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

RUSH: Recursive and Scalable 3D Coarse to Fine Path Planning

Hwajung Lee, *Graduate Student Member, IEEE*, Daegeol Ko, *Graduate Student Member, IEEE*, Jaehyuk Hur, Junwon Lee, Seongbo Ha, *Graduate Student Member, IEEE*, Jong Hwan Ko, *Member, IEEE*, and Hyeonwoo Yu, *Member, IEEE*

***Abstract*—Path planning in large-scale, complex 3D environments is fundamentally constrained by a trade-off between path quality and computational speed. This paper presents RUSH (Recursive and Scalable 3D *Coarse To Fine* Path Planning), a hierarchical framework that resolves this trade-off. RUSH decomposes the long-range planning task into a coarse plan followed by fine-grained, independent subproblems that can be solved in parallel. These subproblems are addressed by a unified, diffusion-based network that refines an initial estimate path by learning its residual to an optimal path. This approach allows RUSH to leverage rich geometric information directly from 3D voxel maps without being bottlenecked by the full map's complexity. We validate our method on large-scale outdoor (KITTI, MulRan) and indoor (HM3D) datasets, each spanning a 200 m× 200m× 6m map. Experimental results demonstrate that RUSH generates feasible, high-quality paths with remarkable efficiency, achieving up to a 12.59× speedup over a hierarchically accelerated A∗ baseline, while maintaining a path cost within 24% of the optimal solution. This performance gain positions RUSH as a powerful and practical solution for applications requiring rapid global path planning in large-scale 3D maps.**

***Index Terms*—Path planning, coarse to fine, diffusion.**

I. Introduction

PATH planning is a fundamental component in autonomous navigation, responsible for generating an optimal and efficient path from a start to a goal location. This task becomes exceptionally challenging in large-scale and geometrically complex 3D environments, where planners face a fundamental trade-off between path quality and computational speed. Planning directly on a high-resolution 3D map provides complete spatial information but often entails prohibitive computational costs. For instance, a 200 m × 200 m × 6m map with a 0.2 m voxel resolution creates a search space of over 20 million voxels, which renders planners like A∗ [10] and Rapidly-exploring Random Trees Star (RRT∗) [14] too slow for applications requiring rapid global replanning.

To address this computational burden, several strategies have been proposed to simplify the environment by projecting it onto 2D maps [13], [35] or 2.5D representations [30], [31]. While this improves speed, the inherent loss of 3D geometric information can lead to suboptimal or physically infeasible paths, especially in vertically complex terrains. This limitation underscores the importance of planning directly on 3D maps to ensure optimal navigation with complete spatial data, particularly in complex scenarios involving cluttered obstacles, multi-level terrain, or dynamic agents [1], [27]. However, this shift to direct 3D planning introduces a critical scalability challenge, especially for modern learning-based planners that promise high-speed inference. Many such approaches either require processing the entire map as a single, high-dimensional input, a task that is infeasible for large maps. To remain efficient, DiPPeST [28] falls back on 2D-centric representations, thereby reintroducing information loss. Consequently, existing learning-based models often struggle to generalize from limited training environments to expansive real-world scenarios. Meanwhile, another line of research seeks to make classical search algorithms more intelligent. A prominent example, Wavestar [25], pioneers an adaptive refinement strategy that focuses computational effort on critical regions requiring high-detail navigation.

In this paper, we propose RUSH, a hierarchical *Coarse To Fine* (CTF) path planner designed to address the scalability challenge of learning-based methods in 3D. This CTF strategy reframes the problem into two stages: a coarse plan to establish the global route, and a fine-grained decomposition into smaller, independent subproblems. To address both the initial coarse plan and the subsequent fine-grained subproblems, we introduce a single, unified diffusion-based network capable of planning across multiple resolutions. This approach creates a powerful synergy: our CTF decomposition allows the network to be executed on all subproblems in a massively parallel fashion, fully leveraging the computational power of GPUs to achieve remarkable inference speed. Through comprehensive experiments, we demonstrate that RUSH successfully reconciles computational efficiency with high path quality in large-scale 3D planning. The main contributions of this paper are summarized as follows:

- Direct and Scalable 3D Planning: A learning-based framework that efficiently operates directly on large-scale 3D voxel maps (e.g., 200 m × 200 m × 6 m), overcoming the

Received 4 September 2025; accepted 12 December 2025. Date of publication 12 January 2026; date of current version 19 January 2026. This article was recommended for publication by Associate Editor J. Zhang and Editor P. Vasseur upon evaluation of the reviewers' comments. This work was supported in part by the National Research Foundation of Korea (NRF) under Grant RS-2024-00359937, in part by the Ministry of Trade, Industry and Energy under Grant RS-2024-00508191, Grant RS-2025-25463654, and Grant P008922, and in part by IITP under Grant RS-2025-10692981 and Grant RS-2025-25442569. *(Corresponding author: Hyeonwoo Yu.)*

Hwajung Lee, Daegeol Ko, Jaehyuk Hur, Seongbo Ha, and Hyeonwoo Yu are with the Department of Intelligent Robotics Sungkyunkwan University, Suwon 16419, South Korea (e-mail: uniesrev0a_@skku.edu; kjs4753_@skku.edu; jack1008_@skku.edu; sobo3607_@skku.edu; hwyu_@skku.edu).

Junwon Lee and Jong Hwan Ko are with the School of Electronic and Electrical Engineering, Sungkyunkwan University, Suwon 16419, South Korea (e-mail: jwlee1218_@skku.edu; jhko_@skku.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2026.3653375, provided by the authors.

Digital Object Identifier 10.1109/LRA.2026.3653375

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

information loss and physical infeasibility inherent in 2D transformation methods.
- Hierarchical *Coarse To Fine* Strategy: A recursive decomposition strategy that partitions large-scale planning into parallelizable subproblems, enabling long-range scalability and reducing computational complexity.
- Diffusion-Based 3D Path Generation: Application of a diffusion model to efficiently refine path segments, generating optimal final paths.
- Single-Network Multi-Resolution Planning: A unified network architecture that generalizes the path refinement task across both coarse global and fine-grained local resolutions, enhancing overall efficiency by eliminating the need for separate models.

II. Related Work

Path planning is often performed on 2D maps for computational efficiency. But this simplification discards geometric information critical for navigating environments with vertical complexity, such as multi-level structures or uneven terrain [13], [35]. To mitigate this, various 2.5D representations like Digital Elevation Maps (DEMs) and 2.5D Normal Distributions Transform (NDT) maps are employed to provide a richer description of terrain [30], [31]. While offering more spatial awareness, these methods remain approximations of the true 3D space, resulting path quality often highly dependent on the specific conversion heuristics employed.

These limitations motivate the need for direct 3D path planning, which leverages complete spatial information for optimal navigation. Classical algorithms like A$*$ [10] and RRT$*$ [14] are theoretically optimal and can operate directly on 3D voxelized representations. However, their practical application is severely hampered by computational complexity. The search space for 3D A$*$ expands exponentially with map size, while sampling-based planners like RRT$*$ often require an impractically large number of samples to achieve optimality [26]. Recent GPU-accelerated methods like GTMP [17] utilize tensorized structures to speed up search operations. However, despite such acceleration, generating high-quality global paths in vast, complex 3D environments remains challenging without heuristic guidance.

To overcome these imitations, learning-based planners have emerged as faster alternatives. Approaches like Neural A$*$ [32] and iA$*$ [6] integrate neural networks to effectively prune the 2D search space. Generative diffusion models represent another promising direction, existing works predominantly focus on specific sub-problems such as local planning [33], mapless navigation from sensor streams [19], or robotic manipulation in constrained environments [4], [36]. Consequently, the application of diffusion models to map-based large-scale 3D global navigation remains limited.

Addressing the challenge of scalability in large and complex environments often involves hierarchical planning, which decomposes problems into manageable steps. For example, CFA$*$ [18] first plans an abstract path on a low-resolution map to drastically reduce the search space for a subsequent refinement stage that finds a detailed path on the high-resolution map. This principle is also applied to learning-based planners like Hierarchical Diffuser [5], which uses a high-level planner to generate sparse subgoals for a low-level planner. However, its applications have been confined to 2D domains or have relied on separate, independent networks for each stage. Alternatively, search-based approaches like Wavestar [25] dynamically refine search resolution in critical regions. While effective, the core challenge remains in optimizing the search strategy itself to balance path quality and speed in large-scale 3D environments.

III. Method

Our proposed method, RUSH, is a hierarchical path planner designed for efficient navigation in large-scale 3D voxel maps. The core of our approach is built upon two key components: a CTF strategy that recursively decomposes the planning problem, and a diffusion-based model that refines the resulting paths to optimality. This hierarchical process is formalized using a principled probabilistic framework to integrate information across multiple resolutions.

A. Problem Formulation

We define a path $\boldsymbol{x}$ as a sequence of N waypoints in a given 3D map $\boldsymbol{z}$, denoted as $\boldsymbol{x} = [x^0, x^1, \dots, x^{N-2}, x^{N-1}] = [x^{start}, x^1, \dots, x^{N-2}, x^{goal}]$, where x^i is the i^{th} waypoint in 3D coordinates. Accurate path planning necessitates high-resolution 3D maps, yet planning directly on them is often computationally intractable. To address this, we employ a hierarchical representation where $\boldsymbol{z}_k$ denotes the map at resolution level k, and $\boldsymbol{z}_{0:k}$ includes all map information up to that level. Even when a ground-truth optimal path $\boldsymbol{x}_k^{gt}$ is available for a map up to that resolution level k, learning its underlying distribution $p(\boldsymbol{x}_k^{gt}|\boldsymbol{z}_{0:k}, x^{start}, x^{goal})$ directly remains a formidable challenge.

To address this, we employ a hierarchical CTF strategy. The core idea is recursive: a path planned at a coarser resolution level, $k-1$, serves as a structural prior to guide planning at the next finer resolution level, k. Let $\boldsymbol{x}_{k-1}=[x_{k-1}^0, \cdots, x_{k-1}^{N-1}]$ denote the path at level $k-1$, generated on a low-resolution map $\boldsymbol{z}_{k-1}$. To generate a more detailed path $\boldsymbol{x}_k$ on a higher-resolution map $\boldsymbol{z}_k$, this coarse path $\boldsymbol{x}_{k-1}$ is exploited as a prior. Specifically, each pair of adjacent waypoints from the coarse path, $\{(x_{k-1}^i, x_{k-1}^{i+1}) \mid i = 0, 1, \cdots, N-2\}$, defines the start/goal of an independent subproblem at the finer level k. Thus, the full path $\boldsymbol{x}_k$ can be expressed as a composition of smaller path segments:

$$\boldsymbol{x}_k = [\underbrace{x_{k-1}^0}_{\text{prior}}, x_k^1, \cdots, x_k^{N-2}, \underbrace{x_{k-1}^1}_{\text{prior}}, \cdots, \underbrace{x_{k-1}^{N-2}}_{\text{prior}}, x_k^{(N-1)^2-(N-2)}, \cdots, x_k^{(N-1)^2-1}, \underbrace{x_{k-1}^{N-1}}_{\text{prior}}]. \quad (1)$$

As illustrated in Fig. 1, each of these $N-1$ subproblems is also formulated to generate a path of N waypoints. This consistency is key to enabling the use of a single, unified network across all hierarchical levels. This hierarchical decomposition transforms a single large-scale planning task into $N-1$ smaller, independent subproblems. By leveraging the coarse path $\boldsymbol{x}_{k-1}$ as a prior, these subproblems can be solved in parallel, significantly enhancing computational efficiency and reducing memory requirements.

B. Recursive Bayesian Path Estimation

We frame the hierarchical planning process as a recursive Bayesian filtering problem. Our goal is to estimate the posterior path distribution $p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k})$ at each resolution level k. This

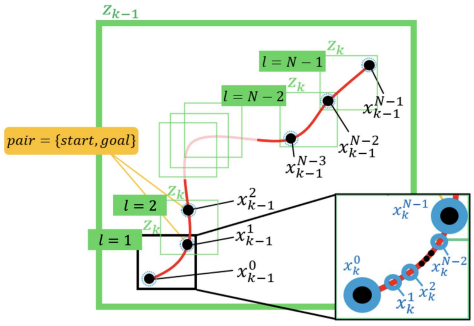

Fig. 1. Our CTF strategy for hierarchical path planning and parallel processing. A path at a coarser level $k-1$ is decomposed into segments. Each adjacent waypoint pair from the coarse path, $(x_{k-1}^i, x_{k-1}^{i+1})$, defines start/goal points for an independent subproblem at the next finer level k.

posterior represents our belief about the optimal path given the accumulated map information.

To formalize this, we introduce two assumptions:

1) Hierarchical Markov Assumption: The path at finer level k is conditionally independent of past map observations $\boldsymbol{z}_{0:k-1}$ given the coarse path $\boldsymbol{x}_{k-1}$. This assumes the coarse path sufficiently encapsulates all relevant information from prior levels.
2) Segment Conditional Independence: Given the waypoints from the coarse path, the generation of each fine-grained path segment is conditionally independent of the other segments.

Under these assumptions, the full posterior distribution over the final path at the highest resolution K can be expressed as a product over all hierarchical levels and path segments:

$$\begin{aligned}\boldsymbol{x}_K &\sim p(\boldsymbol{x}_K|\boldsymbol{z}_{0:K}) = \prod_{k=1}^{K} p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k}) \\ &\propto \prod_{k=1}^{K} p(\boldsymbol{z}_k|\boldsymbol{x}_k) \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{0:k-1})\, d\boldsymbol{x}_{k-1} \\ &= \prod_{k=1}^{K}\prod_{l=1}^{N-1} \underbrace{p(\boldsymbol{z}_k|\boldsymbol{x}_{k,l})}_{\text{likelihood}} \int \underbrace{p(\boldsymbol{x}_{k,l}|\boldsymbol{x}_{k-1})}_{\text{state transition}} \underbrace{p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{0:k-1})}_{\text{prior}} d\boldsymbol{x}_{k-1},\end{aligned} \tag{2}$$

where $\boldsymbol{x}_{k,l} = [x_k^{(l-1)\cdot(N-1)}, \cdots, x_k^{l\cdot(N-1)}]$ is the l^{th} sub-path at level k. An overview is illustrated in Fig. 2.

Each term has a clear interpretation:

- Prior $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{0:k-1})$: The posterior belief of the path from the preceding coarser level $k-1$.
- State Transition $p(\boldsymbol{x}_{k,l}|\boldsymbol{x}_{k-1})$: A probabilistic model for generating a fine-grained path segment from coarse path.
- Likelihood $p(\boldsymbol{z}_k|\boldsymbol{x}_{k,l})$: An environmental compatibility of the path segment with the higher-resolution map $\boldsymbol{z}_k$.

While Eq. (2) describes the complete formulation, its core mechanism is a recursive update from one level to the next. The integral, which marginalizes all possible coarse paths $\boldsymbol{x}_{k-1}$, is computationally intractable. To overcome this, we employ a Maximum a Posteriori (MAP) approximation. Instead of integrating over the entire belief space, we condition our prediction on the single most probable path from the coarser level, $\hat{\boldsymbol{x}}_{k-1} = \text{argmax}_{\boldsymbol{x}_{k-1}} p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{0:k-1})$. This simplifies the intractable integral to:

$$\int p(\boldsymbol{x}_{k,l}|\boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{0:k-1})\, d\boldsymbol{x}_{k-1} \approx p(\boldsymbol{x}_{k,l}|\hat{\boldsymbol{x}}_{k-1}).$$

This approximation reduces our task to defining a proposal distribution for an initial path, $\boldsymbol{x}_k^{init}$, for each subproblem conditioned on the start and goal waypoints $(\hat{x}_{k-1}^i, \hat{x}_{k-1}^{i+1})$ from the MAP coarse path. This initial path serves as the starting point for refinement. We model this proposal as a high-dimensional Gaussian, $p(\boldsymbol{x}_k^{init}) \approx \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$, and explore three strategies for defining its mean $\boldsymbol{\mu}$:

1) Zero-Mean Proposal: A naive baseline that assumes zero-mean Gaussian distribution, $\boldsymbol{\mu} = 0$. This provides no directional bias, compelling the model to infer structure solely from the environmental context and boundary conditions.

$$p(\boldsymbol{x}_k^{init}) \approx \mathcal{N}(\boldsymbol{x}_k^{init};\, 0,\, \sigma^2 I)\,, \tag{3}$$

where σ quantifies the initial uncertainty.

2) Linear Proposal: A more informed approach that provides a strong directional bias by setting the mean $\boldsymbol{\mu}$ to be a linear interpolation between the coarse waypoints, assuming a uniform likelihood over the feasible space.

$$p(\boldsymbol{x}_k^{init}) \approx \mathcal{N}(\boldsymbol{x}_k^{init};\, (\alpha \hat{x}_{k-1}^i + (1-\alpha)\hat{x}_{k-1}^{i+1})\,,\, \sigma^2 I)\,, \tag{4}$$

where $\alpha \in [0, 1]$ is the interpolation factor.

3) Heuristic-Guided Proposal: Complex environments often contain multiple, distinct feasible paths, making the true path distribution inherently multimodal. Modeling this full multimodal distribution is complex. Instead, we use a classical planner (e.g., Greedy Search, Rapidly-exploring Random Trees [16]) as a heuristic to quickly find a single feasible path candidate, $\boldsymbol{x}_k^{cand}$. This candidate represents one dominant mode of the true distribution. We then model our proposal as a Gaussian by setting the mean $\boldsymbol{\mu}$ to be this candidate, effectively committing to one promising path for refinement.

$$p(\boldsymbol{x}_k^{init}) \approx \mathcal{N}(\boldsymbol{x}_k^{init}; \boldsymbol{x}_k^{cand}, \sigma^2 I)\,. \tag{5}$$

This strategy leverages the ability of classical heuristics to find a feasible initial path in complex environments, focusing the refinement process on a high-potential path.

C. Path Distribution Refinement

Using the estimation model described in Section III-B, the initial path distribution $p(\boldsymbol{x}_k^{init}|\boldsymbol{z}_{0:k})$ was estimated in a CTF manner. However, this initial path distribution is a rough approximation and differs from the optimal ground-truth path distribution $p(\boldsymbol{x}_k^{gt}|\boldsymbol{z}_{0:k}, x^{start}, x^{goal})$. To bridge this gap, we further refine the initial path distribution to align with the ground-truth path distribution as follows:

$$p(\boldsymbol{x}_k^{refine}|\boldsymbol{z}_{0:k}) = \mathcal{F}_\theta\big(\, p(\boldsymbol{x}_k^{init}|\boldsymbol{z}_{0:k})\,\big)\,, \tag{6}$$

where $\mathcal{F}_\theta$ is a *learnable functional* which minimizes:

$$D_{KL}(\, p(\boldsymbol{x}_k^{refine}|\boldsymbol{z}_{0:k}) \,\|\, p(\boldsymbol{x}_k^{gt}|\boldsymbol{z}_{0:k}, x^{start}, x^{goal})\,).$$

1) Diffusion-Based Path Refinement: We exploit Denoising Diffusion Probabilistic Model (DDPM) [11] for our $\mathcal{F}_\theta$. In DDPM, the forward diffusion is performed by gradually adding Gaussian noise to the training data, after which the original

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

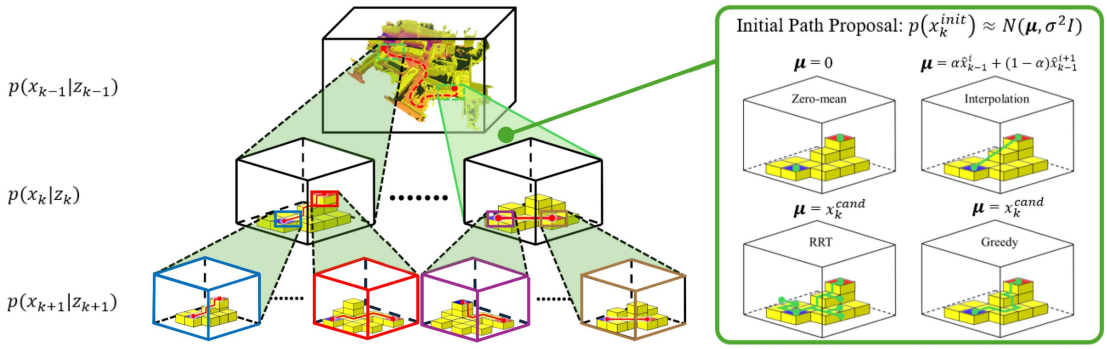

Fig. 2. Overview of our hierarchical estimation framework. This process is modeled within a Bayesian filtering framework as described in Eq. (2). The path belief is propagated via a state transition model (which generates an initial fine path from the coarse prior) and updated by a likelihood model (which evaluates the path's feasibility in the higher resolution map $\boldsymbol{z}_k$). This process operates recursively, where the refined path estimate $\hat{\boldsymbol{x}}_k$ is decomposed to formulate subproblems for the subsequent level $k+1$.

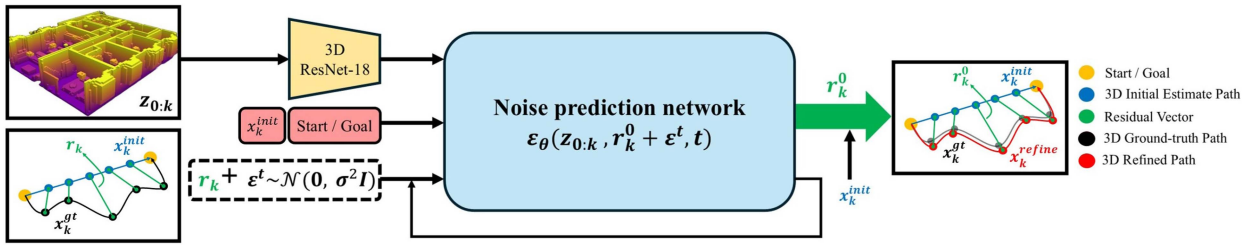

Fig. 3. Architecture of the RUSH network. A 3D ResNet vision encoder extracts spatial features from the input voxel crop map. These features, start/goal points, and initial estimate path $\boldsymbol{x}_k^{init}$, serve as conditions for a 1D UNet. The UNet inputs a noisy residual vector $\boldsymbol{r}_k$ and is trained to predict the added noise ϵ. By subtracting the predicted noise, the model recovers the refined residual $\boldsymbol{r}_k^0$, which is added to the initial path to produce an optimal final path.

data distribution is restored through the reverse diffusion process that removes this noise. Instead of directly diffusing the entire path $\boldsymbol{x}_k$, our model learns to predict the residual vector $\boldsymbol{r}_k = \boldsymbol{x}_k^{gt} - \boldsymbol{x}_k^{init}$, defined as the distribution difference between ground-truth $\boldsymbol{x}_k^{gt}$ and initial estimate path $\boldsymbol{x}_k^{init}$ from Section II-I-B [34]. Therefore, model can effectively refine $p(\boldsymbol{x}_k^{init}|\boldsymbol{z}_{0:k})$ into $p(\boldsymbol{x}_k^{gt}|\boldsymbol{z}_{0:k}, x^{start}, x^{goal})$.

2) DDPM Training: Our diffusion model is trained on the residual vector $\boldsymbol{r}_k$. DDPM takes the residual vector $\boldsymbol{r}_k^t$ with added noise ϵ^t as input and iteratively recovers the noise-free residual vector $\boldsymbol{r}_k^0$ through t denoising iterations as follows:

$$\boldsymbol{r}_k^{t-1} = \alpha\left(\boldsymbol{r}_k^t - \gamma\,\epsilon_\theta\left(\boldsymbol{z}_{0:k}, \boldsymbol{r}_k^t, k\right) + \mathcal{N}\left(0, \sigma^2 I\right)\right), \quad (7)$$

where ϵ_θ is noise prediction network, and $\mathcal{N}(0, \sigma^2 I)$ represents Gaussian noise. The noise schedule parameters α, γ, σ vary with the iteration step t, following the Square Cosine Schedule [22]. For training, the DDPMScheduler [11] was used with 100 diffusion steps. The refined path is obtained as $\boldsymbol{x}_k^{refine} = \boldsymbol{x}_k^{init} + \boldsymbol{r}_k^0$, which is guided to approximate the optimal path $\boldsymbol{x}_k^{gt}$. ϵ_θ is trained to predict the original noise ϵ^t from the residual vector $\boldsymbol{r}_k^t$ with added noise. The training loss for the residual-based diffusion is defined as:

$$\mathcal{L}_{res} = \text{MSE}\left(\epsilon^t,\ \epsilon_\theta\left(\boldsymbol{z}_{0:k}, \boldsymbol{r}_k^0 + \epsilon^t, t\right)\right). \quad (8)$$

Through this process, the model learns to transform an initial estimate path distribution $p(\boldsymbol{x}_k^{init}|\boldsymbol{z}_{0:k})$ into a refined path distribution $p(\boldsymbol{x}_k^{refine}|\boldsymbol{z}_{0:k})$ that approximates the ground-truth path distribution $p(\boldsymbol{x}_k^{gt}|\boldsymbol{z}_{0:k}, x^{start}, x^{goal})$.

An overview of this process is illustrated in Fig. 3.

IV. Implementation

A. Hierarchical Inference Pipeline

During inference, RUSH employs a two-stage process for large-scale environments, executing a single CTF step.

1) Stage 1. Coarse Path Refinement: First, a large-scale, high-resolution map ($H_{fine} \times W_{fine} \times D_{fine}$ at R_{fine} resolution) is resized to its coarse equivalent ($H_{coarse} \times W_{coarse} \times D_{coarse}$ at R_{coarse} resolution). This coarse map, along with the overall start/goal points, is used to generate an initial estimate path $\boldsymbol{x}_{k-1}^{init}$ as described in Section III-B. This initial path is then fed into our unified network for refinement. Specifically, the start/goal waypoints are fixed using an inpainting technique [12]. The network then performs a conditioned reverse diffusion process for T iterations using a DPMSolverMultistepScheduler [21] to predict the refined residual $\boldsymbol{r}_{k-1}^0$. The final coarse path is obtained by adding this residual to the initial path, $\boldsymbol{x}_{k-1}^{refine} = \boldsymbol{x}_{k-1}^{init} + \boldsymbol{r}_{k-1}^0$, resulting in a sequence of N waypoints that outlines the general path.

2) Stage 2. Fine Path Refinement (Parallel): As conceptualized in Fig. 1, the generated coarse path $\boldsymbol{x}_{k-1}$ is then broken

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

down into $N-1$ segments. Each adjacent pair of waypoints $(x_{k-1}^{i}, x_{k-1}^{i+1})$ defines an independent subproblem. For each subproblem, we crop a fine-resolution local map from the original high-resolution map, centered around the segment. These $N-1$ subproblems are processed simultaneously in parallel. For each, an initial path $\boldsymbol{x}_k^{init}$ is generated, and then the same diffusion-based refinement process described in Stage 1 is applied to produce a refined path segment $\boldsymbol{x}_k^{refine}$. Finally, all $N-1$ refined path segments are concatenated to form the complete, high-resolution path, which approximates the ground-truth path distribution $p(\boldsymbol{x}_k^{gt}|\boldsymbol{z}_{0:k}, x^{start}, x^{goal})$.

B. Network Architecture

RUSH addresses large-scale path planning by refining an initial path distribution using a DDPM. Our approach has the ability to solve multi-scale problems within a single, unified network, leveraging the CTF strategy. Our network uses a 3D ResNet-18 [29] as a vision encoder to extract a spatial feature map from the input voxel crop map. This feature map serves as a global condition for the Noise Prediction Network, which adopts a ConditionalUnet1D architecture to perform conditional noise prediction [7]. The UNet is conditioned via Feature-wise Linear Modulation (FiLM) [23], using a feature vector derived from the concatenation of the global map features, the start/goal coordinates, and the initial estimate path $\boldsymbol{x}_k^{init}$. It takes a noisy residual vector $(\boldsymbol{r}_k^0 + \epsilon^t)$ as input and is trained end-to-end to predict the added noise ϵ^t. By predicting this noise, the model learns to reverse the diffusion process and recover the refined residual $\boldsymbol{r}_k^0$. This single network architecture was used for all experiments across all scales and environments (indoor/outdoor, coarse/fine), highlighting its robustness and generalizability.

C. Training Dataset Generation

The robustness and generalization of our model stem from its diverse training dataset of real-world 3D environments.

1) Map Dataset: For outdoor environments, we utilized sequences 0, 4, 6, and 7 from the Semantic KITTI dataset [3] and the DCC and KAIST subsets from the MulRan dataset [15]. For indoor environments, we employed the HM3D dataset [24]. To enable our single network strategy, all maps were cropped onto a uniform voxel grid of $100\times100\times20$ at two distinct resolutions. This grid size was selected to balance the competing demands of representing large-scale environments and capturing geometric detail within our network's computational limits. Within this fixed grid dimension, we varied the voxel resolution to address the different requirements of our hierarchical approach. In our experiments, we used the following configurations:

- Coarse Maps (R_{coarse}): 2 m $\times$ 2 m $\times$ 0.3 m voxel size, representing 200 m $\times$ 200 m $\times$ 6 m physical area for high-level, large-scale outdoor navigation.
- Fine Maps (R_{fine}): 0.2 m $\times$ 0.2 m $\times$ 0.3 m voxel size, representing 20 m $\times$ 20 m $\times$ 6 m physical area for detailed indoor/outdoor path generation.

Empirically, the 0.2m fine resolution is critical to resolve essential features like stairs for feasibility, while the 2m coarse resolution optimally balances global topology with GPU tractability. A single CTF step proves sufficient, as intermediate resolutions offer diminishing returns in resolving geometric complexity. Our final training set contains 7,000 maps, balanced between indoor and outdoor scenarios.

2) Path Dataset: Ground-truth paths were generated using a constrained 3D A$*$ (26-connectivity including diagonals). To ensure physical plausibility for ground vehicles, each waypoint must have an occupied ground voxel within two voxels below it [8]. Paths were standardized to 16 uniformly spaced waypoints. The final dataset includes the multi-scale 3D voxel crop map, start/goal points, ground-truth A$*$ path $\boldsymbol{x}_k^{gt}$, and an initial estimate path $\boldsymbol{x}_k^{init}$ generated via one of the methods described in Section III-B.

V. Experiments

We conducted a series of experiments to validate the performance of RUSH in large-scale 3D environments. The evaluation is designed to assess three key aspects: computational speed, path optimality in terms of cost, and the physical feasibility of the generated paths.

A. Experimental Setup

1) Test Environments: Our evaluation is performed on a challenging test set of 1,500 scenarios constructed from large-scale maps not seen during training. Each map covers a 200 m $\times$ 200 m $\times$ 6 m area, initially represented as a high-resolution voxel grid of $H_{fine}\times W_{fine}\times D_{fine} = 1000 \times 1000 \times 20$. For the first stage of our hierarchical pipeline, this map is resized to a coarse representation of $H_{coarse}\times W_{coarse}\times D_{coarse}=100\times100\times20$. To ensure a comprehensive assessment of generalization, the test set was equally divided among three environment types: purely outdoor (Outdoor), indoor-to-outdoor transitions (In-Outdoor), and outdoor-to-indoor transitions (Out-Indoor).

2) Parameters: To specifically challenge long-range planning capabilities, start/goal points in each scenario were at least 280 m apart. In this context, we set the number of waypoints N to 16, a value selected to balance structural guidance against subproblem complexity; fewer waypoints create complex segments for the diffusion model, whereas more yield redundant density without improving guidance. After one CTF step, this expands into a detailed path of $(N-1)^2+1=226$ waypoints, resulting in an empirically observed average separation of around 1.6m. This granularity provides sufficient detail for a ground robot to execute smooth trajectories without being overly dense. For the diffusion process during inference, the number of iterations T was set to 30, and we used Karras sigmas for step sizes in the noise schedule with a DPMSolver order of 2.

3) Baselines: We compare RUSH against a set of classical and learning-based planners to provide a comprehensive performance analysis.

- 3D A$*$: A constrained A$*$ planner operating directly on the full-resolution map. This serves as the ground-truth for path optimality ($R_{\text{cost}} = 1.00$).
- 3D A$*$ w/ CTF: The same A$*$ algorithm applied within our CTF framework. This baseline isolates the performance gain of the hierarchical strategy and serves as our primary benchmark for computation time ($R_{\text{time}} = 1.00$).
- Informed RRT$*$ [9]: A 3D implementation of the optimized sampling-based planner.
- GTMP [17]: A GPU-accelerated planner utilizing random multipartite graphs. We evaluate configurations (N, M, H) –waypoints per layer, total layers, and probing density– to

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

TABLE I
COMPARISON OF PLANNING METHODS ACROSS LARGE-SCALE 3D ENVIRONMENTS

Category	Method	Variant	CTF	Time (s) ↓	R_{time} ↑	SOR/HOR (%) ↓	R_{cost} ↓	Smooth (rad) ↓	Clear (m) ↑
Classical	A* (3D)	–	✗	504.24	0.004	0 / 0	**1.00/0.00**	0.17/0.10	0.32/0.05
			✓	2.03	**1.00**	0 / 0	**1.00/0.03**	**0.14/0.08**	0.36/0.07
	Informed RRT* (3D)	–	✗	86.16	0.01	73 / 79	1.13/0.15	0.62/0.17	2.34/1.51
			✓	42.85	0.05	36 / 72	1.30/0.24	0.64/0.05	1.21/0.61
	GTMP (3D)	(30, 200, 10)	✗	**0.003**	**676.67**	80 / 99	24.00/3.34	1.59/0.04	**23.43/12.25**
		(500, 10, 200)	✗	0.055	36.91	43 / 65	1.12/0.20	1.01/0.28	10.13/4.95
		(1000, 10, 1000)	✗	0.66	3.06	**35 / 56**	1.02/0.20	0.68/0.20	8.00/4.33
Learning-based	Neural A* (2D)	–	✓	0.78	2.26	– / 72	15.12/3.77	0.72/0.28	0.40/0.32
	iA* (2D)	–	✓	0.73	2.46	– / 71	10.74/4.25	0.40/0.33	**1.61/1.51**
	RUSH (Ours)	Zero-mean	✓	**0.16**	12.25	**14 / 35**	1.24/0.15	**0.26/0.08**	0.54/0.45
		Interpolation	✓	**0.16**	**12.59**	18 / 40	1.24/0.16	0.30/0.08	0.66/0.52
		Greedy	✓	2.90	0.70	20 / 41	1.25/0.17	0.30/0.09	0.79/0.72
		RRT	✓	0.79	2.56	25 / 44	**1.23/0.18**	0.33/0.08	0.79/0.48

TABLE II
COMPARISON WITH WAVESTAR ON THE `Park` SEQUENCE [37]

Method	Variant	CTF	Time (s) ↓	R_{time} ↑	SOR/HOR (%) ↓	R_{cost} ↓
Wavestar	–	✗	28.69	0.13	0 / 0	**1.000**
	Lazy	✗	7.40	0.50	0 / 0	1.001
	Fast	✗	3.67	**1.00**	0 / 0	1.001
RUSH (Ours)	Zero-mean	✓	**0.18**	**20.18**	**1 / 7**	**1.066**
	Interpolation	✓	0.18	20.12	4 / 10	1.140
	Greedy	✓	0.37	9.95	2 / 6	1.074
	RRT	✓	2.42	1.52	8 / 15	1.164

analyze the trade-off between probabilistic completeness and speed.
- Neural A∗ [32] & iA∗ [6]: Two representative learning-based 2D planners, applied within our CTF framework to a 2D projection of the 3D environment.
- RUSH (Ours): Four variants of our method, each utilizing a different initial path as described in Section III-B3: Zero-mean, Interpolation, Greedy, and RRT.

4) Evaluation Metrics:
- Time Ratio (R_{time}): Speedup relative to the 3D A∗ w/ CTF baseline, defined as $T_{\text{A* w/ CTF}}/T_{\text{method}}$.
- Cost Ratio (R_{cost}): Path length relative to the optimal 3D A∗, defined as $C_{\text{method}}/C_{\text{A* w/o CTF}}$ [14] (mean ± std).
- Feasibility (SOR/HOR): Evaluated via Soft (SOR) and Hard (HOR) Off-road Rates [2]. A waypoint is feasible if a road surface voxel exists along the z-axis. SOR is the percentage of infeasible waypoints. HOR is a stricter metric, measuring the percentage of fine path segments containing at least one infeasible waypoint.
- Smoothness & Clearance (Smooth, Clear): We measure Smoothness (rad), the average angle between three consecutive waypoints, and Clearance (m), the average distance to the nearest obstacle [20] (mean ± std).

All experiments were conducted on an AMD Ryzen 9 7900 CPU and an NVIDIA RTX 4090 GPU. For planners failing to generate path in some scenarios, metrics were computed only on the subset of successful trials to ensure a fair comparison.

B. Results and Discussion

The aggregated results in Tables I & II, and examples in Figs. 4 & 5, show that RUSH effectively balances speed, path optimality, and feasibility in large-scale environments.

1) Effectiveness of the Hierarchical Framework: The proposed CTF strategy proves highly effective across different planning methods. For RUSH, the parallelized design is critical, yielding an 8.6× speedup over sequential execution (e.g., 1.38s to 0.16s). Similarly, applying this framework to 3D A∗ resulted in a 248× speedup (504.24s to 2.03s), as the coarse prior confines the search volume, bypassing the exponential node expansion inherent in 3D grids. However, while CTF improved Informed RRT∗ success rate (35% to 99%), its computation time remained high (42.85s), highlighting the persistent challenge of efficient sampling in vast 3D spaces.

2) Limitations of 2D Planners: The performance of learning-based 2D planners underscores the necessity of direct 3D planning. In complex environments with indoor-outdoor transitions and multi-level structures, Neural A∗ and iA∗ achieved limited success rates of 66% and 70%, respectively. These failures stem from losing vertical geometric information. Even when complete paths were found (1% and 8% of scenarios, respectively), exceptionally high cost ratios prove 2D projections insufficient for complex 3D navigation.

3) Comparison With GTMP: While GTMP achieves exceptional speed in its lightest configuration (0.003 s), it suffers from low feasibility (80% SOR). Tuning it for a high-quality configuration significantly improves path cost and feasibility but increases computation time to 0.66 s. Although RUSH (Zero-mean) is slower than the lightest GTMP, it is 4× faster than the high-quality variant while achieving substantially better feasibility (14% SOR). This advantage stems from RUSH's perception-aware framework, which leverages learned spatial features from a 3D vision encoder, whereas GTMP's reliance on blind graph sampling.

4) RUSH Performance Evaluation: RUSH offers a compelling speed-quality trade-off. Fastest variants achieve 12× speedup over the accelerated 3D A∗ w/ CTF baseline while maintaining path costs within 24% of the optimum, producing consistent paths with SOR/HOR distributions heavily skewed toward zero. Scalability is confirmed by near-zero correlation between path length and computation time, though feasibility errors moderately correlate with length due to the increased probability of failures in longer sequences. Regarding smoothness, RUSH generates natural transitions between parallelized segments. Furthermore, lightweight B-spline post-processing efficiently ensures C^2 continuity (0.26 rd to 0.13 rd) without compromising speed or feasibility.

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

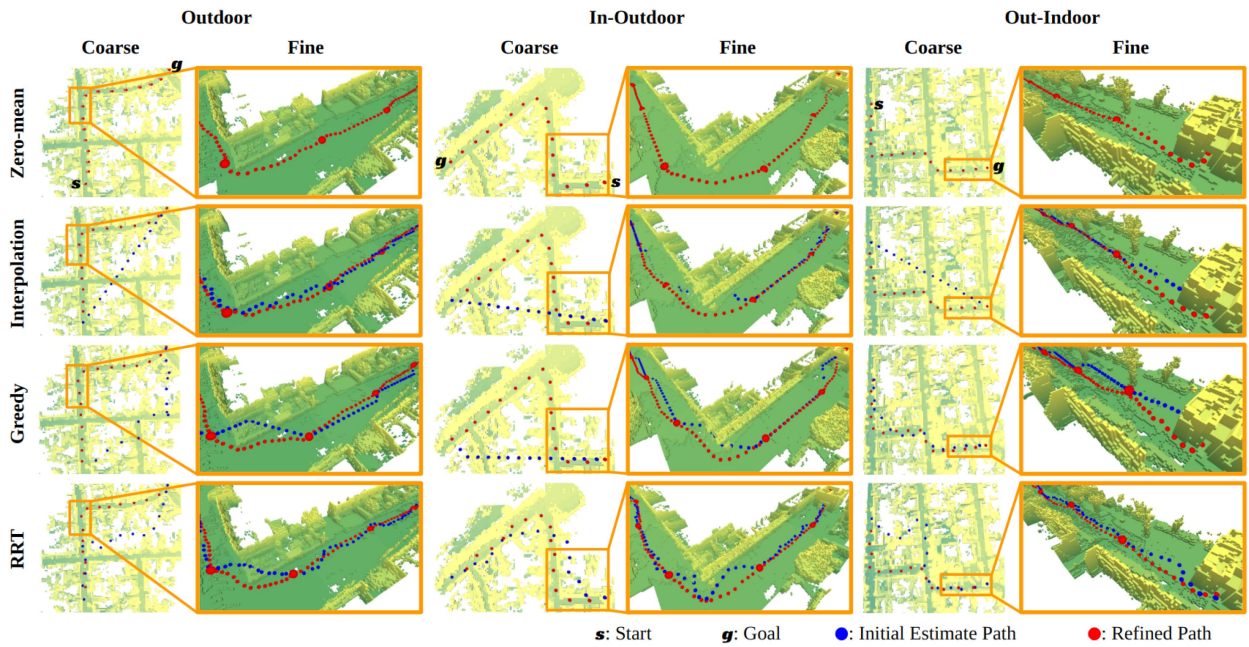

Fig. 4. Results of the RUSH across diverse large-scale 3D environments. The initial estimate path (blue waypoints) is transformed by our diffusion model into the refined, optimal path (red waypoints). The model consistently generates feasible paths, successfully refining initial estimates by leveraging the 3D environmental context.

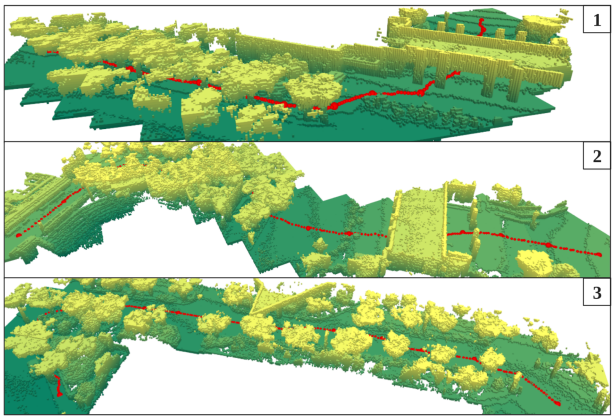

Fig. 5. RUSH navigating vertically complex 3D environments.

5) Comparison With Wavestar [25]: We conducted a separate evaluation on the `Park` sequence from the Newer College Dataset [37]. We compared RUSH against Wavestar variants (Standard, Lazy, Fast). In Table II, where R_{cost} and R_{time} are normalized against the Standard (best quality) and Fast (best speed) Wavestar variants respectively, RUSH (Zero-mean) achieves a 20.18× speedup over the fastest Wavestar variant with only a 6.6% increase in path cost. This confirms RUSH as an efficient solution for rapid global replanning.

C. Ablation Study: Impact of Initial Path Proposal

Our analysis reveals that the initial path proposal critically impacts performance. Notably, the RUSH (Zero-mean) variant –using a naive, unbiased proposal– produced the most feasible paths while being fastest. Heuristic-guided variants (Greedy, RRT), showed slightly lower feasibility and higher average times due to outliers requiring longer heuristic searches. Their median times remained fast (0.21 s and 0.29 s), confirming the overall efficiency of the RUSH framework.

We hypothesize that this result stems from the restrictive nature of heuristic priors. While guaranteeing initial feasibility, a path from a classical heuristic can commit the diffusion model to a specific, potentially suboptimal homotopy class. Consequently, the refinement process becomes constrained by this strong structural bias. Conversely, the zero-mean proposal imposes no such bias, compelling the model to generate paths solely based on the geometric features learned by its 3D vision encoder. The superior performance of this unbiased variant highlights a key strength of our framework: RUSH exhibits intrinsic robustness to local optima. Rather than refining a precise initial guess, the model leverages learned spatial contexts to effectively generate feasible paths from pure noise. This demonstrates that RUSH avoids the initialization sensitivity, proving its capability to generate globally coherent paths directly from learned representations.

VI. Conclusion & Limitations

We presented RUSH, a hierarchical framework balancing path quality and computational speed in large-scale 3D planning. By integrating a recursive CTF strategy with a unified diffusion-based refinement model, RUSH generates long-range, feasible paths significantly faster than baselines. Despite these advantages, we acknowledge limitations opening avenues for future research. First, our current implementation uses uniformly spaced waypoints, which may not optimally capture complex geometric features. Future work will explore non-uniform waypoint distributions that dynamically adjust density based on path curvature to enhance geometric fidelity. Second, while highly efficient, parallelized refinement can occasionally lead to ill-posed subproblems (observed in 8.3% of cases) due to

infeasible coarse waypoints. To mitigate this, we propose generating multiple coarse path candidates in a single batch to ensure that subsequent fine-grained subproblems are formulated with feasible endpoints. Furthermore, we plan to incorporate constraint-aware guidance functions and investigate collision-free blending techniques at segment boundaries to ensure seamless transitions between sub-paths. Finally, to address feasibility errors that correlate with increasing path lengths, we aim to fully exploit the inherent multimodality of diffusion models. By sampling diverse path candidates rather than a single MAP estimate, RUSH can avoid homotopy lock-in and provide robust, high-quality options for downstream tasks in complex environments.

REFERENCES

[1] I. Babataev, A. Fedoseev, N. Weerakkodi, E. Nazarova, and D. Tsetserukou, "HyperGuider: Virtual reality framework for interactive path planning of quadruped robot in cluttered and multi-terrain environments," in *Proc. IEEE Int. Conf. Syst. Man, Cybern.*, 2022, pp. 2037–2042.

[2] M. Bahari et al., "Vehicle trajectory prediction works, but not everywhere," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17123–17133.

[3] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.

[4] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 1916–1923.

[5] C. Chen, F. Deng, K. Kawaguchi, C. Gulcehre, and S. Ahn, "Simple hierarchical planning with diffusion," 2024, *arXiv:2401.02644*.

[6] X. Chen, F. Yang, and C. Wang, "iA*: Imperative learning-based A* search for path planning," 2024, *arXiv:2403.15870*.

[7] C. Chi et al., "Diffusion policy: Visuomotor policy learning via action diffusion," *Int. J. Robot. Res.*, vol. 44, 2025, pp. 1684–1704.

[8] J. Frey, D. Hoeller, S. Khattak, and M. Hutter, "Locomotion policy guided traversability learning using volumetric representations of complex environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., 2022, pp. 5722–5729.

[9] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT$*$: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2997–3004.

[10] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. TSSC-4, no. 2, pp. 100–107, Jul. 1968.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *in Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 6840–6851.

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

[13] K. Kamarudin et al., "Method to convert kinect's 3D depth data to a 2D map for indoor slam," in *Proc. IEEE 9th Int. Colloq. Signal Process. Appl.*, 2013, pp. 247–251.

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[15] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "MulRan: Multimodal range dataset for urban place recognition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6246–6253.

[16] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects: Steven M. LaValle, Iowa State University, A James J. Kuffner, Jr., University of Tokyo, Tokyo, Japan," in *Algorithmic Computat. Robot.* Boca Raton, FL, USA: CRC Press, 2001, pp. 303–307.

[17] A. T. Le et al., "Global tensor motion planning," *IEEE Robot. Automat. Lett.*, vol. 10, no. 7, pp. 7302-7309, Jul. 2025.

[18] J.-Y. Lee and W. Yu, "A coarse-to-fine approach for fast path finding for mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 5414–5419.

[19] J. Liang, A. Payandeh, D. Song, X. Xiao, and D. Manocha, "DTG: Diffusion-based trajectory generation for mapless global navigation," in *Proc. 2024 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 5340–5347.

[20] S. Liu and P. Liu, "Benchmarking and optimization of robot motion planning with motion planning pipeline," *Int. J. Adv. Manuf. Technol.*, vol. 118, no. 3, pp. 949–961, 2022.

[21] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps," *in Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 5775–5787.

[22] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8162–8171.

[23] E. Perez, F. Strub, H. D. Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 3942–3951.

[24] S. K. Ramakrishnan et al., "Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI," 2021, *arXiv:2109.08238*.

[25] V. Reijgwart, C. Cadena, R. Siegwart, and L. Ott, "Efficient hierarchical any-angle path planning on multi-resolution 3D grids," in *Proc. Robot. Sci. Syst.*, Los Angeles, CA, USA, Jul. 2025, Art. no. 49.

[26] S. Russell, P. Norvig, and A. Intelligence, "A Modern Approach," *Artif. Intell.*, vol. 25, no. 27, pp. 79–80, 1995.

[27] C. Scheidemann et al., "Obstacle-avoidant leader following with a quadruped robot," 2024, *arXiv:2410.00572*.

[28] M. Stamatopoulou, J. Liu, and D. Kanoulas, "DiPPeST: Diffusion-based path planner for synthesizing trajectories applied on quadruped robots," in *Proc. 2024 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 7787–7793.

[29] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6450–6459.

[30] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *Proc. 2016 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1184–1189.

[31] S. Yang, S. Yang, and X. Yi, "An efficient spatial representation for path planning of ground robots in 3D environments," *IEEE Access*, vol. 6, pp. 41539–41550, 2018.

[32] R. Yonetani, T. Taniai, M. Barekatain, M. Nishimura, and A. Kanezaki, "Path planning using neural A$*$ search," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12029–12039.

[33] W. Yu et al., "LDP: A local diffusion planner for efficient robot navigation and collision avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 5466–5472.

[34] Z. Yue, J. Wang, and C. C. Loy, "ResShift: Efficient diffusion model for image super-resolution by residual shifting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, vol. 36, pp. 13294–13307 .

[35] A. Yusefi, A. Durdu, and C. Sungur, "ORB-SLAM-based 2D reconstruction of environment for indoor autonomous navigation of UAVs," *Avrupa Bilim ve Teknoloji Dergisi*, Special Isuue, pp. 466–472, 2020.

[36] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations," in *Proc. Robot. Sci. Syst.*, Delft, Netherlands, Jul. 2024.

[37] L. Zhang, M. Camurri, D. Wisth, and M. Fallon, "Multi-camera LiDAR inertial extension to the newer college dataset," 2021, *arXiv:2112.08854*.