y. He, C. Xu, and Z. Hong, "Improved prm path planning in narrow passages based on pso," in *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2022, pp. 41–46.

[2] Y. Huang, H. Wang, L. Han, and Y. Xu, "Robot path planning in narrow passages based on improved prm method," *Intelligent Service Robotics*, vol. 17, no. 3, pp. 609–620, 2024.

[3] R. Yonetani, T. Taniai, M. Barekatain, M. Nishimura, and A. Kanezaki, "Path planning using neural a* search," in *International conference on machine learning*. PMLR, 2021, pp. 12 029–12 039.

[4] J. Liu, S. Lyu, D. Hadjivelichkov, V. Modugno, and D. Kanoulas, "Vit-a*: Legged robot path planning using vision transformer a," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–6.

[5] J. Li, S. Wang, Z. Chen, Z. Kan, and J. Yu, "Lightweight neural path planning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 6713–6718.

[6] Z. Huang, H. Chen, J. Pohovey, and K. Driggs-Campbell, "Neural informed rrt*: Learning-based path planning with point cloud state representations under admissible ellipsoidal constraints," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8742–8748.

[7] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," *arXiv preprint arXiv:2205.09991*, 2022.

[8] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[9] Y. Shan, Z. Zhu, T. Long, Q. Liang, Y. Chang, W. Zhang, and L. Yin, "Contrastive diffuser: Planning towards high return states via contrastive learning," *arXiv preprint arXiv:2402.02772*, 2024.

[10] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1916–1923.

[11] W. Yu, J. Peng, H. Yang, J. Zhang, Y. Duan, J. Ji, and Y. Zhang, "Ldp: A local diffusion planner for efficient robot navigation and collision avoidance," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 5466–5472.

[12] K. Kondo, A. Tagliabue, X. Cai, C. Tewari, O. Garcia, M. Espitia-Alvarez, and J. P. How, "Cgd: Constraint-guided diffusion policies for uav trajectory planning," *arXiv preprint arXiv:2405.01758*, 2024.

[13] J. Liu, M. Stamatopoulou, and D. Kanoulas, "Dipper: Diffusion-based 2d path planner applied on legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9264–9270.

[14] W. Xiao, T.-H. Wang, C. Gan, R. Hasani, M. Lechner, and D. Rus, "Safediffuser: Safe planning with diffusion probabilistic models," in *The Thirteenth International Conference on Learning Representations*, 2023.

[15] J. K. Christopher, S. Baek, and F. Fioretto, "Constrained synthesis with projected diffusion models, 2024," 2024.

[16] A. Li and R. Beeson, "Aligning diffusion model with problem constraints for trajectory optimization," *arXiv preprint arXiv:2504.00342*, 2025.

[17] C. Li, H. Ma, J. Wang, and M. Q.-H. Meng, "Bidirectional search strategy for incremental search-based path planning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7311–7317.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[19] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[20] C. Lantuéjoul, "Skeletonization in quantitative metallography," 1980.

[21] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[22] J. Wen, X. Zhang, Q. Bi, Z. Pan, Y. Feng, J. Yuan, and Y. Fang, "Mrpb 1.0: A unified benchmark for the evaluation of mobile robot local planning approaches," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 8238–8244.

[23] Z. Yuan, F. Lang, T. Xu, and X. Yang, "Liwo: Lidar-inertial-wheel odometry," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1481–1488.