IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

DiffRP: Diffusion-Driven Promising Region Prediction for Sampling-Based Path Planning

Zongwu Xie[1†], Yiming Ji[1†], Yang Liu[1*], Yiqian Xie[2], Zhengpu Wang[1], Boyu Ma[1], and Baoshi Cao[1]

***Abstract*—Utilizing neural networks to predict potential regions containing optimal paths in advance and subsequently biasing the sampling probability towards these promising regions has been proven to effectively enhance the path planning efficiency of sampling-based algorithms. Undoubtedly, the accuracy of the promising regions is of paramount importance. Currently, the generalizability of many CNN- or Transformer-based models remains limited, often performing poorly in unknown environments. To enhance generalization capability, we reformulate the promising region prediction problem as a conditional generation task and address it using a diffusion model, referred to as the DiffRP (Diffusion-based Region Prediction). We propose three paradigms for generating promising regions, among which we innovatively introduce a biased noise initialization method for the diffusion process. Specifically, we bias the mean of the noise distribution using obstacle maps and design a map-conditioned denoising model to progressively generate accurate promising regions from the biased noise. Experiments on public datasets demonstrate that our proposed DiffRP method outperforms existing state-of-the-art models by 35$\sim$42% in promising region prediction accuracy. Moreover, the non-uniform sampling algorithm (DiffRP-RRT*) based on this region achieves a 3$\sim$52% reduction in sample number compared with other neural-network-driven approaches.**

***Index Terms*—AI-based methods, Motion and path planning.**

I. Introduction

PATH planning is essential for autonomous robots to navigate efficiently [1]. Existing methods fall into three categories: graph-based (e.g., A* [2]), data-based (e.g., APF [3]), and sampling-based. Graph-based methods face high computational complexity in large spaces, while data-based approaches often get stuck in local minima.

Sampling-based methods, such as Rapidly-exploring Random Trees, utilize stochastic sampling to efficiently explore the configuration space, providing robust search capabilities and rapid exploration [4]. These methods have been widely adopted due to their computational efficiency in high-dimensional state spaces. Moreover, their asymptotically optimal variants, including RRT* and Probabilistic Roadmaps [5], have been developed to ensure convergence toward optimal solutions.

Sampling-based methods often exhibit two key limitations: 1) reliance on randomized initial solutions and 2) slow convergence to optimality. These issues arise from uniform sampling across the state space, which generates redundant samples and necessitates extensive post-processing. Consequently, computational bottlenecks emerge, particularly in time-intensive operations such as collision detection. To address these challenges, heuristic sampling strategies have been developed to prioritize promising regions—areas of the state space where feasible or near-optimal solutions are more likely to exist.

With the remarkable advancements of deep learning techniques across diverse domains [6][7], researchers have proposed learning-based methods to precompute the aforementioned promising regions. Neural RRT* [8] exemplifies this approach, using a CNN to predict promising regions for non-uniform sampling, improving planning efficiency. Wang et al. further advanced this approach by proposing the NEED model [9]. Ma et al. also investigated the potential of implementing Generative Adversarial Networks to achieve non-uniform sampling distributions in path planning [10]. Building on CNNs, Huang et al. proposed a dual-supervision paradigm using region and guideline signals [11], improving sampling-based planners. Similarly, Johnson et al. introduced a transformer-based method to identify critical regions in configuration space [12], focusing computation on high-probability optimal paths for complex motion planning.

These studies demonstrate that neural network-based approaches, which predict promising regions likely to contain optimal paths based on given maps and start-goal configurations, can effectively accelerate sampling-based path planning methods. Crucially, the sampling efficiency and computational performance improve proportionally with how closely the predicted regions align with the actual optimal path. However, current approaches exhibit limited generalization capability to unfamiliar environments. We argue that this is because existing models fail to comprehensively capture the intricate relationship between the map and the start-goal points, merely performing per-grid predictions at the semantic segmentation level. In essence, they treat the problem as a semantic segmentation task on an image composed of obstacles and start-goal points, attempting to classify pixels belonging to the optimal path. From a statistical perspective, the model's task is to learn the underlying distribution of optimal paths conditioned on obstacle maps and start-goal configurations, thereby predicting promising regions by sampling from the path distribution most aligned with the map. We thus reformulate promising region prediction as a conditional generation problem, motivating the adoption of advanced data-generation techniques to capture the latent distribution of paths and maps.

Inspired by the success of diffusion models in modeling complex distributions [13] as well as their applications in

This work was supported by by the National Natural Science Foundation of China (Grant No. 52470512) and the Young Scientists Fund of the National Natural Science Foundation of China (Grant No. 52105014)

Authors[1] are with State Key Laboratory of Robotics and Systems, Harbin Institute of Technology. Author[2] is with MinJiang University. * Corresponding author: Yang Liu. liuyanghit@hit.edu.cn. † Contributed equally.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

downstream tasks [14][15], we employ them to surpass the limitations of CNN- or Transformer-based methods.

Our framework introduces three novel paradigms: 1) **Direct Generation with Map-Conditioned Diffusion**: We first propose an end-to-end approach where the diffusion model generates promising regions directly from random noise.Compared to classical conditional diffusion generation works, the condition in our task is no longer text or class labels but obstacle maps and start-goal information. Therefore, when designing the denoising model based on the classic U-Net, we additionally introduce Map Feature Fusion and Map-Conditioned Attention modules to enhance map-awareness. 2) **Generation with Biased Initialization**: We innovatively decouple the obstacle map from the start-goal information and propose a biased noise-based diffusion generation mechanism. This mechanism leverages the obstacle map to generate non-uniform initial noise while inheriting the denoising model design from Paradigm-1. 3) **Prediction-Refinement Pipeline**: Recognizing the importance of post-processing refinement, we architect a two-stage generation pipeline that first employs a CNN-based front-end to estimate a coarse prediction then leverages a diffusion-based back-end to progressively refine these predictions through our proposed coarse-to-fine architecture, where the initial coarse predictions replace the binary map bias in Paradigm-2 to guide the generation. while map information continuously directs the iterative refinement process through conditioned denoising steps.

In summary, our study has the following main contributions:

1) The first application of diffusion models to promising region prediction, featuring a novel biased diffusion process that incorporates map information through noise biasing for more accurate generation;

2) An enhanced map-conditioned denoising model incorporating both feature fusion and cross-attention mechanisms, significantly improving upon standard U-Net architectures;

3) Comprehensive experimental validation showing our method outperforms state-of-the-art approaches in prediction accuracy by 30% on benchmark datasets. We further demonstrate the practical effectiveness through DiffRP-RRT*, our diffusion-enhanced path planner that achieves 5.0-70.3% reduction in sample number over existing neural-RRT* variants.

II. Preliminaries

A. Diffusion Models

We provide a concise overview of the formulation of diffusion models as presented in [13]. Diffusion models comprise a forward process and a reverse process. The forward process $q\left(x_{1:T}|x_0\right)$ employs a Markov chain to gradually transform the data distribution $x_0 \sim q\left(x_0\right)$ into standard Gaussian noise $x_T \sim \mathcal{N}\left(0, \mathbf{I}\right)$. The forward process can be formulated as the following Eq. (1).

$$q\left(x_t|x_{t-1}\right) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}\right) \tag{1}$$

Let $\alpha_t = 1 - \beta_t$, which yields Eq. (2) as the following, where β_t is a hyperparameter associated with diffusion steps.

$$x_t = \sqrt{\alpha_t}x_{t-1} + \epsilon\sqrt{1-\alpha_t}, \epsilon \sim \mathcal{N}\left(0, \mathbf{I}\right) \tag{2}$$

The reverse process implements a gradual denoising procedure that reconstructs the original data distribution from random noise. This reverse process is parameterized by θ and formally defined as the following Eq. (3).

$$p_\theta\left(x_{t-1}|x_t\right) = \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right) \tag{3}$$

III. Methodology

A. Biased Diffusion Process

We present a modified denoising initialization for the reverse process in DDPM [13]. Unlike the standard approach that begins with isotropic Gaussian noise $\mathcal{N}\left(0, \mathbf{I}\right)$, our method introduces a biased initialization where each pixel's noise distribution is centered at its corresponding obstacle map value M_o. Formally, the initial noise distribution is as the following Eq. (4):

$$x_T = M_o + \epsilon, \epsilon \sim \mathcal{N}\left(0, \mathbf{I}\right) \tag{4}$$

Where $M_o \in \{0,1\}^{H\times W}$ denotes an binary map where obstacles are marked as 0 and free space as 1. In contrast to the conventional formulation in Eq. (2), we propose a modified forward process incorporating explicit map guidance, as formally defined in Eq. (5).

$$\begin{aligned} x_t = &\sqrt{\alpha_t}x_{t-1} + \\ &M_o\left(\sqrt{1-\bar{\alpha}_t} - \sqrt{\alpha_t - \bar{\alpha}_t}\right) + \sqrt{(1-\alpha_t)}\epsilon \end{aligned} \tag{5}$$

Where $\bar{\alpha}_t = \alpha_t\alpha_{t-1}...\alpha_1$, we employ mathematical induction to derive the final expression in Eq. (6).

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + M_o\left(\sqrt{1-\bar{\alpha}_t}\right) + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{6}$$

In a word, we modify DDPM's forward process by adding a term that makes the diffusion converge to an obstacle map-guided non-standard Gaussian, instead of the original isotropic Gaussian.

We now formalize the reverse process. Beginning from the map-biased noise distribution x_T, this process iteratively removes noise through a learned denoising model, ultimately reconstructing the path region x_0. The denoising trajectory follows a Markov chain conditioned on the obstacle map M_o, ensuring the generated paths respect environmental constraints.

According to Bayse theorem, the posterior probability is as the following Eq. (7).

$$p\left(x_{t-1}|x_t, x_0\right) = \frac{p\left(x_t|x_{t-1}, x_0\right)p\left(x_{t-1}|x_0\right)}{p\left(x_t|x_0\right)} \tag{7}$$

From Eq. (5) and Eq. (6), we derive Eq. (8)-(10).

$$\begin{aligned} &p\left(x_t|x_{t-1}, x_0\right) = p\left(x_t|x_{t-1}\right) \sim \\ &\mathcal{N}\left(\sqrt{\alpha_t}x_{t-1} + M_o\left(\sqrt{1-\bar{\alpha}_t} - \sqrt{\alpha_t - \bar{\alpha}_t}\right), 1-\alpha_t\right) \end{aligned} \tag{8}$$

$$p\left(x_t|x_0\right) \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0 + M_o\left(\sqrt{1-\bar{\alpha}_t}\right), 1-\bar{\alpha}_t\right) \tag{9}$$

$$\begin{aligned} &p\left(x_{t-1}|x_0\right) \sim \\ &\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}x_0 + M_o\left(\sqrt{1-\bar{\alpha}_{t-1}}\right), 1-\bar{\alpha}_{t-1}\right) \end{aligned} \tag{10}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

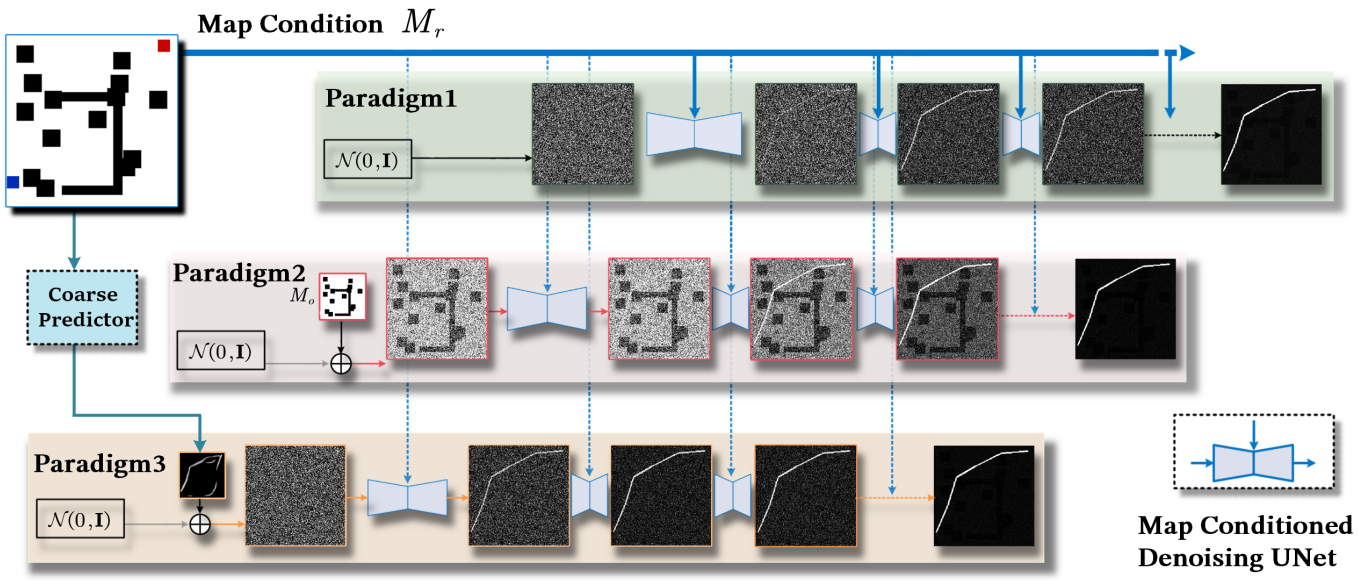

Figure 1: The proposed three paradigms that predicate promising region via diffusion process

Assuming $p\left(x_{t-1}|x_t\right) \sim \mathcal{N}\left(\mu_s, \sigma_s^2\mathbf{I}\right)$, we substitute Eqs. (8)-(10) into Eq. (7) to obtain the following Eq. (11) and Eq. (12).

$$\begin{aligned}\mu_s = &\frac{\left(1-\bar{\alpha}_{t-1}\right)\sqrt{\alpha_t}}{1-\bar{\alpha}_t}x_t - \\ &\frac{\left(1-\bar{\alpha}_{t-1}\right)\sqrt{\alpha_t}M_o\left(\sqrt{1-\bar{\alpha}_t}-\sqrt{\alpha_t-\bar{\alpha}_t}\right)}{1-\bar{\alpha}_t} + \\ &\frac{\left(1-\alpha_t\right)\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}x_0 + \frac{\left(1-\alpha_t\right)M_o\left(\sqrt{1-\bar{\alpha}_{t-1}}\right)}{1-\bar{\alpha}_t}\end{aligned} \tag{11}$$

$$\sigma_s^2 = \frac{\beta_t\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \tag{12}$$

Using Eq. (6), we substitute x_0 in Eq. (11) with the following Eq. (13). And the posterior probability can be computed from Eq. (11) and Eq. (12).

$$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}x_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}M_o - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\epsilon \tag{13}$$

Following the DDPM framework, we avoid direct parameterization of $\mu_\theta\left(x_t, t\right)$ in Eq. (3) via a neural network. Instead, we train a model $\psi_\theta\left(M_r, x_t, t\right)$ to predict the ϵ in Eq. (13).

We adopt the same simplified objective function as the following Eq. (14), where M_r denote the RGB navigation map encoding start (green) and goal (red) positions.

$$L \coloneqq E_{t\sim[1,T],x_0\sim q(x_0),\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\left\|\epsilon - \psi_\theta\left(M_r, x_t, t\right)\right\|^2\right] \tag{14}$$

Song et al [16]. proposed an enhanced sampler that significantly accelerates the inference process. Building on this work, we derive a map-biased accelerated sampling method by assuming x_{t-1} to be a linear combination of x_0, x_t, and noise ϵ, as formalized in Eq. (15).

$$x_{t-1} = kx_0 + mx_t + c + \sigma\epsilon \tag{15}$$

Substituting Eq. (6) into this formulation yields the result in Eq. (16).

$$\begin{aligned}x_{t-1} &= \left(k + m\sqrt{\bar{\alpha}_t}\right)x_0 + mM_o\left(\sqrt{1-\bar{\alpha}_t}\right) + c + \acute{\epsilon}, \\ \acute{\epsilon} &\sim \mathcal{N}\left(0, m^2\left(1-\bar{\alpha}_t\right) + \sigma^2\right)\end{aligned} \tag{16}$$

Comparing with Eq. (10). This allows us to compute the values of parameters k, m, and c. Substituting parameters k, m, and c into the original Eq. (15) and combining with Eq. (6), we derive the final formulation Eq. (17). Similar to the DDIM approach [16], this results in a non-Markovian reverse process where x_{t-1} and x_t can span multiple timesteps. With $\tau = t - \Delta t$, this formulation significantly reduces the sampling steps while maintaining sample quality - a crucial advantage for path planning applications.

Notably, our modified sampler based on Eq. (17) achieves accelerated inference without requiring retraining of the diffusion model, as described in Alg. 3.

$$\begin{aligned}x_\tau = &\sqrt{\bar{\alpha}_\tau}\frac{x_t - M_o\left(\sqrt{1-\bar{\alpha}_t}\right) - \sqrt{1-\bar{\alpha}_t}\psi_\theta\left(\cdot\right)}{\sqrt{\bar{\alpha}_t}} + \\ &\sqrt{1-\bar{\alpha}_\tau-\sigma^2}\psi_\theta\left(\cdot\right) + M_o\left(\sqrt{1-\bar{\alpha}_\tau}\right) + \sigma\epsilon\end{aligned} \tag{17}$$

The training workflow is formally presented in Alg. 1 where each iteration begins by sampling a training pair (P_i, M_i) from the dataset, with M_i being a composite of components M_r and M_o. The training process iteratively computes the loss function defined in Eq. (14).

The inference procedure and acceleration algorithm are formally described in Alg. 2 and Alg. 3, respectively. While Alg. 3 successfully reduces inference time, our experimental validation reveals that this acceleration comes at the cost of generating sparse noise artifacts in the predicted promising regions. To mitigate this quality degradation, we implement a post-processing step using morphological opening operations with a 2×2 kernel. This non-trainable processing step requires less than 0.5ms while effectively eliminating outlier noise

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1 Training

Input: Total diffusion steps T, path and map datasets $\mathcal{D} = \{(P_k, M_k)\}_{k=1}^{K}$, denoising network $\psi_\theta(\cdot)$, for the datasets, M_r represant the $H \times W \times 3$ map with start and goal encoded in RGB value,
Output: Trained denoising network $\psi_\theta(\cdot)$
repeat
  Sample $(P_i, M_i) \sim \mathcal{D}$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
  Sample $t \sim \text{Uniform}(\{1, ..., T\})$
  Calculate $\beta_t = \frac{10^{-4}(T-t)+2\times10^{-2}(t-1)}{T-1}$
  Calculate $\alpha_t = 1 - \beta_t$
  Calculate $\bar{\alpha}_t = \prod_{n=1}^{t} \alpha_n$
  Take gradient step on $\nabla_\theta \left\| \epsilon - \psi_\theta(M_i^r, x_t, t) \right\|, x_t = \sqrt{\bar{\alpha}_{t-1}} P_i + M_i^o(\sqrt{1-\bar{\alpha}_{t-1}}) + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon$
until Convergence

Algorithm 2 Inference

Input: Total diffusion steps T, the RGB map with start and goal M_r, the obstacle map M_o, trained denoising network $\psi_\theta(\cdot)$
Output: Promising region prediction $\hat{P}$
Sample $x_T \sim \mathcal{N}(0, \mathbf{I})$
for t in $\{T, T-1, ..., 1\}$ **do**
  $z \sim \mathcal{N}(0, \mathbf{I})$
  $\beta_t = \frac{10^{-4}(T-t)+2\times10^{-2}(t-1)}{T-1}$
  $\alpha_t = 1 - \beta_t$
  $\bar{\alpha}_t = \prod_{n=1}^{t} \alpha_n$
  $\sigma_t^2 = \frac{\beta_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$
  Estimate $x_0 = \frac{x_t - M_o\sqrt{1-\bar{\alpha}_t} - \psi_\theta(M_r, x_t, t)\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}$
  Calculate $c_1 = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t} M_o\left(\sqrt{1-\bar{\alpha}_t} - \sqrt{\alpha_t - \bar{\alpha}_t}\right)}{1-\bar{\alpha}_t}$
  Calculate $c_2 = \frac{(1-\alpha_t) M_o\left(\sqrt{1-\bar{\alpha}_{t-1}}\right)}{1-\bar{\alpha}_t}$
  Calculate $x_{t-1} = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t} x_t + \frac{(1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t} x_0 - c_1 + c_2 + \mathbf{1}_{[t>1]}\sigma_t z$
end for
Return x_0

Algorithm 3 Accelerated Inference

Input: Total diffusion steps T, the RGB map with start and goal M_r, the obstacle map M_o, trained denoising network $\psi_\theta(\cdot)$
Output: Promising region prediction $\hat{P}$
Generate sub-sequence of length S: $[\tau_1, ..., \tau_S]$
Sample $x_{\tau_S} \sim \mathcal{N}(0, \mathbf{I})$
for t in $\{\tau_S, \tau_{S-1}, ..., \tau_1\}$ **do**
  $\beta_t = \frac{10^{-4}(T-t)+2\times10^{-2}(t-1)}{T-1}$
  $\alpha_t = 1 - \beta_t$
  $\bar{\alpha}_t = \prod_{n=1}^{t} \alpha_n$
  Estimate $x_0 = \frac{x_t - M_o\sqrt{1-\bar{\alpha}_t} - \psi_\theta(M_r, x_t, t)\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}$
  Predict $\epsilon_t = \frac{x_t - \sqrt{\bar{\alpha}_t} x_0}{\sqrt{1-\bar{\alpha}_t}} - M_o$
  $x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_t + M_o(\sqrt{1-\bar{\alpha}_{t-1}})$
end for
Return x_0

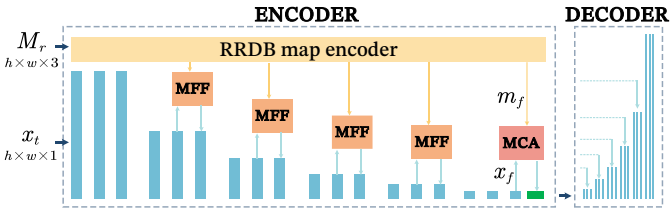

Figure 2: Schematic diagram of the Map-conditioned denoising model architecture.

points, thereby significantly improving the visual quality and topological consistency of the final output without compromising the acceleration benefits.

B. Region Generation Paradigms via Diffusion Models

This paper presents three novel paradigms for promising region prediction using diffusion models, building upon the foundation of Alg. 1. As shown in Fig. 1. The first paradigm generates path regions directly from isotropic Gaussian noise $\mathcal{N}(0, \mathbf{I})$, where the obstacle map M_o is set to zero in both the forward process Eq. (6) and reverse process Eq. (11), while maintaining the denoising model's three conventional inputs: the navigation map M_r, noisy sample x_t, and timestep t. The second paradigm introduces obstacle-aware generation by incorporating the obstacle map M_o into the noise initialization process as detailed in Alg. 1, effectively biasing the diffusion process to respect environmental constraints. The third paradigm decomposes the task into a two-stage prediction-refinement pipeline, where a pre-trained model first produces a coarse region prediction that subsequently replaces the obstacle map M_o in the refinement phase, enabling high-precision generation through this coarse-to-fine approach.

While Paradigm-3 improves region prediction accuracy by incorporating pre-trained model outputs compared to Paradigm-2, it faces inherent limitations due to the uncertainty in coarse mask generation. We conducted detailed experiments on Paradigm-3 in Section IV-D. The significance of Paradigm-3 lies in its provision of a performance-enhancing approach for traditional promising region prediction models, offering insights for future multi-model integration.

C. Map-conditioned Denoising Model

In this section, we present the detailed architecture of the denoising model $\psi_\theta(M_r, x_t, t)$ in Eq. (14). The U-Net [17] serves as the backbone of the denoising model. Additionally, we specifically design a Map Feature Fusion (MFF) module and a Map-Conditioned Attention (MCA) module to effectively integrate map information into the denoising process, thereby enhancing model performance.

1) Map Feature Fusion Module: Inspired by SegDiff [18], we construct a multi-output image encoder (as shown in Fig. 2) using Residual-in-Residual Dense Blocks (RRDBs) [19] that generates hierarchical features aligned with the U-Net encoder's spatial and channel dimensions. Unlike SegDiff's single-output design, the MFF module concatenates the map

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

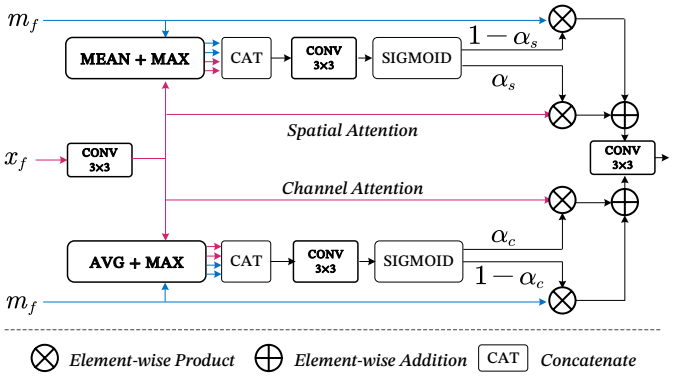

Figure 3: Schematic diagram of the Map Conditioned Attention Module architecture.

encoder's output m_f and U-Net's intermediate features x_f, processes them via depthwise separable convolution with layer normalization and ReLU, adds the result to x_f, then applies a final convolution. This fused feature representation is then hierarchically incorporated into the U-Net network for subsequent processing, enabling efficient and hierarchical map feature utilization while maintaining computational efficiency critical for path planning applications.

2) Map Conditioned Attention Module: Building upon the proven effectiveness of attention fusion modules in semantic segmentation tasks, where they successfully integrate high-level and low-level features to enhance segmentation accuracy [20], we adapt this mechanism to inject high-level map features from the map encoder into the denoising process. We incorporated the mechanisms of spatial attention and channel attention [21], with the MCA module embedded before the bottleneck of the U-Net.

As illustrated in Fig. 3, the MCA module comprises parallel spatial and channel attention branches that jointly process map features m_f and image features x_f to generate channel attention weights α_{ch} and spatial attention weights α_{sp}. The module first applies element-wise multiplication between the input features and their corresponding attention weights in each branch, then combines the refined features through element-wise addition, and finally produces the enhanced output via a convolutional layer for subsequent processing.

IV. Experiment

A. Datasets and Implementation Detail

We trained DiffRP using the public path planning dataset[1], following RGP [11]. The test set was partitioned into 1,200 SEEN cases, which exhibit similarity to the training data, and 1,200 UNSEEN scenarios, which differ from the training distribution. All training and simulation procedures were conducted on a single NVIDIA GeForce RTX 4090 GPU.

During training, the number of diffusion steps was set to 1000, however, during inference, this was reduced to only 5 steps (equivalent to $S = 5$ in Alg. 3), significantly accelerating the inference time.

[1]https://github.com/RTPWDSDM/PPD

Table I: Comparison of computational complexity and model size with existing methods

ID	Models		Size (MB) ↓	FLOPs (G) ↓	F1-SEEN ↑	F1-UNSEEN ↑
1	RGP [11]		39.95	52.74	58.16%	35.82%
	NEED [9]		38.83	45.81	57.12%	33.38%
	MPT [12]		84.32	107.48	55.24%	29.29%
	NDR [22]		110.85	123.52	61.22%	37.31%
	mc	nr				
2	32	2	94.42	(5×) 58.03	56.22%	50.87%
	32	4	112.80	(5×) 68.46	59.27%	51.06%
	64	2	197.76	(5×) 91.84	**62.38%**	**51.45%**
	64	4	250.71	(5×) 123.58	57.94%	48.18%
	128	2	558.79	(5×) 286.81	61.52%	48.44%
	128	4	781.87	(5×) 453.74	57.45%	46.81%

For comparison, we evaluated DiffRP against several baseline methods—RGP [11], NEED [9], MPT [12], and NDR [22]—on the public datasets. Among these, RGP, NEED, and NDR employ CNN-based network architectures, while MPT utilizes a transformer-based framework. For fair comparisons, we adopt the same configuration to train DiffRP and the baseline models: an Adam optimizer [23] with its suggested default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a learning rate of 1e-4. We train all the models for 100 epochs using the same dataset.

B. Promising Region Prediction Result and Comparison

We employ the F1-score metric to evaluate region prediction quality, where a higher F1-score indicates better alignment between model predictions and ground truth. While RGP utilizes two distinct training labels (large-scale-regions and small-scale-guidelines) from the public dataset, we uniformly adopt the 2-pixel ground truth paths as training labels for all models in our comparison.

Consistent with references [24], we employ fvcore[2] to calculate the model parameters and FLOPs required for a single forward pass of the diffusion model. Multiplying this value by the number of sampling steps yields the total computational cost for generating one sample. We evaluate the model's performance using F1 scores in both SEEN and UNSEEN scenarios, with the results presented in Table. I. Here, ID=1 denotes the group of baseline models, while ID=2 represents different configurations of the DiffRP-P2 model (hereafter, we use P1/2/3 to denote Paradigm-1/2/3). The experimental details corresponding to ID=2 are comprehensively analyzed in the ablation study (Section IV-D).

As shown in the Table. I, DiffRP-P2 achieves significant improvements of 40% in prediction accuracy for unseen scenarios compared to existing methods, while maintaining comparable performance to previous SOTA models on seen cases.

We particularly emphasize that our proposed DiffRP method demonstrates significantly superior performance on the UNSEEN testset compared to existing approaches, highlighting its enhanced generalization capability. As illustrated in Fig. 4, in the MAP3 - which was deliberately designed with

[2]https://github.com/facebookresearch/fvcore

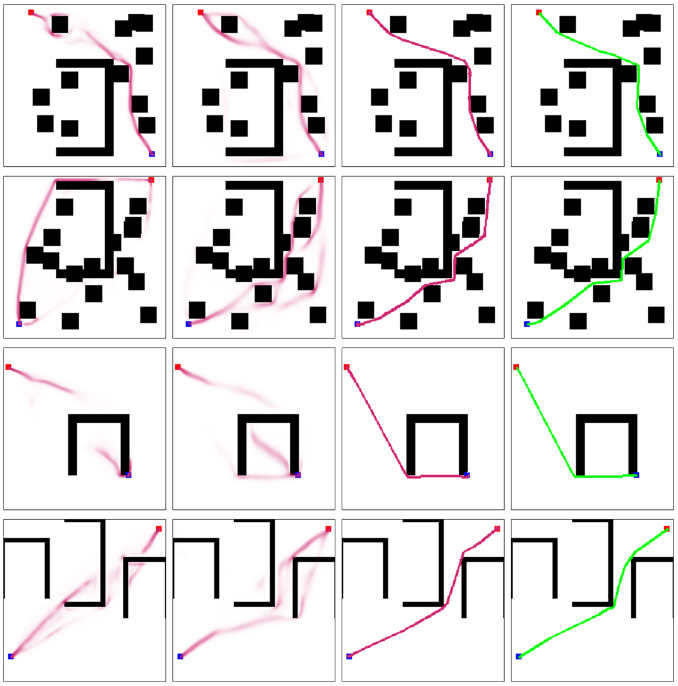

Figure 4: The comparison of RGP (first column), NEED (second column), our proposed DiffRP (third column), and ground truth (GT, fourth column). The first two rows (MAP1 and MAP2) display performance on SEEN testset scenarios, while the last two rows (MAP3 and MAP4) present more challenging UNSEEN testset cases with novel obstacle configurations.

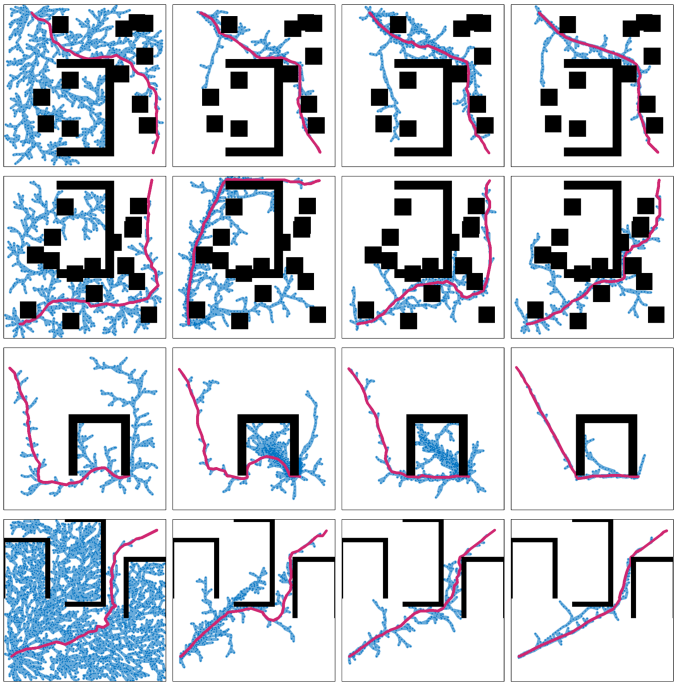

Figure 5: Path planning results. The first column represents the results obtained using RRT*, the second column represents RGP-RRT*, the third column denotes NEED-RRT*, and the fourth column illustrates our proposed DiffRP-RRT*. The blue color indicates the RRT tree, while the final path is marked in red.

the starting point positioned at the opening of a concave obstacle and completely unseen during training - both RGP and NEED models fail by generating predictions that fall into the concave obstacle. This results in ambiguous and discontinuous promising regions, which would undoubtedly degrade the sampling efficiency when guiding the RRT* algorithm, consequently impairing the optimal path search process. In stark contrast, DiffRP successfully predicts feasible regions that not only avoid the concave obstacle trap but also maintain region continuity and boundary clarity. This superior performance is further validated in the fourth-row scenario of Fig. 4, where DiffRP again demonstrates its remarkable capability in handling challenging, unseen environments.

C. Sampling-based Path Planning Result

This subsection employs maps MAP1–MAP4 (Fig. 4) as the simulation testbed to evaluate DiffRP's acceleration effect on RRT*. The modified RRT* allocates 90% sampling probability to the promising region and 10% to uniform sampling, whereas standard RRT* uses purely uniform sampling.

As illustrated in the Fig. 5, the blue branches demonstrate that neural-network-driven path planners benefit from non-uniform sampling, as the biased sampling strategy enables more efficient sample generation. For instance, in the third row of Fig. 4 (MAP3), the promising regions predicted by RGP and NEED fall inside a concave obstacle, leading to a significant number of ineffective sampling attempts within the obstacle (as shown in the third row of Fig. 5). Consequently, the non-uniform sampling RRT* algorithm performs even worse than uniform sampling in this scenario. This indicates that misleading promising regions can bias the algorithm toward excessive sampling in incorrect areas, ultimately degrading performance.

We conducted experiments by running all models for 1,000 iterations across 50 trials on each of the MAP1 to MAP4. The performance metrics, including average sample number, computation time, and success rate, were systematically recorded. A trial was deemed failed if the algorithm did not reach the goal vicinity within 1,000 iterations. All maps were configured at 256×256 resolution, with the results summarized in Table. II. The best-performing results are highlighted in bold font.

We further randomly sampled 100 map instances each from the SEEN and UNSEEN scenarios and computed the average values of all metrics. Experimental results show that conventional RRT* fails to achieve satisfactory success rates within 1,000 iterations, whereas Diff-RRT* overcomes this limitation and demonstrates significant improvement across all evaluation metrics.

D. Ablation Study

This subsection systematically validates: 1) the denoising model's complexity, 2) the accelerated sampling algorithm (Alg. 3), 3) the three proposed denoising paradigms, and 4) the effectiveness of the Map-conditioned Denoising Model.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Table II: Path planning results

		Diff-RRT*	NDR-RRT*	RGP-RRT*	NEED-RRT*	MPT-RRT*	RRT*
MAP1	Sample number↓	**162**	170	233	175	268	358
	Time↓	**0.11**	0.13	0.18	0.15	0.32	0.35
	Success rate↑	**1**	**1**	0.98	0.98	0.92	0.48
MAP2	Sample number↓	**225**	232	343	236	387	474
	Time↓	**0.15**	0.20	0.67	0.42	0.49	0.56
	Success rate↑	**1**	0.98	0.64	0.98	0.82	0.40
MAP3	Sample number↓	**144**	424	440	466	482	538
	Time↓	**0.14**	0.75	1.44	1.52	1.62	0.80
	Success rate↑	**1**	0.9	0.72	0.8	0.60	0.62
MAP4	Sample number↓	**250**	364	427	436	467	536
	Time↓	**0.22**	0.46	0.84	0.99	1.12	0.73
	Success rate↑	**0.98**	0.70	0.60	0.56	0.52	0.60
Avg 100 SEEN	Sample number↓	**168**	178	174	178	252	352
	Time↓	**2.21**	2.42	2.49	3.47	3.54	2.97
	Success rate↑	**0.98**	0.95	0.94	0.80	0.72	0.48
Avg 100 UNSEEN	Sample number↓	**137**	144	153	**137**	243	326
	Time↓	**2.54**	2.76	2.91	3.16	3.24	2.94
	Success rate↑	**0.96**	0.84	0.74	0.69	0.66	0.61

Table III: Ablation Study of Accelerated Sampling Algorithm

Mode	Use Alg. 3	Coarse Predictor	F1-SEEN (%)	F1-UNSEEN (%)	Time (s)
P1	✗	/	62.35	52.14	35.32
	✓, S=10	/	59.79	46.37	0.31
	✓, S=5	/	58.55	45.92	0.16
P2	✗,	/	63.92	55.31	35.26
	✓, S=10	/	62.66	52.72	0.31
	✓, S=5	/	62.38	51.45	0.15
P3	✓, S=5	NEED (ep5) UT	46.45→51.23	35.64→41.52	0.16
		NEED (ep10) T	55.32→63.36	38.48→53.23	0.16
		RGP (ep5) UT	44.28→50.73	35.16→42.67	0.15
		RGP (ep10) T	56.04→63.27	39.77→53.46	0.16

Table IV: Ablation Study of Map-conditioned Denoising Model

Models	Size (MB)	F1-SEEN	F1-UNSEEN
BU ($mc = 64$, $nr = 2$)	122.32	56.81%	45.33%
BU+MFF×4	175.52	61.06%	48.97%
BU+MFF×4+MCA	197.76	**62.38%**	**51.45%**
BU+MFF×3	174.59	59.43%	47.62%
MU ($mc = 32$, $nr = 4$)	52.84	53.68%	42.41%
MU+MFF×4+MCA	112.80	**59.27%**	**51.06%**

We evaluated the denoising performance by varying two UNet hyperparameters, `model-channels` (mc = 32, 64, 128) and `num-resblocks` (nr = 2, 4), while maintaining the original architecture. Using Paradigm-2 with standard training, we report the results in Table. I (ID=2). Experiments demonstrate an optimal balance between model complexity and performance. The configuration (mc=64, nr=2) achieves optimal performance while minimizing both parameters and computational cost.

To validate the acceleration effect of Alg. 3, we designed the experiments in Table. III. The second column indicates whether the Alg. 3 is employed, where '✗' denotes sampling using Alg. 2 with both sampling and training steps set to 1000. The 'S' represents the number of substeps in Alg. 3. It can be observed that the smaller the value of S, the faster the sampling. Compared to the 1000-step denoising process, the proposed Alg. 3 significantly reduces the processing time (200× faster) while incurring only a minor performance loss (max to 7%). Table. III further demonstrate that our proposed method of biasing the noise initialization process using an obstacle map (P2) significantly outperforms the process of sampling from unbiased uniform noise (P1).

Next, we conduct robustness experiments on Paradigm-3. We obtain coarse predictions with varying levels of noise by incompletely training the NEED/RGP model. In Table. III, 'ep5' indicates training for 5 epochs. Specifically, the configuration 'NEED (ep10) (T)' denotes that we first save the incompletely trained NEED model and preserve its coarse predictions for each map as the training set for DiffRP-P3 (where 'T' stands for trained). The notation '55.32→61.36' indicates that the F1 score of the NEED (ep10) model is 55.32%, while the score of the further trained DiffRP-P3 based on this model reaches to 61.36%. The configuration '(UT)' indicates that the DiffRP model was not specifically trained on the coarse predictions from predictor model. Instead, we used the DiffRP model trained on NEED (ep10) to evaluate its performance improvement when fed with the coarse predictions from NEED (ep5). The results demonstrate that the DiffRP-P3 model can refine the noisier coarse predictions to some extent, though the performance gain is limited. Notably, when DiffRP-P3 encounters coarse prediction inputs that were not seen during training, its performance degrades below that of DiffRP-P1. However, it is worth emphasizing that DiffRP-P3 still leverages its learned path-planning capability to refine the coarse predictions, achieving a 10% performance improvement.

To systematically assess the Map-conditioned denoising model, we implemented DiffRP-P2 as detailed in Table. IV. Here, 'BU' denotes the base UNet configuration, while 'MU' represents the mini UNet. They utilize `model-channels`=64 and 32, and `num-resblock`=2 and 4, respectively.

The configuration 'BU + MFF×4 + MCA' is depicted as in Fig. 2. By progressively removing the MFF modules from lower to higher levels and eliminating the MCA module, we obtain the MFF×3 configurations.

Table V: Performance Impact of Different Loss Functions

Loss	F1-SEEN	F1-UNSEEN
$\mathcal{L}_2$	62.38%	51.45%
$\mathcal{L}_p$	64.59%	54.40%

Table. IV shows that model accuracy in predicting promising regions improves with increasing MFF modules. Notably, the MCA module significantly enhances performance on the UNSEEN testset, demonstrating that the spatial-channel attention mechanism can improve model generalizability. Despite comparable parameter counts, the 'MU+MFF×4+MCA' configuration surpassed the BU model, achieving a 1.5% improvement on the UNSEEN scenarios This indicates that our proposed MFF and MCA modules yield greater performance gains than conventional denoising model scaling.

The loss objective plays a critical role in shaping the learned distribution of diffusion models. Our study employs the L2 loss (Eq. 14), while recent research has demonstrated that perceptual loss can enhance sample quality [25]. We further examines the impact of structural perceptual loss on DiffRP performance. For brevity, mathematical details are omitted. The Table. V indicate that the perceptual loss ($\mathcal{L}_p$) yields a performance improvement of 3% to 5% for the DiffRP-P2 model. After years of development, the field of diffusion models has produced numerous advanced techniques for refinement. We posit that applying these techniques to DiffRP could further enhance its performance.

V. Conclusions

We propose DiffRP, a diffusion-based approach for generating promising regions. Specifically, we propose three distinct denoising paradigms: 1) standard denoising from uniform Gaussian noise, 2) obstacle map-biased Gaussian noise initialization, and 3) noise biasing using predictions from a coarse predictor. Experimental results on public benchmarks demonstrate that our method significantly improves promising region prediction accuracy in unseen environments, achieving a state-of-the-art F1-score of 51.45%. Furthermore, we validate that more accurate promising regions substantially enhance the performance of sampling-based path planning algorithms (exemplified by RRT*). When equipped with identical non-uniform samplers, our DiffRP-RRT* exhibits superior performance compared to other neural-network-driven approaches.

References

[1] Z. Bai, H. Pang, Z. He, B. Zhao, and T. Wang, "Path planning of autonomous mobile robot in comprehensive unknown environment using deep reinforcement learning," *IEEE Internet of Things Journal*, 2024.

[2] T. Liao, F. Chen, Y. Wu, H. Zeng, S. Ouyang, and J. Guan, "Research on path planning with the integration of adaptive a-star algorithm and improved dynamic window approach," *Electronics*, vol. 13, no. 2, p. 455, 2024.

[3] D. Zhang, C. Chen, and G. Zhang, "AGV path planning based on improved a-star algorithm," in *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 7. IEEE, 2024, pp. 1590–1595.

[4] R. Zhang, H. Guo, D. Andriukaitis, Y. Li, G. Królczyk, and Z. Li, "Intelligent path planning by an improved rrt algorithm with dual grid map," *Alexandria Engineering Journal*, vol. 88, pp. 91–104, 2024.

[5] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, no. 2, pp. 267–274, 2010.

[6] H. Huang, L. Jin, and Z. Zeng, "A momentum recurrent neural network for sparse motion planning of redundant manipulators with majorization-minimization," *IEEE Transactions on Industrial Electronics*, 2025.

[7] M. Liu, Y. Li, Y. Chen, Y. Qi, and L. Jin, "A distributed competitive and collaborative coordination for multirobot systems," *IEEE Transactions on Mobile Computing*, 2024.

[8] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural RRT*: Learning-based optimal path planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.

[9] J. Wang, J. Liu, W. Chen, W. Chi, and M. Q.-H. Meng, "Robot path planning via neural-network-driven prediction," *IEEE transactions on artificial intelligence*, vol. 3, no. 3, pp. 451–460, 2021.

[10] N. Ma, J. Wang, J. Liu, and M. Q.-H. Meng, "Conditional generative adversarial networks for optimal path planning," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 2, pp. 662–671, 2021.

[11] Y. Huang, C.-T. Tsao, and H.-H. Lee, "Efficiency improvement to neural-network-driven optimal path planning via region and guideline prediction," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1851–1858, 2024.

[12] J. J. Johnson, U. S. Kalra, A. Bhatia, L. Li, A. H. Qureshi, and M. C. Yip, "Motion planning transformers: A motion planning framework for mobile robots," *arXiv preprint arXiv:2106.02791*, 2021.

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

[14] F. Liu, H. Yu, J. Huang, X. Hua, H. Liu, Z. Ma, Z. Wang, Z. Du, and L. Sun, "RefineCatDiff: Toward high-quality medical image segmentation via a categorical diffusion refinement framework," *Advanced Intelligent Systems*, p. 2401125, 2025.

[15] S. Yang, X. Chen, and J. Liao, "Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 3190–3199.

[16] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[17] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International conference on machine learning*. PMLR, 2021, pp. 8162–8171.

[18] T. Amit, T. Shaharbany, E. Nachmani, and L. Wolf, "SegDiff: Image segmentation with diffusion probabilistic models," *arXiv preprint arXiv:2112.00390*, 2021.

[19] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[20] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International journal of computer vision*, vol. 129, pp. 3051–3068, 2021.

[21] J. Peng, Y. Liu, S. Tang, Y. Hao, L. Chu, G. Chen, Z. Wu, Z. Chen, Z. Yu, Y. Du *et al.*, "PP-liteseg: A superior real-time semantic segmentation model," *arXiv preprint arXiv:2204.02681*, 2022.

[22] Y. Ji, K. Yun, Y. Liu, Z. Xie, and H. Liu, "Neural-network-driven reward prediction as a heuristic: Advancing q-learning for mobile robot path planning," *arXiv preprint arXiv:2412.12650*, 2024.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] S. Li, T. Hu, J. van de Weijer, F. Shahbaz Khan, T. Liu, L. Li, S. Yang, Y. Wang, M.-M. Cheng *et al.*, "Faster diffusion: Rethinking the role of the encoder for diffusion model inference," *Advances in Neural Information Processing Systems*, vol. 37, pp. 85 203–85 240, 2024.

[25] S. Lin and X. Yang, "Diffusion model with perceptual loss," *arXiv preprint arXiv:2401.00110*, 2023.