M^{2}Diffuser: Diffusion-based Trajectory Optimization for Mobile Manipulation in 3D Scenes

Sixu Yan, Zeyu Zhang, *Member, IEEE*, Muzhi Han, Zaijin Wang, Qi Xie, Zhitian Li, Zhehan Li, Hangxin Liu, *Member, IEEE*, Xinggang Wang, *Senior Member, IEEE*, and Song-Chun Zhu, *Fellow, IEEE*

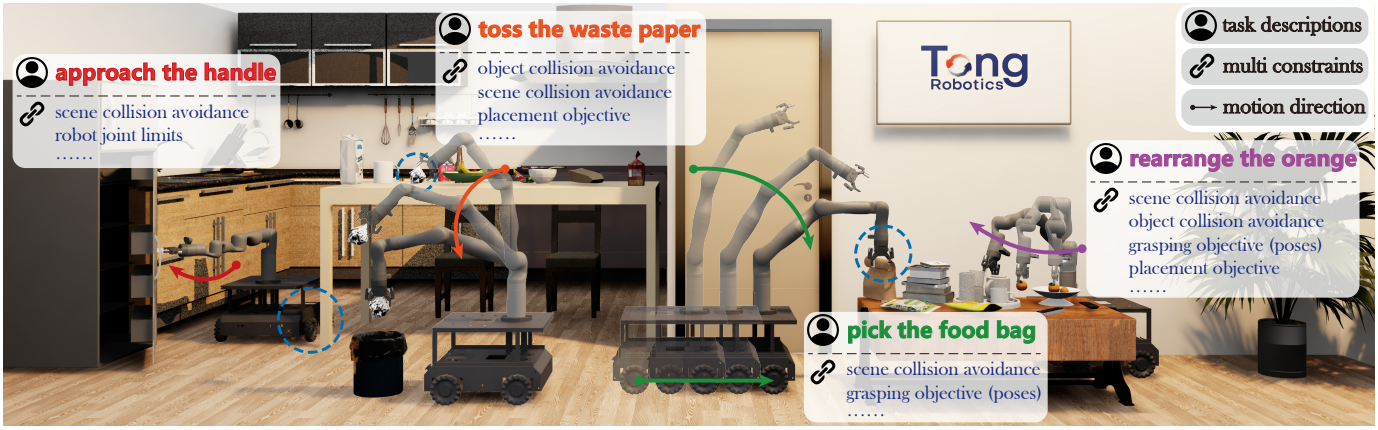


Fig. 1. Challenges in mobile manipulation. Mobile manipulation plays a pivotal role in core robotic tasks such as object grasping, placement, rearrangement, and articulated object manipulation. Each of these tasks poses distinct challenges, requiring the motion generator to holistically account for constraints posed by environmental geometry and robot embodiment to accomplish task-specific objectives.

***Abstract*—Recent advances in diffusion models have opened new avenues for research into embodied AI agents and robotics. Despite significant achievements in complex robotic locomotion and skills, mobile manipulation—a capability that requires the coordination of navigation and manipulation—remains a challenge for generative AI techniques. This is primarily due to the high-dimensional action space, extended motion trajectories, and interactions with the surrounding environment. In this paper, we introduce M^{2}Diffuser, a diffusion-based, scene-conditioned generative model that directly generates coordinated and efficient whole-body motion trajectories for mobile manipulation based on robot-centric 3D scans. M^{2}Diffuser first learns trajectory-level distributions from mobile manipulation trajectories provided by an expert planner. Crucially, it incorporates an optimization module that can flexibly accommodate physical constraints and task objectives, modeled as cost and energy functions, during the inference process. This enables the reduction of physical violations and execution errors at each denoising step in a fully differentiable manner. Through benchmarking on three types of mobile manipulation tasks across over 20 scenes, we demonstrate that M^{2}Diffuser outperforms state-of-the-art neural planners and successfully transfers the generated trajectories to a real-world robot. Our evaluations underscore the potential of generative AI to enhance the generalization of traditional planning and learning-based robotic methods, while also highlighting the critical role of enforcing physical constraints for safe and robust execution. Videos, code and more details are available at https://m2diffuser.github.io.**

***Index Terms*—Mobile Manipulation, Embodied AI, Diffusion Model, Trajectory Generation and Optimization**

The work was done when Sixu Yan interned at BIGAI. *(Corresponding authors: Xinggang Wang and Hangxin Liu.)*

Sixu Yan and Xinggang Wang are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yansixu@hust.edu.cn; xgwang@hust.edu.cn).

Zeyu Zhang, Zaijin Wang, Qi Xie, Hangxin Liu and Song-Chun Zhu are with the State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI), Beijing 100080, China (e-mail: zhangzeyu@bigai.ai; wangzaijin@bigai.ai; xieqi@bigai.ai; liuhx@bigai.ai; sczhu@bigai.ai).

Muzhi Han is with the Center for Vision, Cognition, Learning, and Autonomy (VCLA), Statistics Department, University of California, Los Angeles (UCLA), USA (e-mail: muzhihan@ucla.edu).

Zhitian Li is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China, and also interned at BIGAI (e-mail: lizhitian1998@buaa.edu.cn).

Zhehan Li is with the School of Artificial Intelligence, Xidian University, Xi'an 710126, China, and also interned at BIGAI (e-mail: zhehanli_robot@stu.xidian.edu.cn).

Song-Chun Zhu is also with the School of Artificial Intelligence and the Institute for Artificial Intelligence, Peking University, Beijing 100871, China.

I. Introduction

RESEARCH into Embodied Artificial Intelligence (EAI) increasingly emphasizes interaction with the environment, progressing from passive observation in learning visual navigation [1], [2] to active manipulation in object rearrangement [3], [4], and more recently, to integrate large foundation models to tackle highly interactive tasks [5], [6], [7], [8], [9]. However, mobile manipulation [10]—a core capability enabling agents to perform a wide range of tasks across large spaces—remains challenging for EAI agents.

The key difficulty in solving mobile manipulation tasks is the need to jointly account for agent embodiment, large-scale environment geometry, and task-specific objectives and constraints. For instance, as illustrated in Fig. 1, when approaching and grasping an object, the success of the agent's motion execution depends not only on its own configuration but also on the state of its surroundings along its movements. Furthermore, even when picking the same object, the variations in context require

task objectives tailored to ensure that the agent's base position allows its arm to reach the object, the arm can avoid collisions with the environment, and the end effector can achieve a specific pose to execute the desired grasp, which all eventually lead to various types of motion constraints for the agent. However, the interdependencies—and sometimes conflicts—among these constraints present significant challenges for both traditional planning-based methods and learning-based approaches to solve the corresponding motion generation problems.

Encoding task objectives and related motion constraints implicitly in demonstration data or carefully designed reward or loss functions [11], [12], [13], [14], Imitation Learning (IL) and Reinforcement Learning (RL) are usually used for EAI agents to learn sophisticated skills [15], [16], [17], [18], [19], [20], [21], complex locomotion [4], [22], [23], [24], [25], whole-body motion [26], [27], [28], or advanced policies for long-horizon tasks [29], [30], [31]. However, they often struggle to fully eliminate violations of physical constraints in model inference. Moreover, expensive new data collection and model re-training are typically required to incorporate new task requirements. On the other hand, the field of robotics has a long history of developing planning and control methods to ensure robust and efficient executions for robots, a more concrete form of EAI agents. With substantial modeling techniques, various constraints can be formulated for robots to accomplish complex mobile manipulation tasks through whole-body control [32], [33], [34] and base-arm coordination [35], [36], [37]. However, these approaches heavily rely on perfect knowledge of the environment [38], [39] and engineered goal proposals (*e.g.*, grasp poses) [40], [41], limiting their scalability in real-world deployments.

Recently, generative AI has demonstrated the remarkable ability to produce diverse and even novel content in text [42], images [43], [44], and videos [45], [46] with high spatial and temporal consistency. However, the success of generative AI techniques has not yet been demonstrated in complex robotic tasks like mobile manipulation, primarily due to (i) the high dimensionality of the solution space, which requires efficient modeling and high-quality training data, and (ii) the strict requirement for physically precise execution, which demands high-fidelity model outputs.

In this paper, we explore leveraging generative modeling techniques to produce holistic mobile manipulation motion that not only coordinates navigation and manipulation for obstacle avoidance, but also strictly satisfies task objectives with high precision, such as grasping objects. Specifically, we propose the Mobile Manipulation Diffuser (M^2Diffuser), a scene-conditioned diffusion model that takes robot-centric 3D scans to generate whole-body coordinated mobile manipulation motion. With the guided sampling mechanism inherent to diffusion models, M^2Diffuser incorporates explicit physical constraints (*e.g.*, joint limits, scene collision and motion smoothness), as well as implicit task objectives (*e.g.*, grasping pose selection), as cost and energy functions during its inference process. These functions are differentiable, effectively guiding the optimization of sampled trajectories to reduce physical violations and execution errors.

To develop M^2Diffuser, we first collect high-quality training data in simulated scenes by using an expert motion planner to generate whole-body mobile manipulation trajectories with smooth base-arm coordination. Second, we train M^2Diffuser by using robot-centric 3D scans, *i.e.*, local point clouds represented in the robot's base coordinate, as model conditioning to improve scalability. By evaluating M^2Diffuser in both a physics-based simulator and real-world environments, we demonstrate that M^2Diffuser significantly outperforms state-of-the-art neural motion planners in generating robot motion for mobile manipulation tasks, where the robot must navigate toward and grasp 15 types of target objects. Furthermore, we show that the architecture of M^2Diffuser is flexible enough to be adopted to new mobile manipulation tasks, such as object placement and goal reaching. Our findings reveal two key insights for generative AI and EAI: (i) For motion generation, diffusion-based models offer a promising alternative to classical motion planning approaches, which require extensive prior knowledge and manual design, as well as to learning-based autoregressive planning approaches, which struggle to ensure physical safety and precise executions; and (ii) Even SOTA generative AI techniques, when trained with high-quality data from expert mobile manipulation planners, are still insufficient to guarantee safe execution. Effective enforcement of physical constraints during the generation process is critical for success in complex robotic applications.

A. Related Work

Motion Generation in 3D Scenes: Generating robot motion requires understanding object geometry and scene context. Various 3D representations can be derived from raw observations, such as point clouds [14], [47], [48], [49], voxels [47], [50], [51], [52], [53], and implicit fields [54], which support the learning of dexterous skills [47] and object manipulation [48], [51], [52], [54], [55]. Similarly, neural planners have been developed to imitate expert planner behaviors based on 3D representations of the environment [14], [49], [50]. While these methods accelerate motion generation, they primarily model robot motion generation as an autoregressive process, which struggles to capture complex trajectory distribution. In this work, the proposed M^2Diffuser learns trajectory-level distributions directly through a diffusion process, mitigating the weakness of existing methods in learning high-dimensional trajectory generation for mobile manipulation.

Diffusion Models in Robotics: Due to their advantages in modeling multi-modal data distributions with stable training, diffusion models have been widely applied to robotic tasks like stationary manipulation [47], [48], [54], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], autonomous navigation [68], [69], [70], quadruped locomotion [71], drone flight [72], dexterous manipulation [73], [74], [75], and mechanical structures [76], [77]. However, applying diffusion models to mobile manipulation tasks requires high-quality training data, which is difficult to obtain. In this work, we leverage an expert planner from our previous research [35], [36] to collect a large set of whole-body mobile manipulation trajectories and use them to unlock the capability of diffusion models in complex robotic tasks.

Learning-based Trajectory Optimization: Trajectory optimization [78], [79] has enabled robots to generate smooth and efficient movements. However, the requirement to explicitly define the objective and constraints limits the scalability of these techniques in complex tasks and challenging real-world environments. Existing work has addressed this limitation by applying deep learning to: (i) learn task goals such as grasp poses [80], [81] and object affordance [53], [82], [83], (ii) learn implicit objectives and constraint functions for grasping [63], [81], [84] and collision avoidance [85], [86], [87], and (iii) learn neural motion planners from demonstrations for generating collision-free [14], [62] or kinematically feasible motion [14], [59], [62], [88]. In this work, M^{2}Diffuser draws inspiration from these efforts and aims to tackle mobile manipulation. It directly learns trajectory-level distributions from expert demonstrations, followed by guided trajectory optimization with differentiable cost functions of physical constraints and task-oriented energy functions, which can be either explicitly defined or learned from data.

Scene Understanding in Mobile Manipulation: Existing IL- and RL-based household mobile manipulation primarily relies on egocentric or overhead observations, including RGB [20], RGB-D [21], [89], [90], and depth [4], [19], [91] images. This setup provides only partial visual input, insufficient for capturing object-geometric and scene-spatial information, leading to poor understanding of constraints (*e.g.*, occluded obstacles) between the robot embodiment and the environment. In this work, M^{2}Diffuser uses robot-centric 3D scans as visual input. Such 3D scans encode detailed geometric and spatial relationships around the robot, enhancing the model's understanding for physical constraints and task objectives. This design benefits decision-making and safe motion execution in cluttered 3D environments.

B. Contribution

To our knowledge, ours is the first work that applies diffusion models to solve robotic mobile manipulation tasks. It makes four major contributions:

1) We propose M^{2}Diffuser, the first scene-conditioned motion generator tailored for mobile manipulation in EAI. It seamlessly integrates multiple physical constraints and flexibly handles different task objectives, and directly generates highly coordinated whole-body motion trajectories with physical plausibility from 3D scans.
2) We highlight the importance of integrating physical constraints and task objectives into the generative process via a guided optimization mechanism, which ensures physical plausibility and task completion of the generated motion.
3) We demonstrate that the diffusion-based planner, compared to previous autoregressive planners, is better suited for generating high-dimensional mobile manipulation motion. It ensures spatial and temporal consistency of the generated trajectories.
4) We also show that taking local 3D scans around the robot as model visual input can be more effective for real-world generalization and deployment.

C. Overview

The remainder of this paper is organized as follows. In Sec. II, we define the problem and provide a detailed introduction to our method. Sec. III presents the experimental setup and compares M^{2}Diffuser with baseline models across three mobile manipulation tasks in various simulated 3D environments. In Sec. IV, we validate our method in real-world 3D household settings. Finally, Sec. V discusses the limitations and outlines potential directions for future research, followed by Sec. VI where we conclude the paper.

II. Mobile Manipulation Diffuser

A. Problem Statement and Diffusion Model

Given the robot-centric 3D scan of a scene $\mathcal{S}$, M^{2}Diffuser aims to generate an efficient and coordinated trajectory that fulfills the task objective $\mathcal{O}$, enabling the robot to complete tasks such as object grasping, object placement, or reaching a target pose without physical violations. We denote the trajectory as $\boldsymbol{\tau} = (\mathbf{q}_0, \cdots, \mathbf{q}_i, \cdots, \mathbf{q}_H)$, where $\mathbf{q}_i \in \mathbb{R}^d$ is the robot's joint position and the trajectory is discreted by the task horizon H. Here we assume that a low-level controller can robustly drive the robot's configuration $\mathbf{q}_i$ to $\mathbf{q}_{i+1}$ as long as they are physically feasible.

Diffusion model [92] consists of a forward and a reverse diffusion process, respectively corresponding to the model training and inference. During the forward process $q(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{t-1})$, the initial data $\boldsymbol{\tau}_0 \sim q(\boldsymbol{\tau}_0)$ sampled from the dataset is perturbed by adding gradually-decreased noise, which eventually turns the data into Gaussian noise $\boldsymbol{\tau}_T$. In the reverse process, the data is reconstructed from $\boldsymbol{\tau}_T$ following an iterative denoising process with learned Gaussian kernels. M^{2}Diffuser formulates mobile manipulation as trajectory optimization and solves it with the spirit of optimization as inference, *i.e.*, by sampling the trajectory-level distribution learned by the diffusion model. Leveraging the diffusion model with loss-guided sampling and flexible conditioning, M^{2}Diffuser models the probability of mobile manipulation trajectory conditioned on 3D scan $\mathcal{S}$ and objective $\mathcal{O}$ as:

$$p(\boldsymbol{\tau}_0 \mid \mathcal{S}, \mathcal{O}) = \int p(\boldsymbol{\tau}_T \mid \mathcal{S}, \mathcal{O}) \prod_{t=1}^{T} p(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t, \mathcal{S}, \mathcal{O}) \, \mathrm{d}\boldsymbol{\tau}_{1:T}, \tag{1}$$

where T denotes the maximum time step in diffusion process, and $p(\boldsymbol{\tau}_T \mid \mathcal{S}, \mathcal{O})$ is a standard Gaussian distribution. To sample from $p(\boldsymbol{\tau}_0 \mid \mathcal{S}, \mathcal{O})$, we must iteratively sample from the conditional distribution $p(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t, \mathcal{S}, \mathcal{O})$, which follows

$$\begin{aligned} p(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t, \mathcal{S}, \mathcal{O}) =& \frac{p_\theta(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t, \mathcal{S}) \, p_\phi(\mathcal{O} \mid \boldsymbol{\tau}_{t-1}, \mathcal{S})}{p(\mathcal{O} \mid \mathcal{S})} \\ \propto& p_\theta(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t, \mathcal{S}) \, p_\phi(\mathcal{O} \mid \boldsymbol{\tau}_{t-1}, \mathcal{S}). \end{aligned} \tag{2}$$

B. Trajectory Generation via Conditional Diffusion

$p_\theta(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t, t, \mathcal{S})$ represents the probability of generating scene-conditioned trajectory $\boldsymbol{\tau}_{t-1}$ at denoising step t and is independent of task objective $\mathcal{O}$. In this work, we model it using a scene-conditioned diffusion model similar to [62]. According

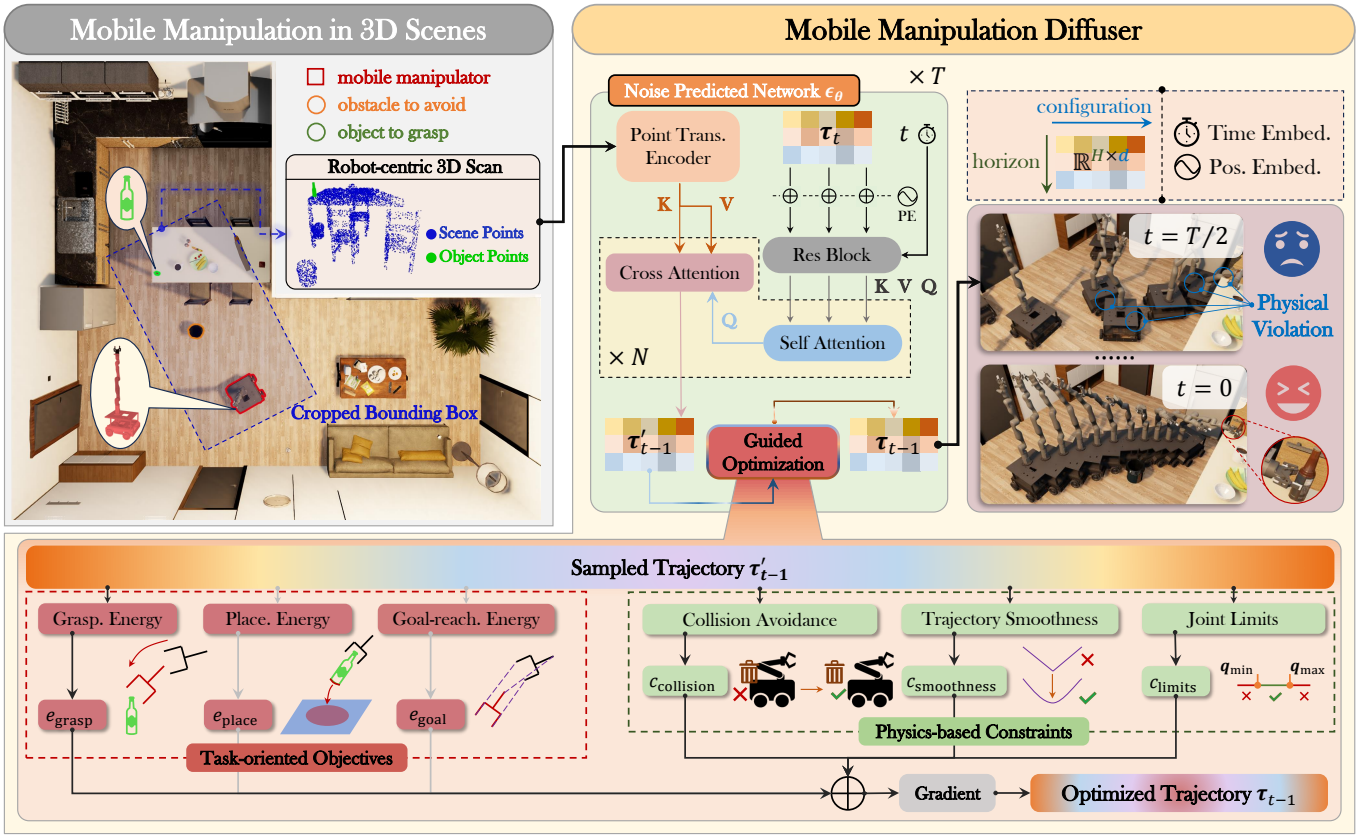


Fig. 2. Overview of the M^{2}Diffuser, a diffusion-based motion planner designed to sample and optimize whole-body coordinated trajectories directly from natural 3D scans, efficacious for mobile manipulation in 3D scenes. Using robot-centric 3D scans as visual input, M^{2}Diffuser employs an iterative denoising process to generate task-specific trajectories. It optimizes the sampled results at each denoising diffusion step guided by cost and energy functions, ensuring physical plausibility and task completion of generated trajectories.

to the formulation of the diffusion model in [92], it can be written as

$$p_\theta\left(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t, \mathcal{S}\right) = \mathcal{N}\left(\boldsymbol{\tau}_{t-1}; \boldsymbol{\mu}_\theta\left(\boldsymbol{\tau}_t, t, \mathcal{S}\right), \boldsymbol{\Sigma}_\theta\left(\boldsymbol{\tau}_t, t, \mathcal{S}\right)\right). \tag{3}$$

For simplicity, we only learn the mean $\boldsymbol{\mu}_\theta$, while the covariance $\boldsymbol{\Sigma}_\theta$ is decided by noise schedules. Ingeniously, [92] formulates $\boldsymbol{\mu}_\theta$ as

$$\boldsymbol{\mu}_\theta\left(\boldsymbol{\tau}_t, t, \mathcal{S}\right) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{\tau}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{\tau}_t, t, \mathcal{S}\right)\right), \tag{4}$$

where α_t and $\bar{\alpha}_t$ are defined by noise schedules in the forward process [92], [93], [94]. Then $\boldsymbol{\mu}_\theta$ can be learned with a noise prediction network $\boldsymbol{\epsilon}_\theta$ via a MSE loss:

$$\begin{aligned}\mathcal{L}_\theta\left(\boldsymbol{\tau}_0 \mid \mathcal{S}\right) &= \mathbb{E}_{t,\boldsymbol{\epsilon},\tau_0}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\boldsymbol{\tau}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathcal{S}\right)\right\|^2\right] \\ &= \mathbb{E}_{t,\boldsymbol{\epsilon},\tau_0}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\boldsymbol{\tau}_t, t, \mathcal{S}\right)\right\|^2\right],\end{aligned} \tag{5}$$

with $t \sim \mathcal{U}(1, T)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\tau_0 \sim q\left(\tau_0\right)$. Specifically, we utilize the architecture $\boldsymbol{\epsilon}_\theta$ (as shown in Fig. 2) to predict the noise at each diffusion step.

C. Trajectory Optimization via Guided Sampling

$p_\phi\left(\mathcal{O} \mid \boldsymbol{\tau}_{t-1}, \mathcal{S}\right)$ indicates the likelihood of accomplishing the task objective $\mathcal{O}$ in scene $\mathcal{S}$ with trajectory $\boldsymbol{\tau}_{t-1}$. Practically, achieving $\mathcal{O}$ also implies that the trajectory $\boldsymbol{\tau}_{t-1}$ is subject to the constraints imposed by the 3D scene $\mathcal{S}$ and robot embodiment. Therefore, we write $p_\phi\left(\mathcal{O} \mid \boldsymbol{\tau}_{t-1}, \mathcal{S}\right)$ in its exponential and decompose it w.r.t. the task objective and constraints:

$$\begin{aligned}p_\phi\left(\mathcal{O} \mid \boldsymbol{\tau}_{t-1}, \mathcal{S}\right) &\propto \exp\left(\varphi\left(\boldsymbol{\tau}_{t-1}, \mathcal{S}\right)\right) \\ &= \exp\left(-e\left(\boldsymbol{\tau}, \mathcal{S}\right) - \sum_i \lambda_i c_i\left(\boldsymbol{\tau}, \mathcal{S}\right)\right),\end{aligned} \tag{6}$$

where $e\left(\boldsymbol{\tau}, \mathcal{S}\right)$ represents the energy function for task completion, and each $c_i\left(\boldsymbol{\tau}, \mathcal{S}\right)$ represents the cost function of violating a physical constraint. The energy function varies in different tasks, such as object grasping, object placement, or reaching a goal configuration. We define multiple cost functions that penalize failures to meet collision avoidance, trajectory smoothness, and joint limit constraints, balanced with weight λ_i. The design of these functions is detailed in Sec. II-D and Sec. II-E.

According to the definition of $\boldsymbol{\Sigma}_t$ in [92], as the diffusion time step t approaches 0, the noise covariance $\|\boldsymbol{\Sigma}_t\| \to 0$. Consequently, we can approximate $\log p_\phi\left(\mathcal{O} \mid \boldsymbol{\tau}_{t-1}, \mathcal{S}\right)$ using a first-order Taylor expansion around $\boldsymbol{\tau}_{t-1} = \boldsymbol{\mu}_t$ following [61], [62]:

$$\log p_\phi\left(\mathcal{O} \mid \boldsymbol{\tau}_{t-1}, \mathcal{S}\right) \approx \left(\boldsymbol{\tau}_{t-1} - \boldsymbol{\mu}_t\right)^T \boldsymbol{g} + C, \tag{7}$$

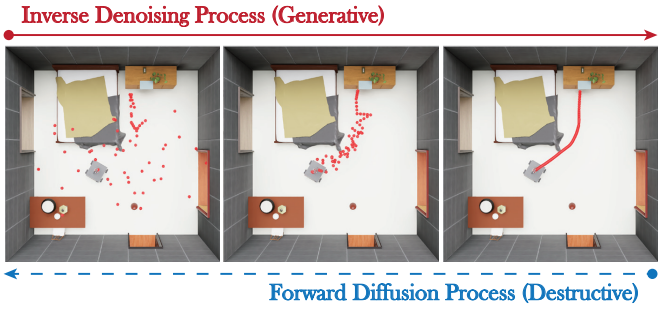


Fig. 3. The diffusion and denoising process of M^{2}Diffuser. The example shows the diffusion and denoising process of the robot's end effector trajectory in a grasping task (*e.g.*, grasping a book).

where $\boldsymbol{\mu}_t = \boldsymbol{\mu}_\theta(\boldsymbol{\tau}_t, t, \mathcal{S})$, $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_\theta(\boldsymbol{\tau}_t, t, \mathcal{S})$, C is a constant, and the gradient

$$\begin{aligned} \boldsymbol{g} &= \nabla_{\boldsymbol{\tau}_{t-1}} \log p_\phi(\mathcal{O} \mid \boldsymbol{\tau}_{t-1}, \mathcal{S})|_{\boldsymbol{\tau}_{t-1}=\boldsymbol{\mu}_t} \\ &= \nabla_{\boldsymbol{\tau}_{t-1}} \varphi(\boldsymbol{\tau}_{t-1}, \mathcal{S})|_{\boldsymbol{\tau}_{t-1}=\boldsymbol{\mu}_t}. \end{aligned} \tag{8}$$

Further, we can rewrite Eq. 2 as:

$$p(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t, \mathcal{S}, \mathcal{O}) = \mathcal{N}(\boldsymbol{\tau}_{t-1}; \boldsymbol{\mu}_t + \boldsymbol{\Sigma}_t \boldsymbol{g}, \boldsymbol{\Sigma}_t), \tag{9}$$

which follows a Gaussian distribution that is easy to sample from. Then trajectory optimization with M^{2}Diffuser is to iteratively apply guided sampling until convergence. We present the complete inference process of M^{2}Diffuser in Alg. 2 and the training process in Alg. 1.

D. Defining Task Objective as Energy Function

As described in Sec. II-C, trajectory optimization with M^{2}Diffuser relies on defining the task objective with an energy function. We consider three alternative task objectives, *i.e.*, object grasping, object placement, and goal reaching. Below, we elaborate the definition of the corresponding energy functions.

a) Grasping Energy: For a 3D object, it's usually nontrivial to define an energy function to measure the quality of a SE(3) grasping pose due to the multi-modal natural of the solution space. Therefore, we define a data-driven grasping energy function following [63] to guide the diffusion process to jointly optimize grasp sampling and trajectory generation:

$$e_{\text{grasp}} = E_\theta\left(P_o, \phi_{ee}^{\mathbf{H}}(\mathbf{q}_{H-1}), t\right), \tag{10}$$

where E_θ is pre-trained as in [63], P_o is the point cloud of target object, $\phi_{ee}^{\mathbf{H}}(\cdot) : \mathbb{R}^d \to \text{SE}(3)$ is the robot's forward kinematics that maps joint position into end effector pose, and $\mathbf{q}_{H-1}$ is the last joint position of sampled trajectories and t is current diffusion time step.

b) Placement Energy: We define a placement energy function to guide the robot to place the object on the target area with physical plausibility:

$$\begin{aligned} e_{\text{place}} = &\sum_{p_{1i} \in P_1} \min_{p_{2j} \in P_2} \left(\|p_{1i} - p_{2j}\|_2^2\right) + \\ &\sum_{p_{2j} \in P_2} \min_{p_{1i} \in P_1} \left(\|p_{2j} - p_{1i}\|_2^2\right), \end{aligned} \tag{11}$$

Algorithm 1: Training of M^{2}Diffuser

Input: Trajectories in 3D scene $(\boldsymbol{\tau}_0, \mathcal{S})$, Noise prediction model of conditional diffusion $\boldsymbol{\epsilon}_\theta$, learning rate η and noise schedule terms $\bar{\alpha}_t$

// train base generation model

repeat

  // sample trajectory

  $\boldsymbol{\tau}_0 \sim p(\boldsymbol{\tau}_0 \mid \mathcal{S})$

  // sample noise and iteration step

  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \sim \mathcal{U}(1, T)$

  // compute loss and update gradient

  $\boldsymbol{\tau}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{\tau}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$

  $\theta = \theta - \eta \nabla_\theta \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}_t, t, \mathcal{S})\|^2$

until *converged*;

Algorithm 2: Inference of M^{2}Diffuser

Modules: Noise prediction model of conditional diffusion $\boldsymbol{\epsilon}_\theta$, initial joint position $\mathbf{q}_0$, energy function $e(\cdot)$, cost functions $\{c_i(\cdot)\}$ with weights $\{\lambda_i\}$

// one-step guided sampling

function sample $(\boldsymbol{\tau}_t, \varphi)$**:**

  // compute the mean and covariance

  $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_t, t, \mathcal{S})$, $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_\theta(\boldsymbol{\tau}_t, t, \mathcal{S})$

  // compute gradient

  $\boldsymbol{g} = \nabla_{\boldsymbol{\tau}_{t-1}} \varphi(\boldsymbol{\tau}_{t-1}, \mathcal{S})|_{\boldsymbol{\tau}_{t-1}=\boldsymbol{\mu}_t}$

  // sample with guidance

  $\boldsymbol{\tau}_{t-1} \sim \mathcal{N}(\boldsymbol{\tau}_{t-1}; \boldsymbol{\mu}_t + \boldsymbol{\Sigma}_t \boldsymbol{g}, \boldsymbol{\Sigma}_t)$

  // set initial state

  $\boldsymbol{\tau}_{t-1}[0] = \mathbf{q}_0$

  return $\boldsymbol{\tau}_{t-1}$

// trajectory optimization

Input: initial trajectory $\boldsymbol{\tau}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\tau_T[0] := \mathbf{q}_0$

// iterative denoising by guided sampling

for $t = T, \cdots, 0$ **do**

  $\boldsymbol{\tau}_{t-1} =$ **sample**$(\boldsymbol{\tau}_t, -\left[e + \sum_i \lambda_i c_i\right])$

// additional steps to improve convergence

for $k = K, \cdots, 1$ **do**

  $\boldsymbol{\tau}_0 =$ **sample**$(\boldsymbol{\tau}_1, -\left[e + \sum_i \lambda_i c_i\right])$

  $\boldsymbol{\tau}_1 = \boldsymbol{\tau}_0$

return $\boldsymbol{\tau}_0$

where P_2 denotes point cloud of the given target area (*e.g.*, desk surface), and P_1 denotes point cloud of the object's placement surface that transforms with $\mathbf{q}_{H-1}$ and the grasping pose. Notably, P_2 can be predicted by O2O-Afford [82] or specified manually, while P_1 can be detected by UOP-Net [83]. In practice, we obtain these point clouds ahead of time in a pre-processing step before we solve for the trajectory.

c) Goal-reaching Energy: The goal-reaching task [14] is to reach an end effector pose represented as the rendered end effector point cloud. We define its energy function to punish the chamfer distance between the goal point cloud and the actual end effector point cloud at configuration $\mathbf{q}_{H-1}$:

$$\begin{aligned} e_{\text{goal}} = &\sum_{p_{ee}^i \in P_{ee}} \min_{p_g^j \in P_g} \left(\|p_{ee}^i - p_g^j\|_2^2\right) + \\ &\sum_{p_g^i \in P_g} \min_{p_{ee}^j \in P_{ee}} \left(\|p_g^i - p_{ee}^j\|_2^2\right), \end{aligned} \tag{12}$$

where P_g denotes the goal point cloud, and P_{ee} denotes the actual point cloud of end effector at $\mathbf{q}_{H-1}$.

E. Defining Physical Constraints as Cost Functions

In the following, we further define the list of cost functions that penalize the violation of physical constraints.

a) Collision-avoidance Cost: We define the collision-avoidance cost function to penalize the physical collision between the scene and the robot. Instead of calculating mesh collision between scene objects and robots, we estimate the collision depth between Signed Distance Field (SDF) of the scene and N sampled points on robot's surface. Then the collision-avoidance cost is defined following [95]:

$$c_{\text{collision}} = \sum_i \sum_h \Phi_s\left(p_h^i\right), \tag{13}$$

where

$$\Phi_s\left(p_h^i\right) = \begin{cases} -\mathcal{D}_s\left(p_h^i\right) + \frac{1}{2}\varepsilon_c & \text{if } \mathcal{D}_s\left(p_h^i\right) < 0, \\ \frac{1}{2\varepsilon_c}\left(\mathcal{D}_s\left(p_h^i\right) - \varepsilon\right)^2 & \text{if } 0 \leqslant \mathcal{D}_s\left(p_h^i\right) \leqslant \varepsilon_c, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

We denote a surface point on the robot's end effector at time step h as p_h^i, and its signed distance in the 3D scene as $\mathcal{D}_s(p_h^i)$. We set a safety margin $\varepsilon_c > 0$ for collision avoidance.

b) Trajectory Smoothness Cost: The smoothness of robot trajectories is essential to prevent abrupt changes in speed and acceleration and improve safety. We define the trajectory smoothness cost function to minimize the difference in joint velocity between adjacent time steps:

$$c_{\text{smoothness}} = \sum_i \sum_h \left\| p_h^{i+2} - 2p_h^{i+1} + p_h^i \right\|_2^2. \tag{15}$$

c) Joint Limit Cost: We define the joint limit cost function to punish the violation of joint limits $\mathbf{q}_{\max}$ and $\mathbf{q}_{\min}$:

$$c_{\text{limit}} = \sum_h \mathcal{J}\left(\mathbf{q}_h\right), \tag{16}$$

where,

$$\mathcal{J}\left(\mathbf{q}_h\right) = \begin{cases} \left\|\mathbf{q}_{\text{lower}} - \mathbf{q}_h\right\|_2^2 & \text{if } \mathbf{q}_h < \mathbf{q}_{\text{lower}}, \\ 0 & \text{if } \mathbf{q}_{\text{lower}} \leq \mathbf{q}_h \leq \mathbf{q}_{\text{upper}}, \\ \left\|\mathbf{q}_{\text{upper}} - \mathbf{q}_h\right\|_2^2 & \text{otherwise.} \end{cases} \tag{17}$$

Here, $\mathbf{q}_{\text{lower}} = \mathbf{q}_{\min} + \varepsilon_l$, $\mathbf{q}_{\text{upper}} = \mathbf{q}_{\max} - \varepsilon_l$, $\mathbf{q}_h$ denotes the joint position at the time step h, and $\varepsilon_l > 0$ defines the safety margin for joint limit violation.

F. Model Architecture

As shown in Fig. 2, M^{2}Diffuser expects three inputs, the current diffusion time step t, intermediate sampled trajectory $\boldsymbol{\tau}_t$ and a scene point cloud cropped from the 3D scan based on the bounding box around the robot. The noise prediction network $\boldsymbol{\epsilon}_\theta$ builds on previous work [62] and adopts a PointTransformer to encode the 3D observation and output latent per-point features as the key and value for the cross-attention module. Moreover, $\boldsymbol{\epsilon}_\theta$ utilizes a fully-connected layer along with positional embedding to extract high-dimensional features from the trajectory. These features are then fused with the diffusion time step embedding through a ResBlock. The fused results are subsequently fed into a self-attention module and served as the query for the cross-attention module. Following that, $\boldsymbol{\epsilon}_\theta$ estimates the noise in the current time step by employing a feedforward layer. Finally, leveraging the estimated noise, M^{2}Diffuser samples the next intermediate trajectory $\boldsymbol{\tau}_{t-1}$ with guidance from the energy and cost functions.

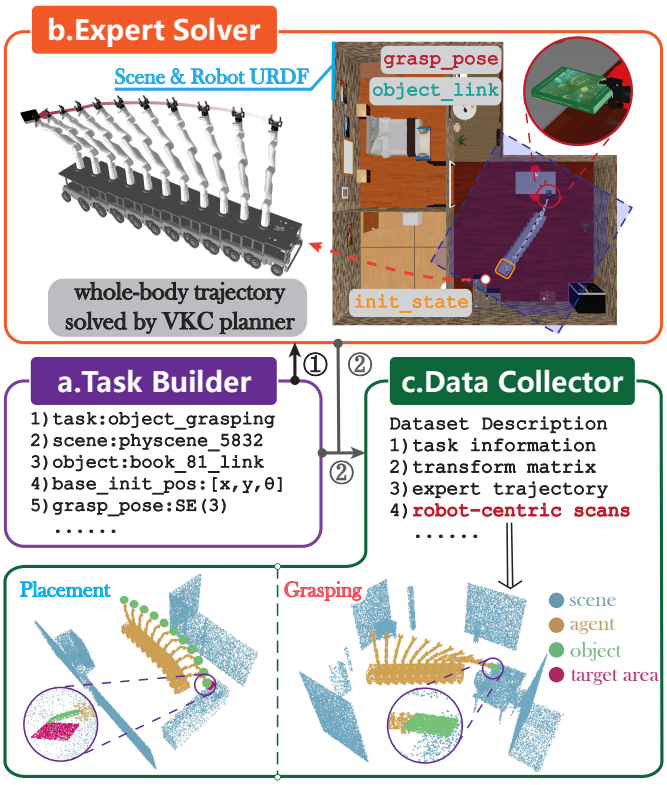


Fig. 4. Dataset collection procedure. (a) The **Task Builder** enables the construction of mobile manipulation tasks through high-level configurations, including scene and robot URDF, manipulated object link, target end effector goal, and task type. (b) The **Expert Solver** computes optimal whole-body coordinated trajectories by leveraging the VKC algorithm [35], [36]. (c) The **Data Collector** is responsible for recording the planned trajectories, and processing the segmented point clouds cropped from the perfect 3D scan based on the bounding box around the robot's initial position.

III. Experiments in Simulated 3D Scenes

A. Dataset Preparation

To collect a large volume of whole-body coordinated expert trajectories, we utilize the autonomous tool developed in our previous work [96]. The data collection procedure is illustrated in Fig. 4. The collected grasping and placement expert trajectories cover 26 common objects with diverse geometries across 24 and 32 simulated 3D scenes, respectively. Specifically, the 24 scenes used for grasping data collection are divided into two groups. From 17 of these scenes, we collected 10673 grasping expert trajectories, which are split into training and testing sets with a 9:1 ratio. The remaining 335 trajectories, collected from the other 7 scenes, are used exclusively for evaluating the model's generalizability to novel scenes. Similarly, the 32 scenes used for placement data collection are divided into two groups. From 24 of these scenes, we collect 8996 placement expert trajectories, which are also divided into training and testing sets following the same ratio. The remaining 351 trajectories from the other 8 scenes are

TABLE I
STATISTICAL ANALYSIS OF WHOLE-BODY TRAJECTORY DATASETS.

Trajectory Dataset	Pan	Book	Fork	Knife	Spoon	Cup	Bowl	Sponge	Bottle	Shaker	Spatula	Ladle	Mug	Egg	Potato	Statue	Plate
Grasp. Set One	771/84	548/71	275/31	341/34	751/80	373/33	229/21	592/78	1015/129	598/77	360/49	33/4	360/91	116/16	119/8	773/79	1324/129
Grasp. Set Two	0/21	0/17	0/13	0/21	0/25	0/34	0/26	0/21	0/26	0/21	0/21	0/21	0/21	0/21	0/26	✗	✗
Place. Set One	407/44	626/77	278/26	615/59	611/68	140/13	187/25	967/111	1084/105	1084/105	159/17	1352/144	364/48	19/0	12/3	16/0	184/25
Place. Set Two	✗	0/48	0/47	✗	0/48	0/48	0/6	✗	0/24	0/1	0/48	0/24	0/8	0/3	✗	✗	✗

The grasping and placement trajectories respectively collected from 24 and 32 simulated scenes are divided into two subsets: set one and set two (generalization evaluation set). The data in the table indicates the number of trajectories used for training (left) and testing (right). We train our model and baselines only on set one and respectively evaluate models performance on both set one (seen scenes) and set two (unseen scenes).

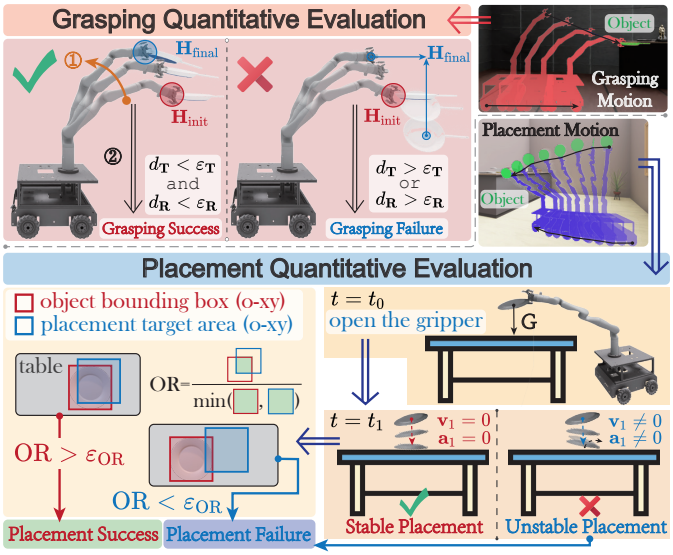


Fig. 5. Quantitative evaluation metrics of grasping and placement tasks. Previous work [3], [19], [91], [98] evaluate object grasping by the contact between the end effector's bounding sphere and the object surface. This evaluation strategy often fails to reflect how the grasping performs in real-world scenarios. In this paper, we evaluate object grasping and placement quality by success rate in simulated scenes with physical simulation enabled. We use NVIDIA Isaac Sim as the physical simulator.

used solely for generalizability evaluation. Tab. I displays some of the manipulated objects along with the number of expert demonstrations used for training and testing. Notably, we reuse the grasping trajectories from our dataset to train and test the target-reaching task.

All our collected expert demonstrations are planned by VKC algorithm [35], [36], an optimization-based global planner specifically designed to solve whole-body trajectory optimization. We use a planning horizon of 50 time steps. The simulated 3D scenes used in our work are sourced from PhyScene [97], a scene synthesis method that generates realistic 3D household scenes with rich interactive objects, tailored for robot learning.

B. Mobile Manipulation Tasks for Evaluation

a) Object Grasping: For a given grasping task, the model input is the segmented and cropped point cloud observed in the robot's initial base frame (see Fig. 4). These points encode three segmentation classes: target object $P_s \in \mathbb{R}^{4096\times3}$, object geometry $P_o \in \mathbb{R}^{512\times3}$ and optional robot geometry at current state $P_r \in \mathbb{R}^{1024\times3}$ (only for baselines). Here, the input of MπFormer is a sequence of these observed point clouds.

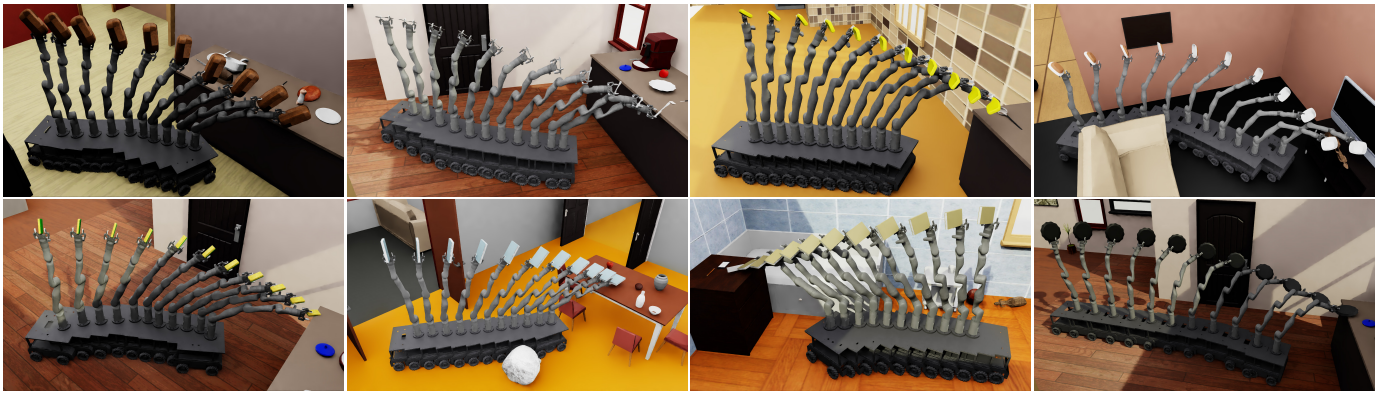
We quantitatively define the *successful* grasping with the assistance of NVIDIA Isaac Sim (as shown in Fig. 5). In the last frame of the generated trajectory, we gradually close the gripper to the smallest opening and note the transformation matrix from the object to the end effector as $\mathbf{H}_{\text{init}} = [\mathbf{R}_{\text{init}} \mid \mathbf{T}_{\text{init}}]$. Then, we lift the arm up a certain height, and the transformation matrix at this point is noted as $\mathbf{H}_{\text{final}} = [\mathbf{R}_{\text{final}} \mid \mathbf{T}_{\text{final}}]$. If the object is gripped successfully, $d_{\mathbf{T}} = \|\mathbf{T}_{\text{init}} - \mathbf{T}_{\text{final}}\|_2 < \varepsilon_{\mathbf{T}}$ and $d_{\mathbf{R}} = \left\|\text{LogMap}\left(\mathbf{R}_{\text{init}}^{\top}\mathbf{R}_{\text{final}}\right)\right\| < \varepsilon_{\mathbf{R}}$, we consider this grasping successful. In the grasping task evaluation, we set $\varepsilon_{\mathbf{T}}$ to 2cm and $\varepsilon_{\mathbf{R}}$ to 15°.

b) Object Placement: The input point clouds in placement tasks consist of four types of points (see Fig. 4): scene points, points on the object's stable placement surface, points in the target placement area $P_p \in \mathbb{R}^{512\times3}$, and optional robot surface points at the current state (only for baselines). The target area is defined as the projection of the object's 3D bounding box onto the horizontal plane when stably placed, with an example shown by the blue box in Fig. 5.

We also quantitatively define the *successful* placement in NVIDIA Isaac Sim. In the last frame of the generated motion, we smoothly open the robot's gripper until it's fully open. Assuming that after 600 simulation steps the object no longer moves, and the overlap ratio (OR) of the bounding box of the object and the target area in the horizontal direction is above $\varepsilon_{\mathbf{OR}}$, we consider this placement successful (see Fig. 5). In the placement evaluation, $\varepsilon_{\mathbf{OR}}$ is set to 0.5.

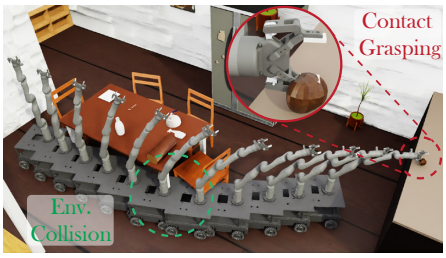
c) Goal-reaching: Given the scene's 3D scan and the surface point cloud of the end effector at the target pose, the motion generator requires to generate a whole-body motion which makes the end effector finally reach the target goal. Input point cloud includes three class points, there are scene points, goal points $P_g \in \mathbb{R}^{512\times3}$ and optional robot surface points at current state (only for baselines). For simplicity, we reuse the grasping trajectories in the dataset for training and testing of goal-reaching task by replacing object points with the surface points of the end effector at the target pose. If the position and orientation target errors of final end effector are below 4cm and 20° respectively, we consider this generated goal-reaching motion a success.

(a) successful grasping trajectories generated by M^{2}Diffuser

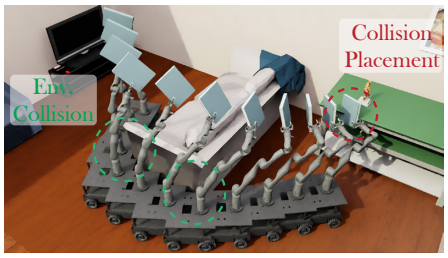


(b) successful placement trajectories generated by M^{2}Diffuser

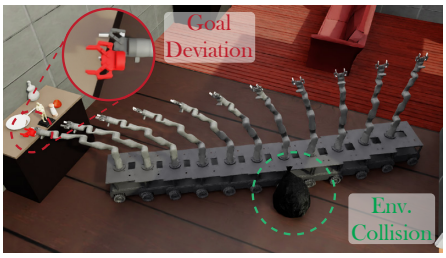
Fig. 6. Successful trajectories generated by M^{2}Diffuser on object grasping and placement tasks. These figures illustrate the successful trajectories generated by our method in (a) grasping and (b) placement tasks involving various objects.



(a) failed object grasping



(b) failed object placement



(c) failed goal-reaching

Fig. 7. Typical failure cases of baseline models. The trajectories generated by baselines fail to (a) grasp the object due to collisions and physical contact, (b) place the object due to an improper pose, and (c) reach the target end effector goal.

C. Experiment Setup

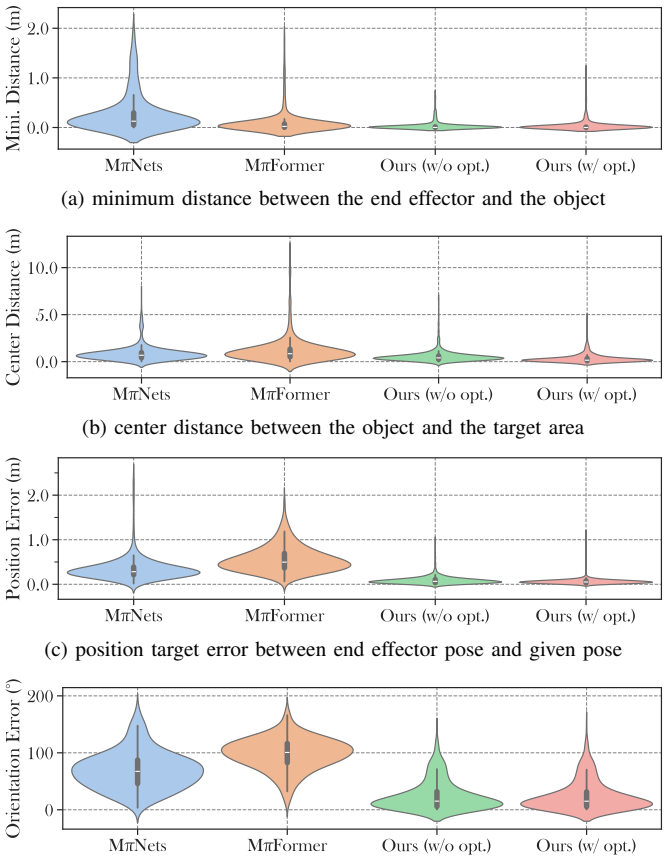
a) Baseline Methods: To the best of our knowledge, M^{2}Diffuser is the first attempt to learn a whole-body neural motion planner for achieving the trajectory generation for mobile manipulation in 3D scenes. Reviewing studies akin to our research, we select MπNets [14] and design MπFormer as compared baselines.

- MπNets [14] is recognized as the state-of-the-art model for addressing collision-free goal-reaching problem, with demonstrated success in 3D-based tabletop manipulation. MπNets is a reactive motion planner, generating the entire trajectories based on autoregressive planning. We extend this model to the mobile manipulation domain to simultaneously predict the base-arm coordinated configuration states. To adapt MπNets for mobile manipulation in 3D environments, we replace the scene-centric observation originally used with the robot-centric observation as the visual input. Since MπNets represents the environment using simple primitive shapes (*e.g.*, cubes, cylinders), we utilize the method from [99] to convert non-watertight meshes into SDF, allowing MπNets to compute the same collision loss for more complex 3D scenes.
- MπFormer is an advanced variant of the Skill Transformer [91] with three key modifications. First, we integrate the action prediction module and the skill prediction module from the original network into a unified whole-body action generation module. This module directly generates coordinated movements of both the base and arms. Second, we enhance the model by incorporating the visual encoder from MπNets to process 3D scans, replacing the original depth encoder. Lastly, MπFormer

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON THREE MOBILE MANIPULATION TASKS.

Test Set	Methods	Success Rate (%)↑	Solving Time (s)↓
Test Set One (seen scenes)	MπNets	0.00	\
	MπFormer	3.93	0.51 ± 0.01
	Ours (w/o opt.)	21.95	$\mathbf{0.47 \pm 0.16}$
	Ours (w/ opt.)	**30.54**	4.74 ± 0.14
	MπNets	2.33	$\mathbf{0.63 \pm 0.05}$
	MπFormer	0.89	0.64 ± 0.07
	Ours (w/o opt.)	4.67	0.77 ± 0.33
	Ours (w/ opt.)	**22.89**	4.93 ± 0.64
	MπNets	3.24	$\mathbf{0.46 \pm 0.08}$
	MπFormer	1.18	0.88 ± 0.01
	Ours (w/o opt.)	25.49	0.55 ± 0.26
	Ours (w/ opt.)	**30.49**	3.89 ± 1.16
Test Set Two (unseen scenes)	MπNets	0.00	\
	MπFormer	3.28	0.73 ± 0.15
	Ours (w/o opt.)	9.25	$\mathbf{0.66 \pm 0.20}$
	Ours (w/ opt.)	**14.33**	6.43 ± 0.29
	MπNets	0.85	$\mathbf{0.59 \pm 0.01}$
	MπFormer	0.28	0.61 ± 0.11
	Ours (w/o opt.)	5.70	0.71 ± 0.25
	Ours (w/ opt.)	**12.25**	4.23 ± 0.17
	MπNets	0.90	$\mathbf{0.49 \pm 0.12}$
	MπFormer	0.00	\
	Ours (w/o opt.)	9.19	0.64 ± 0.21
	Ours (w/ opt.)	**12.61**	4.93 ± 0.25

We qualitatively compare the success rate and solving time of our model with two baseline models across three mobile manipulation tasks (*i.e.*, object grasping, object placement and goal-reaching), followed by a systematic evaluation of the performance of three neural motion planners on both familiar training scenes and previously unseen scenes.



(a) minimum distance between the end effector and the object

(b) center distance between the object and the target area

(c) position target error between end effector pose and given pose

(d) orientation target error between end effector pose and given pose

Fig. 8. Analysis of errors in trajectories generated by different methods. (a) the minimum distance between the final end effector of generated motion and the manipulated object across all grasping tasks. (b) the horizontal distance between the centers of the bounding boxes of the object at final position and the target area across all placement tasks. (c) the position target error between the pose of final end effector and the given pose across all goal-reaching tasks. (d) the orientation target error between the pose of final end effector and the given pose across all goal-reaching tasks.

utilizes the transformer architecture from the Decision Transformer [100] to improve sequence modeling for long-horizon mobile manipulation.

b) Evaluation Metrics: We use some quantitative metrics to evaluate the physical plausibility and task-related completion of generated motion over three diverse tasks.

- *Success Rate*: A trajectory is successful if there are no physical violations, and the position and orientation of final end effector completes the specific task.
- *Time*: The wall time of *successful* trajectory generation for solving the specific task.
- *Collision Rate*: The rate of self and scene collisions.
- *Joint Violation*: The rate of joint values out of the limits.
- *Smoothness*: Same as [14], we compute the Spectral Arc Length (SPARC) [101] values for joint-space trajectory and end effector trajectory. If all these values are below -1.6, we consider the trajectory to be smooth. The smaller the SPARC value, the smoother the trajectory.

c) Training Implementation: We implement M^{2}Diffuser and two baselines in Ubuntu 20.04 with PyTorch, training them on a desktop with an AMD Ryzen 9 7950X 16-Core CPU, two NVIDIA GeForce RTX 4090 GPU, and 128GB of RAM. During the training of M^{2}Diffuser, we use the Adam optimizer with a learning rate 0.0001 to update the model parameters. The maximum diffusion step is set to 50, and we train the model for 2000 epochs with a batch size of 256 per task from the dataset. Specifically, we train MπNets for 100 epochs using a batch size of 256. The other hyperparameters and configurations for MπNets remain unchanged from the original setup [14]. Likewise, we train MπFormer for 100 epochs with a batch size of 64, using the AdamW optimizer with an initial learning rate of 0.0008 and a weight decay of 0.003, where a cosine schedule is employed with 10 epochs warmup. We select the best model in terms of performance on the validation split throughout the training process. More details about model training can be found in our code.

D. Results Analysis

As illustrated in Fig. 6, we visualize the trajectories generated by our method to successfully grasp and place objects. We conduct a comprehensive evaluation of the performance of three models on various testing sets. As shown in Tab. II, M^{2}Diffuser demonstrates the highest success rates across the three mobile manipulation tasks. Specifically, for seen environments during training phase, M^{2}Diffuser achieves the best success rates of 30.54% in object grasping, 22.89% in object placement, and 30.49% in goal-reaching. However, the success rates of MπNets

and MπFormer did not exceed 5% in any task. For tasks in unknown 3D scenes, M^{2}Diffuser also demonstrates stronger generalizability for novel environments than two baselines, with success rates of 14.33%, 12.25%, and 12.61% across three mobile manipulation tasks.

Additionally, we find that the trajectories generated by the baselines generally bring the end effector close to the manipulated object (see Fig. 8a), target placement area (see Fig. 8b) or target end effector goal (see Fig. 8c and see Fig. 8d). However, the inherent shortsightedness of step-by-step autoregressive planning commonly prevents the end effector from successfully converging to an effective grasping pose (see Fig. 7a), precise placement (see Fig. 7b) or reaching the target goal (see Fig. 7c). In contrast, M^{2}Diffuser directly generates and optimizes an entire whole-body motion through an iterative denoising process, thus avoiding these issues, though this comes with an increased solving time.

Furthermore, we observe that frequent jittery movements of the end effector during the transition between adjacent states are the most common failure factor for the baselines. This issue arises mainly due to the frequent distributional shifts and data variance in model training, as well as the inherent shortsightedness of the autoregressive process, which typically results in an ambiguous decision boundary. Autoregressive planning considers only the partially observed sequence of states, without accounting for the global goal of the task. This shortsightedness results in suboptimal planning, especially in long-horizon and high-dimensional mobile manipulation, where the final end effector pose often deviates from the optimal solution, compromising the overall task performance.

The M^{2}Diffuser has three advantages over the previous SOTA neural motion planners in handling complex mobile manipulation tasks with the 3D environments.

a) Motion generation with trajectory optimization: Effective mobile manipulation necessitates robots to interact with their surroundings, while adhering to multiple constraints imposed by both the agent embodiment and the environmental context. Unlike baselines, M^{2}Diffuser avoids the paradigm to design explicit loss functions during model training for learning certain constraints (*e.g.*, collision avoidance [14]). Instead, M^{2}Diffuser integrates physical constraints into the diffusion model and design a guided optimization mechanism in the generation process, leading to an efficient way to introduce implicit task requirements in a differentiable manner for jointly optimizing the task goal sampling and the trajectory generation, ensuring physical plausibility and task-related completeness of the generated motion. This design also facilitates the integration of multiple constraints and fine-tuning of hyperparameters.

As shown in Tab. III, M^{2}Diffuser without tajectory optimization exhibits a higher collision rate with the seen environment compared to MπNets, because collision-avoidance constraint was not considered during the its training phase. However, by introducing the collision-avoidance guided function defined in Eq. 13, the optimized trajectories demonstrate the lowest collision rates on the familiar 3D scenes. Similarly, as indicated in 6th and 7th columns of Tab. III, the SPARC value of the trajectories sampled by M^{2}Diffuser decreases with the guidance of Eq. 15. These smooth trajectories benefits the action execution for the low-level controller. Moreover, our optimization framework also supports data-driven objective functions. As illustrated in Fig. 9, by incorporating implicit grasping and placement energy functions, the task completion of the generated trajectories for grasping and placing objects significantly improves.

TABLE III
PHYSICAL METRICS OF TRAJECTORIES GENERATED BY DIFFERENT METHODS ACROSS THREE MOBILE MANIPULATION TASKS.

Test Set	Methods	Coll. Rate (%)	Avg. Coll. Depth (cm)	Med. Coll. Depth (cm)	Avg. Config SPARC	Avg. End Eff SPARC	Joint Viol. (%)
Test Set One (seen scenes)	MπNets	26.40	5.03	3.65	−4.55	−2.72	5.59
	MπFormer	37.06	4.71	2.94	−4.35	−2.42	1.86
	Ours (w/o opt.)	34.27	3.67	2.34	−2.69	−2.22	0.21
	Ours (w/ opt.)	20.29	3.77	2.50	−3.29	−2.94	0.41
	MπNets	32.44	5.81	4.04	−6.87	−3.16	31.00
	MπFormer	63.33	5.46	3.23	−6.81	−3.20	30.00
	Ours (w/o opt.)	42.44	4.58	2.56	−2.69	−2.40	0.78
	Ours (w/ opt.)	24.78	5.60	3.68	−3.90	−2.63	0.33
	MπNets	21.22	5.61	4.33	−4.56	−2.55	0.69
	MπFormer	71.86	6.78	4.74	−4.67	−3.57	0.00
	Ours (w/o opt.)	27.06	4.09	2.72	−2.70	−2.29	0.20
	Ours (w/ opt.)	20.20	4.24	3.04	−2.80	−2.39	0.29
Test Set Two (unseen scenes)	MπNets	33.73	3.22	1.47	−4.96	−2.87	3.88
	MπFormer	32.54	1.96	1.40	−4.67	−2.54	0.60
	Ours (w/o opt.)	53.13	4.79	2.42	−2.61	−2.01	0.30
	Ours (w/ opt.)	36.42	5.47	2.62	−3.16	−2.62	0.30
	MπNets	37.89	6.60	4.37	−6.30	−3.36	33.33
	MπFormer	66.10	6.08	3.63	−6.65	−3.32	15.67
	Ours (w/o opt.)	57.83	8.75	6.80	−2.51	−2.24	0.85
	Ours (w/ opt.)	27.68	9.22	7.70	−4.06	−3.76	0.28
	MπNets	20.90	7.64	6.46	−4.62	−2.75	0.36
	MπFormer	49.19	7.60	5.45	−4.62	−3.56	0.00
	Ours (w/o opt.)	50.45	6.09	3.65	−2.65	−2.17	0.00
	Ours (w/ opt.)	43.60	6.53	4.29	−2.70	−2.26	0.00

We calculate a series of physical metrics for the trajectories generated by different models. These metrics include the overall collision rate, the average and median collision depths with the environment for all collision-planning cases, the average SPARC values in both the configuration space and end effector space, as well as overall joint violation rate across all mobile manipulation tasks.

Abovementioned optimization framework not only enhances the adaptability of learned planner to complex tasks but also outperforms baseline methods in ensuring safe and successful manipulations in various scenarios.

b) Generalizability and robustness in diverse environments: M^{2}Diffuser utilizes a robot-centric 3D scan for visual observation, which enhances its generalizability across diverse scenarios compared to scene-centric models. By focusing on the local environment around the robot rather than the entire scene, M^{2}Diffuser is more readily extensible to unknown and real-world scenarios. It has demonstrated robust performance in a variety of settings. As evidenced by the results in Tab. II, M^{2}Diffuser achieves success rates of 14.33%, 12.25%, and 12.61% for three evaluation tasks, respectively, in previously unseen scenes. Furthermore, when deployed in real household environments (see Sec. IV), the model trained on simulated data can be directly applied to real 3D environments without any performance gap. Furthermore, our method exhibits strong generalizability at the object level, facilitating the manipulation of a diverse range of geometric shapes and object categories. As demonstrated in Tab. IV, our approach significantly surpasses baseline models in terms of success rates for grasping and placing various objects.

c) Near optimal trajectory generation: M^{2}Diffuser generates trajectories via an iterative denoising process, which infers the entire action sequence rather than only single-step

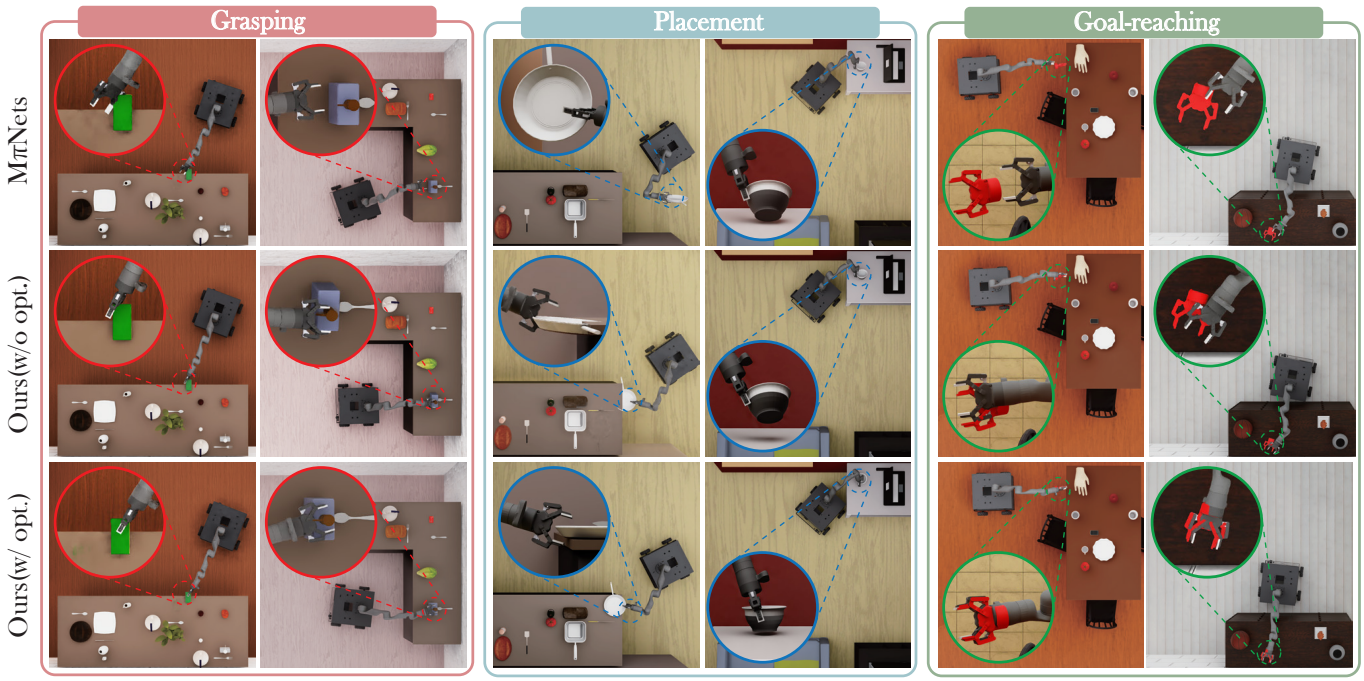


Fig. 9. Final states of generated trajectory: M^{2}Diffuser (Ours) *vs.* MπNets (Baseline). This illustration, rendered in NVIDIA Isaac Sim, displays the final states of trajectories generated by various methods across three mobile manipulation tasks, highlighting the comparative performance of M^{2}Diffuser and MπNets in task completion.

TABLE IV
SUCCESS RATE (%) FOR GRASPING AND PLACING VARIOUS OBJECTS USING DIFFERENT METHODS.

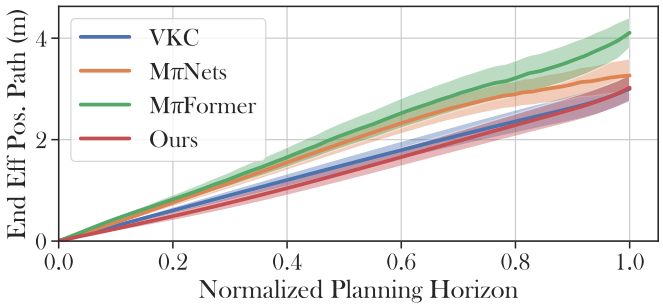
Test Set	Methods	All	Pan	Book	Fork	Knife	Spoon	Cup	Bowl	Sponge	Bottle	Shaker	Spatula	Mug	Egg	Statue	Plate
Test Set One	MπNets	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MπFormer	3.93	3.57	18.31	0.00	8.82	2.50	3.03	4.76	1.28	3.10	1.30	0.00	0.00	0.00	0.00	6.98
	Ours (w/o opt.)	21.95	32.14	**30.99**	0.00	8.82	13.75	15.15	14.29	21.79	**24.81**	28.57	4.65	18.37	**12.50**	26.58	27.13
	Ours (w/ opt.)	**30.54**	**39.29**	28.17	**6.45**	**11.76**	**20.00**	**27.27**	**23.81**	**30.77**	22.48	**37.84**	**18.60**	**28.57**	**12.50**	**55.70**	**45.74**
	MπNets	2.33	0.00	3.90	0.00	1.69	0.00	0.00	4.00	1.80	4.76	0.00	0.00	\	0.00	0.00	6.67
	MπFormer	0.89	6.82	1.30	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00	\	0.00	0.00	3.33
	Ours (w/o opt.)	4.67	9.09	16.88	3.85	1.69	0.00	0.00	8.00	0.00	2.86	0.00	3.47	\	0.00	8.00	8.89
	Ours (w/ opt.)	**22.89**	**36.36**	**64.94**	**3.85**	**3.39**	**4.41**	**15.38**	**32.00**	**28.83**	**3.81**	**0.00**	**11.81**	\	**33.33**	**20.00**	**58.89**
Test Set Two	MπNets	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	\	\
	MπFormer	3.28	4.76	0.00	0.00	0.00	0.00	**11.76**	**19.23**	4.76	0.00	0.00	0.00	0.00	0.00	\	\
	Ours (w/o opt.)	9.25	9.52	11.76	0.00	0.00	**24.00**	0.00	7.69	19.05	**3.85**	23.81	9.52	**23.81**	**4.76**	\	\
	Ours (w/ opt.)	**14.33**	**38.10**	**23.53**	**7.69**	**4.76**	4.00	8.82	3.85	**33.33**	0.00	**38.10**	**33.33**	19.05	**4.76**	\	\
	MπNets	0.85	\	6.25	0.00	\	0.00	0.00	0.00	\	0.00	0.00	0.00	0.00	0.00	\	\
	MπFormer	0.28	\	2.08	0.00	\	0.00	0.00	0.00	\	0.00	0.00	0.00	0.00	0.00	\	\
	Ours (w/o opt.)	5.70	\	25.00	0.00	\	2.08	0.00	16.67	\	0.00	0.00	**12.50**	0.00	0.00	\	\
	Ours (w/ opt.)	**12.25**	\	**62.50**	0.00	\	0.00	**14.58**	**50.00**	\	0.00	0.00	4.17	0.00	0.00	\	\

We report the success rates of M^{2}Diffuser and two baselines for grasping and placing 15 objects from the test set. Test set one includes the results from testing in familiar scenes that were encountered during training, while test set two reflects the results in novel scenes that were not seen during training.

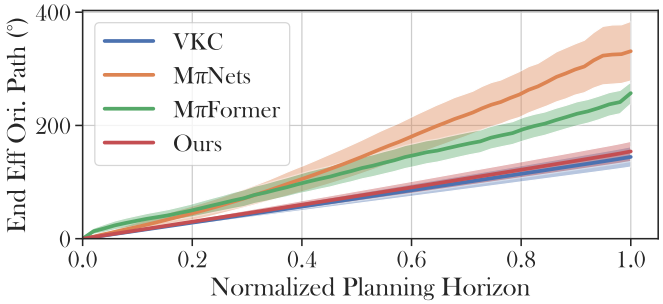
actions. This property inherently promotes global optimality, as it considers the long-term effects of each action throughout the sequence, avoiding the pitfalls of myopic planning. This is crucial for high-dimensional mobile manipulation, where even small errors can be costly (see the 1st row of Fig. 9). Consequently, the trajectories produced by M^{2}Diffuser not only exhibit temporal consistency but also closely align with the globally optimal paths planned by the expert planner, as shown in Fig. 10. These results highlight the superiority of M^{2}Diffuser in learning high-dimensional, whole-body mobile manipulation with a focus on global optimization.

E. Ablation Experiments

To confirm the significance of the learned diffusion priors for trajectory sampling and optimization, we present both physical and task-related performance comparison for trajectories sampled with and without diffusion priors. As demonstrated in Tab. V, our method considerably outperforms the direct use



(a) end effector position paths generated via different planners



(b) end effector orientation paths generated via different planners

Fig. 10. Performance comparison of different models in learning globally optimal expert planner. We summarize the planning results of goal-reaching tasks, where MπNets, MπFormer and the expert planner (VKC) all successfully perform. Then, we plot the continuous path curves for position and orientation of the end effector as the planning progresses. In the path curves, each point represents the average path length of position or orientation traversed by the end effector at the current normalized planning step, while the shaded area indicates the variance of the path length. Obviously, the planning results of M^{2}Diffuser closely align with those of the globally optimal expert planner.

of inverse Langevin diffusion in terms of convergence speed and success rate. By learning a prior trajectory-level generator, M^{2}Diffuser efficiently guides the optimization algorithm to rapidly search robust and high-quality solutions within a reduced search space. In our experiments, the sampling process of the inverse Langevin diffusion is defined by referencing to [63], where

$$\boldsymbol{\tau}_{k-1} = \boldsymbol{\tau}_k + 0.5\alpha_k^2 \nabla_{\boldsymbol{\tau}_k} \varphi(\boldsymbol{\tau}_k) + \alpha_k \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (18)$$

with pre-defined step dependent coefficient α_k and objective function $\varphi(\cdot)$ previously defined in Eq. 6.

TABLE V
QUANTITATIVE COMPARISON OF M^{2}DIFFUSER AND INVERSE LANGEVIN DIFFUSION IN SOLVING GOAL-REACHING TASK.

Diffusion Priors	Iterative Step	Succ. Rate (%)	Coll. Depth (m)
✗	50	0.00	7.15
✗	500	0.00	6.94
✗	1000	0.00	6.79
✗	2000	0.00	6.65
✓	**50**	**51.00**	**4.79**

We randomly select 100 goal-reaching tasks to test the performance of trajectory optimization with and without the learned diffusion priors.

IV. EXPERIMENTS IN REAL-WORLD 3D SCENES

This section illustrates the application of our method to a real mobile manipulator performing objects rearrangement and handover tasks in a real household environment. The real-world 3D environment and our robot system are depicted in Fig. 11. To the best of our knowledge, this is the first study to directly apply an IL-based neural motion planner trained on simulated data to real-world mobile manipulation tasks. We confirm that our method can seamlessly transfer from simulation to the real world. The following subsections introduce robot system settings (in Sec. IV-A), real-world experiment setup (in Sec. IV-B), and real-world experiment results (in Sec. IV-C).

A. Robot System Settings

The robot used in our real-world experiments is a 10-DoF mobile manipulator (see Fig. 11b), which consists of a 3-DoF Dingo base with omnidirectional Mecanum wheels, a 7-DoF Kinova Gen3 arm, and a Robotiq-2F-85 gripper. It shares the same geometric embodiment as the robot model used in simulations and has the capacity to perform intricate mobile manipulation in man-made environments.

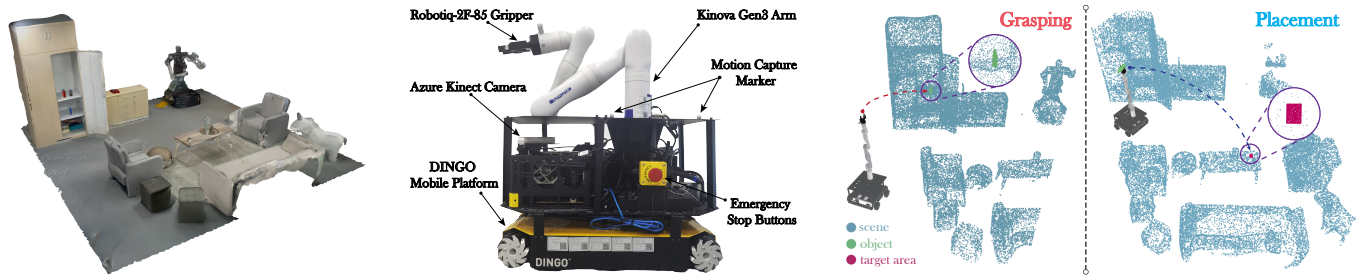
B. Experiment Setup

a) Mobile Manipulation Tasks: We set up a series of object rearrangement tasks involving three common geometric shapes: planar, cuboids, and cylindrical objects. In these tasks, the robot is required to pick objects from their initial locations and place them to specified target areas (*e.g.*, on a table surface or around a person) based on the segmented scene's natural 3D scan and object segmented masks. To successfully complete the task, the robot must not only execute each grasping and placement accurately but also avoid collisions or other physical violations. A similar object rearrangement task was explored in [4] and has been shown to be highly challenging in real-world environments. The experimental site is set up in a real living room environment with various objects and obstacles. The model only trained with simulated data will be directly used in the real-world setting without any fine-tuning.

b) Experiment Preparation: As shown in Fig. 11a, we scan and reconstruct the household environment. The reconstructed scene's point cloud is then segmented and cropped to serve as the visual input (see Fig. 11c) for M^{2}Diffuser. Additionally, we adopt the algorithm proposed by [99] to calculate the SDF of the reconstructed scene for collision-avoidance cost computation during trajectory optimization. In real-world experiments, the overhead VICON system provides real-time localization of the robot.

C. Experiment Results

As shown in Fig. 12, we conduct a series of pick-and-place tasks involving various common objects, including a bottle, a chip bag, a book and a tea box. For each task, the M^{2}Diffuser first generates a trajectory for grasping the object and then plans a subsequent trajectory for placing it in a target area on the table or around a person. In real-world experiments, we attempt



(a) real-world 3D scan (b) mobile manipulator (c) robot-centic 3D scans for grasping and placement

Fig. 11. Real-world 3D environment and robot system. (a) The real-world 3D environment is scanned and reconstructed by using PolyCam software. (b) The mobile manipulation system comprises a 3-DoF mobile base, a 7-DoF manipulation arm, and additional attachments. (c) The robot-centric 3D scans are utilized for grasping and placements tasks in experiment one of Fig. 12.

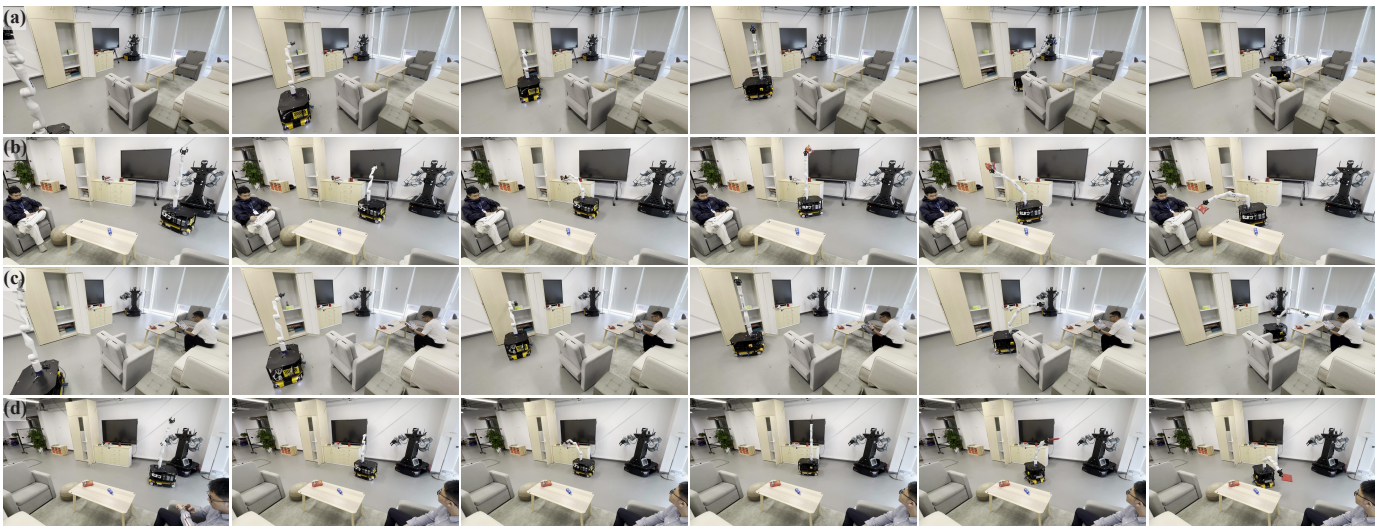


Fig. 12. Real-world experiments. From top to bottom, the figures depict the mobile manipulator (a) taking a bottle from the cabinet and placing it on the table, (b) handing a chip bag to a seated person, (c) retrieving a tea box from the cabinet and setting it on the table, and (d) delivering a book to a seated person.

to directly apply the model trained on simulated data to pick-and-place tasks in real-world 3D scenes and unseen objects, achieving significant success. By leveraging robot-centric 3D scans as visual input, our model first achieves seamless sim-to-real transfer in learning-based mobile manipulation. However, previous works either failed to directly apply models trained in simulation to real-world scenarios or were limited to highly structured environments. Additionally, we also demonstrate the generalizability and robustness of the trajectory optimization framework in handling previously unseen environments and objects.

V. Limitations and Future Work

The primary limitations of the M^{2}Diffuser include its slow training and inference speed and strong dependence on the objective function designs. More details are as follows:

a) Slow Training and Inference: The training and inference of M^{2}Diffuser are both slow due to the numerous iterative steps required for generating outputs, a common issue with diffusion models. As M^{2}Diffuser optimizes the sampled trajectories at each iterative denoising step, it is incompatible with sampling acceleration algorithms such as DDIM [102], which often compromise optimization performance. As shown in Tab. II, the introduction of optimized guidance terms results in a 5 to 10-fold decrease in inference speed for M^{2}Diffuser.

b) Strong Dependence on Objective Designs: M^{2}Diffuser optimizes the sampled trajectories, depending heavily on the design of the energy and cost functions, as well as meticulous hyper-parameter tuning. These functions can be either explicitly defined through heuristic designs or implicitly derived from data-driven models. However, for the multi-stage tasks in [59] and the human-like skills in [103], task-related optimization often proves impractical due to the challenges on designing smooth objective functions.

In future work, we will attempt to solve the aforementioned limitations by exploring the latest advancements in diffusion model acceleration and loss guidance algorithms to reduce the number of inference steps required without sacrificing optimized performance, such as new noise schedules [104] and LGD-MC [105].

VI. Conclusion

We proposed M^{2}Diffuser, the first scene-conditioned motion generator tailored for mobile manipulation in EAI. M^{2}Diffuser seamlessly integrates multiple physical constraints and task objective, and employs generative modeling techniques to directly generate highly coordinated whole-body motion trajectories

with physical plausibility and task completion from natural 3D scans. We demonstrate that the M^{2}Diffuser outperforms previous SOTA neural motion planners by a large margin on various tasks, establishing its efficacy and flexibility. Furthermore, we also demonstrated that the diffusion-based planning paradigm, along with using robot-centric 3D scans as visual observation, can be more effective for the real-world generalization and deployment of mobile manipulation.

REFERENCES

[1] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7272–7281. 1

[2] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3357–3364. 1

[3] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi *et al.*, "Rearrangement: A challenge for embodied ai," *arXiv preprint arXiv:2011.01975*, 2020. 1, 7

[4] J. Gu, D. S. Chaplot, H. Su, and J. Malik, "Multi-skill mobile manipulation for object rearrangement," *arXiv preprint arXiv:2209.02778*, 2022. 1, 2, 3, 12

[5] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023. 1

[6] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans, "Foundation models for decision making: Problems, methods, and opportunities," *arXiv preprint arXiv:2303.04129*, 2023. 1

[7] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024. 1

[8] M. Han, Y. Zhu, S.-C. Zhu, Y. N. Wu, and Y. Zhu, "Interpret: Interactive predicate learning from language feedback for generalizable task planning," in *Robotics: Science and Systems (RSS)*, 2024. 1

[9] J. Shang, K. Schmeckpeper, B. B. May, M. V. Minniti, T. Kelestemur, D. Watkins, and L. Herlant, "Theia: Distilling diverse vision foundation models for robot learning," *arXiv preprint arXiv:2407.20179*, 2024. 1

[10] O. Khatib, "Mobile manipulation: The robotic assistant," *Robotics and Autonomous Systems*, vol. 26, no. 2-3, pp. 175–183, 1999. 1

[11] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 31–36. 2

[12] C. Li, F. Xia, R. Martín-Martín, and S. Savarese, "HRL4IN: hierarchical reinforcement learning for interactive navigation with mobile manipulators," in *Conference on Robot Learning (CoRL)*, vol. 100, 2019, pp. 603–616. 2

[13] S. Feng, B. Sebastian, and P. Ben-Tzvi, "A collision avoidance method based on deep reinforcement learning," *Robotics*, vol. 10, no. 2, p. 73, 2021. 2

[14] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox, "Motion policy networks," in *Conference on Robot Learning (CoRL)*, vol. 205, 2022, pp. 967–977. 2, 3, 5, 8, 9, 10

[15] Y. Lee, S. Sun, S. Somasundaram, E. S. Hu, and J. J. Lim, "Composing complex skills by learning transition policies," in *International Conference on Learning Representations (ICLR)*, 2019. 2

[16] Y. Lee, J. Yang, and J. J. Lim, "Learning to coordinate manipulation skills via skill behavior diversification," in *International Conference on Learning Representations (ICLR)*, 2020. 2

[17] Y. Lee, J. J. Lim, A. Anandkumar, and Y. Zhu, "Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization," in *Conference on Robot Learning (CoRL)*, vol. 164, 2021, pp. 406–416. 2

[18] B. Wu, R. Martín-Martín, and L. Fei-Fei, "M-EMBER: tackling long-horizon mobile manipulation via factorized domain transfer," in *International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 690–11 697. 2

[19] T. Ni, K. Ehsani, L. Weihs, and J. Salvador, "Towards disturbance-free visual mobile manipulation," in *Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 5208–5220. 2, 3, 7

[20] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. A. Funkhouser, "Tidybot: personalized robot assistance with large language models," *Autonomous Robots*, vol. 47, no. 8, pp. 1087–1102, 2023. 2, 3

[21] H. Xiong, R. Mendonca, K. Shaw, and D. Pathak, "Adaptive mobile manipulation for articulated objects in the open world," *arXiv preprint arXiv:2401.14403*, 2024. 2, 3

[22] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation," in *International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4583–4590. 2

[23] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter, "Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 2377–2384, 2022. 2

[24] C. Sun, J. Orbik, C. M. Devin, B. H. Yang, A. Gupta, G. Berseth, and S. Levine, "Fully autonomous real-world reinforcement learning with applications to mobile manipulation," in *Conference on Robot Learning (CoRL)*, vol. 164, 2021, pp. 308–319. 2

[25] N. Yokoyama, A. Clegg, J. Truong, E. Undersander, T. Yang, S. Arnaud, S. Ha, D. Batra, and A. Rai, "ASC: adaptive skill coordination for robotic mobile manipulation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 9, no. 1, pp. 779–786, 2024. 2

[26] J. Hu, P. Stone, and R. Martín-Martín, "Causal policy gradient for whole-body mobile manipulation," *arXiv preprint arXiv:2305.04866*, 2023. 2

[27] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning (CoRL)*, vol. 205, 2022, pp. 138–149. 2

[28] M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Yang, and X. Wang, "Visual whole-body control for legged loco-manipulation," *arXiv preprint arXiv:2403.16967*, 2024. 2

[29] D. Honerkamp, T. Welschehold, and A. Valada, "Learning kinematic feasibility for mobile manipulation through deep reinforcement learning," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 6289–6296, 2021. 2

[30] D. Honerkamp and T. Welschehold, "N2m2: Learning navigation for arbitrary mobile manipulation motions in unseen and dynamic environments," *Transactions on Robotics (T-RO)*, vol. 39, no. 5, pp. 3601–3619, 2023. 2

[31] L. Naik, S. Kalkan, and N. Krüger, "Pre-grasp approaching on mobile robots: A pre-active layered approach," *IEEE Robotics and Automation Letters (RA-L)*, vol. 9, no. 3, pp. 2606–2613, 2024. 2

[32] M. V. Minniti, F. Farshidian, R. Grandia, and M. Hutter, "Whole-body MPC for a dynamically stable mobile manipulator," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 4, pp. 3687–3694, 2019. 2

[33] M. Stuede, K. Nuelle, S. Tappe, and T. Ortmaier, "Door opening and traversal with an industrial cartesian impedance controlled mobile robot," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 966–972. 2

[34] J. Chiu, J. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, "A collision-free MPC for whole-body dynamic locomotion and manipulation," in *International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4686–4693. 2

[35] Z. Jiao, Z. Zhang, X. Jiang, D. Han, S. Zhu, Y. Zhu, and H. Liu, "Consolidating kinematic models to promote coordinated mobile manipulations," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 979–985. 2, 6, 7

[36] Z. Jiao, Z. Zhang, W. Wang, D. Han, S. Zhu, Y. Zhu, and H. Liu, "Efficient task planning for mobile manipulation: a virtual kinematic chain perspective," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8288–8294. 2, 6, 7

[37] Z. Li, Y. Niu, Y. Su, H. Liu, and Z. Jiao, "Dynamic planning for sequential whole-body mobile manipulation," *arXiv preprint arXiv:2405.15377*, 2024. 2

[38] M. Han, Z. Zhang, Z. Jiao, X. Xie, Y. Zhu, S. Zhu, and H. Liu, "Scene reconstruction with functional objects for robot autonomy," *International Journal of Computer Vision (IJCV)*, vol. 130, no. 12, pp. 2940–2961, 2022. 2

[39] Z. Zhang, L. Zhang, Z. Wang, Z. Jiao, M. Han, Y. Zhu, S. Zhu, and H. Liu, "Part-level scene reconstruction affords robot interaction," in *International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 11 178–11 185. 2

[40] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, no. 1, pp. 265–293, 2021. 2

[41] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science robotics*, vol. 8, no. 84, 2023. 2

[42] C. Liu, H. Wu, Y. Zhong, X. Zhang, Y. Wang, and W. Xie, "Intelligent grimm - open-ended visual storytelling via latent diffusion models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 6190–6200. 2

[43] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 674–10 685. 2

[44] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," pp. 3813–3824, 2023. 2

[45] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis, "Align your latents: High-resolution video synthesis with latent diffusion models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 22 563–22 575. 2

[46] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts *et al.*, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023. 2

[47] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy," *arXiv preprint arXiv:2403.03954*, 2024. 2

[48] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv preprint arXiv:2402.10885*, 2024. 2

[49] M. Dalal, J. Yang, R. Mendonca, Y. Khaky, R. Salakhutdinov, and D. Pathak, "Neural mp: A generalist neural motion planner," *arXiv preprint arXiv:2409.05864*, 2024. 2

[50] J. J. Johnson, L. Li, F. Liu, A. H. Qureshi, and M. C. Yip, "Dynamically constrained motion planning networks for non-holonomic robots," in *International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6937–6943. 2

[51] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning (CoRL)*, vol. 205, 2022, pp. 785–799. 2

[52] Y. Ze, G. Yan, Y. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang, "Gnfactor: Multi-task real robot learning with generalizable neural feature fields," in *Conference on Robot Learning (CoRL)*, vol. 229, 2023, pp. 284–301. 2

[53] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023. 2, 3

[54] G. Yan, Y.-H. Wu, and X. Wang, "Dnact: Diffusion guided multi-task 3d policy learning," *arXiv preprint arXiv:2403.04115*, 2024. 2

[55] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y. Chao, and D. Fox, "RVT: robotic view transformer for 3d object manipulation," in *Conference on Robot Learning (CoRL)*, vol. 229, 2023, pp. 694–710. 2

[56] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision making?" in *International Conference on Learning Representations (ICLR)*, 2023. 2

[57] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning (ICML)*, vol. 162, 2022, pp. 9902–9915. 2

[58] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton, "Structdiffusion: Language-guided creation of physically-valid structures using unseen objects," in *Robotics: Science and Systems (RSS)*, 2023. 2

[59] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Robotics: Science and Systems (RSS)*, 2023. 2, 3, 13

[60] I. Kapelyukh, V. Vosylius, and E. Johns, "Dall-e-bot: Introducing web-scale diffusion models to robotics," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 7, pp. 3956–3963, 2023. 2

[61] 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 *International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1916–1923. 2, 4

[62] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S. Zhu, "Diffusion-based generation, optimization, and planning in 3d scenes," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 16 750–16 761. 2, 3, 4, 6

[63] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," in *International Conference on Robotics and Automation (ICRA)*, 2023. 2, 3, 5, 12

[64] U. A. Mishra and Y. Chen, "Reorientdiff: Diffusion model based reorientation for object manipulation," *arXiv preprint arXiv:2303.12700*, 2023. 2

[65] L. Yang, Z. Huang, F. Lei, Y. Zhong, Y. Yang, C. Fang, S. Wen, B. Zhou, and Z. Lin, "Policy representation via diffusion probability model for reinforcement learning," *arXiv preprint arXiv:2305.13122*, 2023. 2

[66] X. Ma, S. Patidar, I. Haughton, and S. James, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 081–18 090. 2

[67] W. K. Kim, M. Yoo, and H. Woo, "Robust policy learning via offline skill diffusion," *arXiv preprint arXiv:2403.00225*, 2024. 2

[68] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *International Conference on Robotics and Automation (ICRA)*, 2024, pp. 63–70. 2

[69] D. Song, J. Liang, X. Xiao, and D. Manocha, "Tgs: Trajectory generation and selection using vision language models in mapless outdoor environments," *arXiv preprint arXiv:2408.02454*, 2024. 2

[70] W. Yu, J. Peng, H. Yang, J. Zhang, Y. Duan, J. Ji, and Y. Zhang, "Ldp: A local diffusion planner for efficient robot navigation and collision avoidance," *arXiv preprint arXiv:2407.01950*, 2024. 2

[71] M. Stamatopoulou, J. Liu, and D. Kanoulas, "Dippest: Diffusion-based path planner for synthesizing trajectories applied on quadruped robots," *arXiv preprint arXiv:2405.19232*, 2024. 2

[72] A. Das, R. D. Yadav, S. Sun, M. Sun, S. Kaski, and W. Pan, "Dronediffusion: Robust quadrotor dynamics learning with diffusion models," *arXiv preprint arXiv:2409.11292*, 2024. 2

[73] Z. Zhang, L. Zhou, C. Liu, Z. Liu, C. Yuan, S. Guo, R. Zhao, M. H. Ang Jr, and F. E. Tay, "Dexgrasp-diffusion: Diffusion-based unified functional grasp synthesis pipeline for multi-dexterous robotic hands," *arXiv preprint arXiv:2407.09899*, 2024. 2

[74] Z. Weng, H. Lu, D. Kragic, and J. Lundell, "Dexdiffuser: Generating dexterous grasps with diffusion models," *arXiv preprint arXiv:2402.02989*, 2024. 2

[75] J. Yamada, S. Zhong, J. Collins, and I. Posner, "D-cubed: Latent diffusion trajectory optimisation for dexterous deformable manipulation," *arXiv preprint arXiv:2403.12861*, 2024. 2

[76] T. J. Wang, J. Zheng, P. Ma, Y. Du, B. Kim, A. Spielberg, J. B. Tenenbaum, C. Gan, and D. Rus, "Diffusebot: Breeding soft robots with physics-augmented generative diffusion models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[77] X. Xu, H. Ha, and S. Song, "Dynamics-guided diffusion model for robot manipulator design," *arXiv preprint arXiv:2402.15038*, 2024. 2

[78] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998. 3

[79] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017. 3

[80] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 438–13 444. 3

[81] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 2901–2910. 3

[82] K. Mo, Y. Qin, F. Xiang, H. Su, and L. J. Guibas, "O2o-afford: Annotation-free large-scale object-object affordance learning," in *Conference on Robot Learning (CoRL)*, vol. 164, 2021, pp. 1666–1677. 3, 5

[83] S. Noh, R. Kang, T. Kim, S. Back, S. Bak, and K. Lee, "Learning to place unseen objects stably using a large-scale simulation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 9, no. 3, pp. 3005–3012, 2024. 3, 5

[84] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, "Planning multi-fingered grasps as probabilistic inference in a learned deep network," in *Robotics Research*, 2020, pp. 455–472. 3

[85] M. Danielczuk, A. Mousavian, C. Eppner, and D. Fox, "Object rearrangement using learned implicit collision functions," in *International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6010–6017. 3

[86] J. Urain, A. T. Le, A. Lambert, G. Chalvatzaki, B. Boots, and J. Peters, "Learning implicit priors for motion optimization," in *International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 7672–7679. 3

[87] M. Koptev, N. Figueroa, and A. Billard, "Neural joint space implicit signed distance functions for reactive robot manipulator control," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 480–487, 2023. 3

[88] Z. Zhu, H. Zhao, H. He, Y. Zhong, S. Zhang, Y. Yu, and W. Zhang, "Diffusion models for reinforcement learning: A survey," *arXiv preprint arXiv:2311.01223*, 2023. 3

[89] J. Wong, A. Tung, A. Kurenkov, A. Mandlekar, L. Fei-Fei, S. Savarese, and R. Martín-Martín, "Error-aware imitation learning from teleoperation data for mobile manipulation," in *Conference on Robot Learning (CoRL)*, vol. 164, 2021, pp. 1367–1378. 3

[90] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 1647–1654. 3

[91] X. Huang, D. Batra, A. Rai, and A. Szot, "Skill transformer: A monolithic policy for mobile manipulation," in *International Conference on Computer Vision (ICCV)*, 2023, pp. 10 818–10 828. 3, 7, 8

[92] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 4

[93] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations (ICLR)*, 2021. 4

[94] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 11 895–11 907. 4

[95] N. D. Ratliff, M. Zucker, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: gradient optimization techniques for efficient motion planning," in *International Conference on Robotics and Automation (ICRA)*, 2009, pp. 489–494. 6

[96] Z. Zhang, S. Yan, M. Han, Z. Wang, X. Wang, S.-C. Zhu, and H. Liu, "M^3bench: Benchmarking whole-body motion generation for mobile manipulation in 3d scenes," *arXiv preprint arXiv:2410.06678*, 2024. 6

[97] Y. Yang, B. Jia, P. Zhi, and S. Huang, "Physcene: Physically interactable 3d scene synthesis for embodied AI," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16 262–16 272. 7

[98] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi, "Manipulathor: A framework for visual object manipulation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4497–4506. 7

[99] P. Wang, Y. Liu, and X. Tong, "Dual octree graph networks for learning adaptive volumetric shape representations," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 103:1–103:15, 2022. 8, 12

[100] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 15 084–15 097. 9

[101] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami, and E. Burdet, "On the analysis of movement smoothness," *Journal of neuroengineering and rehabilitation*, vol. 12, pp. 1–11, 2015. 9

[102] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations (ICLR)*, 2021. 13

[103] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024. 13

[104] T. Chen, "On the importance of noise scheduling for diffusion models," *arXiv preprint arXiv:2301.10972*, 2023. 13

[105] J. Song, Q. Zhang, H. Yin, M. Mardani, M. Liu, J. Kautz, Y. Chen, and A. Vahdat, "Loss-guided diffusion models for plug-and-play controllable generation," in *International Conference on Machine Learning (ICML)*, vol. 202, 2023, pp. 32 483–32 498. 13

Sixu Yan received the M.S. degree from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2024, and the B.E. degree from Ocean University of China (OUC), Qingdao, China, in 2021, both in Mechanical Engineering. He is currently a first-year Ph.D. student at the School of Electronic Information and Communications, Huazhong University of Science and Technology (HUST). His research interests include robotics and computer vision.

Zeyu Zhang (Member, IEEE) received his Ph.D. degree in Computer Science from the University of California, Los Angeles (UCLA) in 2023. He is currently a research scientist at State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). He received an M.S. degree in Computer Science from UCLA in 2019 and B.S. degree in Computer Science from Hunan University in 2017. His research interests focus on robot perception, learning, and cognitive robotics

Muzhi Han received a B.E. in Mechanical Engineering from Tsinghua University, Beijing, China, in 2019. He is currently a final-year Ph.D. candidate at the University of California, Los Angeles (UCLA), advised by Prof. Song-Chun Zhu. His research interests include robotics and machine perception.

Zaijin Wang received the M.S. degree in Mechanical Engineering from North China University of Technology in 2013, and the B.E. degree in Mechanical Engineering from Qingdao University in 2010. He is currently a research engineer at the State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). His research interests include computer vision, machine learning, motion control, and cognitive robotics.

Qi Xie received the M.S. degree in Applied Data Science from University of Southern California in 2022. She is now a research engineer at State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). Her research interests include embodied AI and cognitive science.

Zhitian Li received the M.S. degree in Mechanical Engineering from Beijing Institute of Technology (BIT), Beijing, China, in 2023, and the B.E. degree from BIT in 2020. He is currently a first-year Ph.D. student at the School of Automation Science and Electrical Engineering, Beihang University (BUAA). His research interests include robot planning and control.

Zhehan Li received the B.E. degree from Xidian University, Xi'an, China, in 2022. He is currently a Ph.D. student at the School of Artificial Intelligence, Xidian University. His research interests include robotics and unmanned systems.

Hangxin Liu (Member, IEEE) received his Ph.D. degree in Computer Science from the University of California, Los Angeles (UCLA) in 2021. He is currently the leader of the robotics lab and a research scientist at State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). He received an M.S. degree in Mechanical Engineering from UCLA in 2018 and two B.S. degrees in Mechanical Engineering and Computer Science, both from Virginia Tech in 2016. His research interests focus on robot perception, learning, human-robot interaction, and cognitive robotics.

Xinggang Wang (Senior Member, IEEE) received the B.S. and Ph.D. degrees in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2009 and 2014, respectively. He is currently a Professor at the School of Electronic Information and Communications, HUST. He serves as Co-Editor-in-Chief of Image and Vision Computing and area chair of CVPR and ICCV. His research interests include computer vision and deep learning.

Song-Chun Zhu (Fellow, IEEE) received a Ph.D. degree from Harvard University in 1996, and is a Chair Professor jointly at Tsinghua University and Peking University, Dean of Institute for Artificial Intelligence at Peking University. He worked at Brown, Stanford, Ohio State, and UCLA before returning to China in 2020 to launch a non-profit organization—Beijing Institute for General Artificial Intelligence (BIGAI). He has published over 300 papers in computer vision, statistical modeling and learning, cognition, language, robotics, and AI. He received the Marr Prize in 2003, the Aggarwal prize from the Intl Association of Pattern Recognition in 2008, the Helmholtz Test-of-Time prize in 2013, twice Marr Prize honorary nominations in 1999 and 2007, the Sloan Fellowship, the US NSF Career Award, and the ONR Young Investigator Award in 2001. He served as General co-Chair for CVPR 2012 and CVPR 2019.