GeoDiff: Geometry-Conditioned Diffusion Policy for Refined Robotic Trajectory Generation

Weikang Xie
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou, China
760301162@qq.com

Hankz Hankui Zhuo
National Key Lab. for Novel Software Tech.
School of AI, Nanjing University
Nanjing, China
hankz@nju.edu.cn

Rong Pan
School of Artificial Intelligence
Sun Yat-sen University
Guangzhou, China
panr@sysu.edu.cn

***Abstract*—Learning physically consistent and robust manipulation policies from perceptual inputs remains a key challenge in robot learning. Most existing diffusion-based approaches condition only on raw point clouds or robot states, failing to exploit the underlying geometric relations that govern feasible object interactions. To address this gap, we propose GeoDiff, a *geometry-conditioned diffusion policy* for refined robotic trajectory generation. GeoDiff constructs object-centric geometric representations via clustering-based point cloud segmentation and encodes relational features capturing spatial dependencies between the robot and surrounding objects. Conditioned on these geometric features, the diffusion policy generates multiple stochastic trajectory candidates under consistent initial conditions. A physics-aware evaluation module then scores each candidate based on smoothness, goal accuracy, and collision safety, selecting the optimal physically valid trajectory. We leverage a composite loss combining denoising reconstruction and differentiable physical consistency to further enforces smooth, goal-directed, and collision-free motion generation. Extensive experiments across three well-known simulated manipulation benchmarks demonstrate that GeoDiff achieves over 15% improvement in task success rate and motion smoothness compared with state-of-the-art diffusion and optimization-based baselines. Those results highlight the importance of geometric conditioning and physics-guided refinement for reliable diffusion-based robotic manipulation.**

***Index Terms*—Robotic manipulation, diffusion policy, geometric conditioning, trajectory refinement, physical consistency.**

I. Introduction

Learning robust visuomotor policies that produce physically consistent motion in complex 3D environments remains a core challenge in robot manipulation. While end-to-end visuomotor methods [1] have demonstrated that mapping raw visual observations to motor commands is feasible, they frequently underperform when generalizing to diverse object geometries and multi-contact interactions. Recent work on diffusion models offers a compelling generative approach for sequential decision making or planning: by progressively denoising stochastic inputs, these models synthesize smooth, reliable action trajectories [2], [3]. Diffusion-based control has shown strong promise in motion planning [4] and imitation learning [5], delivering improved sample efficiency and training stability relative to standard reinforcement learning baselines, and thereby advancing the practicality of visuomotor policy learning. [39], [54], [55]

Despite recent progress , leading diffusion-based visuomotor policies (e.g., Diffusion Policy [6], 3D Diffusion Policy (DP3) [7]) typically condition on point clouds and proprioception while omitting explicit inter-object geometric relations that are critical for spatial reasoning, contact prediction, and collision avoidance. This lack of a relational inductive bias often yields discontinuous or physically inconsistent trajectories in cluttered, multi-object environments. [56] In parallel, geometry-centric encoders (PointNet++ [8], Point Transformer [9]) capture structured 3D context, and trajectory optimizers (CHOMP and TrajOpt [10], [11]) enforce smoothness and safety via explicit constraints. A principled fusion of generative diffusion with geometry-aware physical reasoning – so that denoising respects relational structure and task constraints – remains largely open.

To bridge this gap, we propose `GeoDiff`: a **Geo**metry-Conditioned **Diff**usion Policy for refined robotic trajectory generation. `GeoDiff` derives object-centric geometric features by clustering point clouds into instances and encoding inter-object relations (e.g., centroids, pairwise distances), and uses these features to condition denoising so that sampled trajectories respect spatial structure and avoid collisions. At inference, `GeoDiff` draws multiple candidate trajectory and ranks them with a physics-aware score that balances smoothness, goal attainment, and safety constraints [12], [13]. Training further incorporates a differentiable physical-consistency loss to promote dynamically feasible motion [14]. Inspired by contact-aware manipulation [15] and physics-guided diffusion [16], this design achieves both trajectory diversity and physics robustness, yielding geometrically consistent plans in cluttered, multi-object scenes.

We evaluate `GeoDiff` on standard manipulation benchmarks, including Meta-World [17], DexArt [18], and Adroit [19]. Experiments show that `GeoDiff` consistently surpasses diffusion-based and optimization-based baselines, yielding over 15% gains in success rate and trajectory smoothness, underscoring the benefit of coupling geometric conditioning with physics-aware consistency in diffusion control.

In summary, our contributions are:

- `GeoDiff`: geometry-conditioned diffusion policy that explicitly encodes object-centric spatial relations for ma-

nipulation planning.

- Multi-sample trajectory refinement: generate and rank candidates using physics-based consistency metrics to select robust trajectories.
- Differentiable physical-consistency loss: enforce smoothness, goal accuracy, and safety during training to promote dynamically feasible motion.
- State-of-the-art results: significant improvements across diverse manipulation benchmarks over prior diffusion-based and optimization-based methods.

II. Related Work

A. Diffusion-based Policies for Robotic Control

Diffusion models have recently become a powerful paradigm for sequential decision making and control. Beyond planning-as-denoising [2], [3], Diffusion Policy and 3D Diffusion Policy (DP3) demonstrate effective visuomotor control with action- and 3D-conditioned denoising [6], [7]. A rich line of work studies diffusion for structured motion generation, including human motion diffusion [20], controllable guidance [21], and physics-informed human–object interactions [22]. Transformer-based diffusion for motion prediction [23] and large diffusion foundation models for manipulation [24] further underscore the scalability of this family. Language-conditioned diffusion policies extend to instruction-following manipulation [25], while unified planner–controller formulations for physics-based characters provide an integrated optimization perspective [26]. Recent advances in adaptive and test-time improved diffusion policies [27], [34] and diffusion for scene-level planning and optimization [28] reflect a growing trend toward incorporating prior knowledge and constraints during sampling; physics-guided motion diffusion also emphasizes physical plausibility in generation [29], [42], [57]. Compared with these methods, our work targets *robotic manipulation* and focuses on *explicit geometric conditioning* and *physics-aware refinement* to improve trajectory smoothness and safety.

B. Geometry-Aware and Relational Representations

Explicit 3D geometry is crucial for manipulation. Point cloud encoders such as PointNet/PointNet++ [8], [30], DGCNN [31], KPConv [32], and PointCNN [33] learn robust local and global features; recent pretraining and scaling studies (e.g., Point-BERT and PointNeXt) highlight the benefits of large-scale representation learning [34], [35]. For scene-level reasoning, representation designs including Hough voting for detection (VoteNet) [36] and SE(3)-equivariant attention [37] capture relational structure and invariances that are directly relevant to object-centric control. Our method builds on these insights by constructing object-centric clusters and relational features (centroids, robot–target distances, spatial configurations) that condition a diffusion policy, thus making spatial dependencies explicit rather than implicit. [43]

C. Physics-Consistent and Constraint-Guided Motion Generation

Classical motion planning emphasizes physical feasibility via smoothness, collision avoidance, and optimality. Trajectory optimization methods such as STOMP and ITOMP [38], Gaussian Process Motion Planning and its continuous-time variants [40], [41], and optimal sampling-based planning [44] provide principled mechanisms to handle constraints and search efficiency. In contact-rich domains, contact-implicit trajectory optimization explicitly reasons about impacts and mode switches [45]. Complementary to these methods, GeoDiff integrates *physics-aware* scoring at inference (smoothness, goal accuracy, collision safety) and a differentiable physical-consistency loss during training (Sec. III-B), bridging generative diffusion with constraint-aware planning.

III. Method

The proposed method, termed **GeoDiff**, generates robot manipulation trajectories conditioned on geometric observations via a diffusion-based generative framework. An overview of the overall pipeline is shown in Fig. 1, which includes five stages: (a) RGB-D perception, (b) geometry and task representation, (c) trajectory generation and sampling, (d) physics-aware evaluation, and (e) refined trajectory execution. GeoDiff leverages geometric features to guide the diffusion process, ensuring that the generated trajectories are physically consistent and collision-free.

The following subsections describe the main components in detail.

A. Clustering and Geometric Representation

We utilize the point cloud $P = \{p_i \mid p_i \in \mathbb{R}^3\}$ to represent the 3D geometry of the workspace. A clustering algorithm is applied to segment P into N clusters $\{C_1, C_2, \ldots, C_N\}$, each corresponding to an individual object instance. For each cluster C_i, its centroid is computed as:

$$c_i = \frac{1}{|C_i|} \sum_{p \in C_i} p, \tag{1}$$

and is regarded as the geometric center of the object.

To construct the conditional inputs for the diffusion model, we define a feature representation that includes the target object centroid c_{target}, the robot state $s_r = [T_r, R_r] \in \mathbb{R}^7$, where $T_r \in \mathbb{R}^3$ denotes the end-effector translation and $R_r \in \mathbb{R}^4$ represents its quaternion rotation. The Euclidean distance between the end-effector and the target is defined as $d = \|c_{target} - T_r\|$. The raw point cloud features P and a task description label l are also included. The label l describes the current manipulation goal and may be decomposed into multiple subgoals for complex tasks. For example, in a *soccer* task, the description can be divided into "reach the ball" and "push the ball toward the goal." These features are encoded as a conditional representation:

$$\mathcal{F} = f(c_{target}, s_r, d, P, l), \tag{2}$$

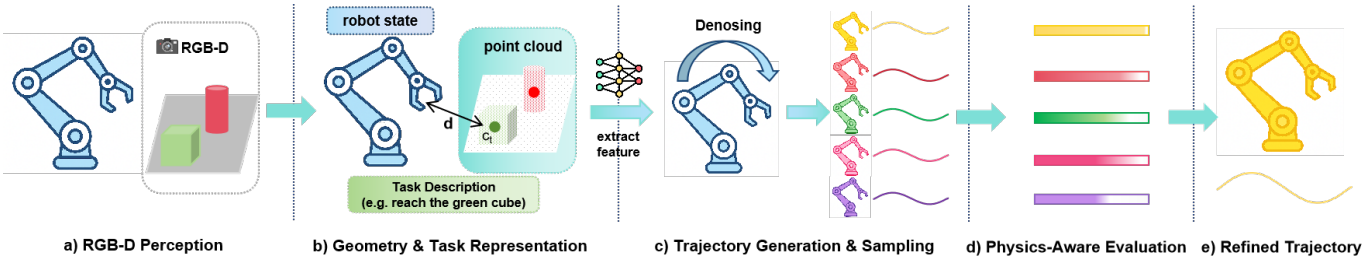

Fig. 1. Pipeline of the proposed GeoDiff method. The RGB-D perception module (a) captures both visual and depth information. The geometry and task representation stage (b) encodes the robot state, point cloud, and task goal into geometric features. The conditional diffusion model (c) generates multiple trajectory samples. These trajectories are then evaluated by physics-aware metrics (d), and the best-performing refined trajectory (e) is executed by the robot.

and are fed into the diffusion model, which iteratively refines Gaussian noise into a feasible action trajectory:

$$\tau = \{a_1, a_2, \ldots, a_T\}, \quad a_i = (a_{\text{tran}}, a_{\text{grip}}), \tag{3}$$

where $a_{\text{tran}} \in \mathbb{R}^3$ denotes the translational action of the end-effector, corresponding directly to its Cartesian position (x, y, z) in the robot workspace, and $a_{\text{grip}} \in \mathbb{R}$ represents the scalar gripper control command. Specifically, $a_{\text{grip}} = -1$ indicates that the gripper is closing, whereas $a_{\text{grip}} = 1$ indicates that the gripper is opening.

This formulation explicitly separates the robot state s_r from the action sequence τ, allowing the diffusion model to learn a distribution over physically consistent control commands.

The forward diffusion process gradually perturbs a clean trajectory sample $\tau^{(0)}$ into a sequence of noisy latent variables $\tau^{(k)}$ along diffusion step k:

$$q(\tau^{(k)} \mid \tau^{(0)}) = \mathcal{N}\Big(\tau^{(k)}; \sqrt{\bar{\alpha}_k}\tau^{(0)}, (1 - \bar{\alpha}_k)\mathbf{I}\Big), \tag{4}$$

where $\bar{\alpha}_k = \prod_{i=1}^{k}(1 - \beta_i)$ denotes the cumulative noise schedule. During the reverse denoising phase, the model predicts the clean trajectory distribution conditioned on $\mathcal{F}$ as:

$$p_\theta\big(\tau^{(k-1)} \mid \tau^{(k)}, \mathcal{F}\big) = \mathcal{N}\Big(\tau^{(k-1)}; \mu_\theta\big(\tau^{(k)}, k, \mathcal{F}\big), \Sigma_\theta\big(\tau^{(k)}, k, \mathcal{F}\big)\Big). \tag{5}$$

This conditional reverse process allows the model to iteratively reconstruct a geometrically consistent and dynamically feasible end-effector trajectory guided by $\mathcal{F}$.

B. Trajectory Evaluation under Physical Constraints

To enhance robustness, GeoDiff generates N action trajectories in parallel during inference. All trajectories share the same initial robot state, scene configuration, and conditional inputs $\mathcal{F}$, and are produced in a single batched diffusion process. The diversity among candidates comes solely from independent stochastic noise samples injected into each trajectory within the batch. This ensures fair comparison under identical conditions and enables selecting the best physically feasible trajectory.

Each trajectory $\tau = \{a_1, a_2, \ldots, a_T\}$ is composed of discrete actions $a = [a_{\text{tran}}, a_{\text{grip}}] \in \mathbb{R}^4$, where $a_{\text{tran}} \in \mathbb{R}^3$ denotes the translational action of the end-effector and $a_{\text{grip}} \in \mathbb{R}$ represents the gripper opening action. Each trajectory is evaluated based on its physical consistency, considering motion smoothness, goal accuracy, and collision safety.

We define the objective scoring function $\mathcal{J}(\tau)$ as:

$$\mathcal{J}(\tau) = \begin{cases} \alpha \cdot \mathcal{S}(\tau) + \beta \cdot \mathcal{G}(\tau), & \text{if } \mathcal{C}(\tau) = 0, \\ 0, & \text{if } \mathcal{C}(\tau) = 1, \end{cases} \tag{6}$$

where α and β are weighting coefficients, and $\mathcal{C}(\tau)$ indicates whether a collision occurs during the trajectory. If all sampled trajectories yield $\mathcal{J}(\tau) = 0$, a re-sampling process is triggered.

The smoothness metric $\mathcal{S}(\tau)$ quantifies the continuity of the translational motion of the end-effector. First, the average second-order difference of the trajectory is computed as:

$$\kappa(\tau) = \frac{1}{T-2} \sum_{t=2}^{T-1} \|a_{\text{tran},t+1} - 2a_{\text{tran},t} + a_{\text{tran},t-1}\|. \tag{7}$$

Then, the bounded smoothness score is defined as:

$$\mathcal{S}(\tau) = \max\big(0,\, 1 - \kappa(\tau)/\sigma_s\big), \tag{8}$$

where σ_s is a scaling factor set to the median smoothness value measured from expert demonstration trajectories. A higher $\mathcal{S}(\tau)$ indicates smoother and more dynamically consistent motion.

The goal accuracy term $\mathcal{G}(\tau)$ measures how close the final end-effector position is to the target centroid c_{target}:

$$\mathcal{G}(\tau) = 1 - \frac{\|a_{\text{tran},T} - c_{target}\|}{D_{max}}, \tag{9}$$

where D_{max} represents the maximum possible distance between the initial and target positions within the workspace.

The collision indicator $\mathcal{C}(\tau)$ determines whether any translational action point violates spatial safety constraints. Instead of computing the minimum distance to each obstacle, we utilize the Euclidean Signed Distance Field (ESDF) $D(\cdot)$, which returns the signed distance from any 3D position to the nearest obstacle surface (positive outside and negative inside the obstacle region). Given a safety margin ϵ, the collision indicator is defined as:

$$\mathcal{C}(\tau) = \begin{cases} 0, & \text{if } D(a_{\text{tran},t}) \geq \epsilon,\ \forall t = 1, \ldots, T, \\ 1, & \text{if } \exists\, t \text{ such that } D(a_{\text{tran},t}) < \epsilon, \end{cases} \tag{10}$$

Algorithm 1 GeoDiff: Geometrically-Conditioned Diffusion

Input: Point cloud P, robot state s_r, task description l, distance d, number of samples N
Cluster P to obtain centroids $\{c_i\}$; select target c_{target}
Construct conditional features $\mathcal{F} = f(c_{\text{target}}, s_r, d, P, l)$
Initialize noisy batch of trajectories:

$$\tau^{(K)} \sim \mathcal{N}(0, I), \quad \tau^{(K)} \in \mathbb{R}^{N \times T \times D_{\text{action}}}$$

for $k = K$ down to 1 **do**
    Reverse diffusion step (parallel for all N samples):

$$\tau^{(k-1)} \leftarrow \text{Denoise}\big(\tau^{(k)}, \mathcal{F}, k\big)$$

end for
Obtain final trajectories batch $\tau^{(0)} = \{\tau_1^{(0)}, \ldots, \tau_N^{(0)}\}$
Evaluate each trajectory independently ($n = 1, \ldots, N$):

$$\mathcal{J}(\tau_n^{(0)}) = \begin{cases} \alpha \mathcal{S}(\tau_n^{(0)}) + \beta \mathcal{G}(\tau_n^{(0)}), & \text{if } \mathcal{C}(\tau_n^{(0)}) = 0 \\ 0, & \text{otherwise} \end{cases}$$

Output: $\tau^* = \arg\max_{\tau_n^{(0)}} \mathcal{J}(\tau_n^{(0)})$

where $D(a_{\text{tran},t})$ denotes the ESDF value at the translational position of the end-effector. When $\mathcal{C}(\tau) = 1$, the trajectory is considered physically invalid and is assigned a zero score.

Finally, the optimal trajectory is selected as:

$$\tau^* = \arg\max_{\tau_i} \mathcal{J}(\tau_i), \tag{11}$$

ensuring that the chosen trajectory best satisfies the overall physical constraints of smoothness, accuracy, and safety.

Algorithm 1 illustrates the complete process of diffusion-based trajectory sampling and evaluation under physical constraints. It summarizes how GeoDiff performs iterative denoising conditioned on geometric features, evaluates multiple stochastic action trajectories, and selects the optimal motion plan τ^*.

C. Loss Function Design

During training, the objective is to encourage the diffusion model to generate action trajectories that are physically consistent, smooth, and goal-directed. Accordingly, the total loss function is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{phys}}, \tag{12}$$

where $\mathcal{L}_{\text{recon}}$ denotes the reconstruction loss that drives the model to accurately denoise actions during the diffusion process, and $\mathcal{L}_{\text{phys}}$ represents a differentiable physical consistency loss that enforces the physical constraints introduced in Section III-B.

a) Reconstruction loss.: The reconstruction term $\mathcal{L}_{\text{recon}}$ follows the standard denoising diffusion objective, guiding the network to predict the Gaussian noise added to clean action trajectories:

$$\mathcal{L}_{\text{recon}} = \text{MSE}\Big(\epsilon_k,\, \epsilon_\theta\big(\sqrt{\bar{\alpha}_k}\,\tau^{(0)} + \sqrt{1-\bar{\alpha}_k}\,\epsilon_k,\, k, \mathcal{F}\big)\Big), \tag{13}$$

where $\tau^{(0)} = \{a_1, a_2, \ldots, a_T\}$ denotes the clean ground-truth action trajectory, ϵ_k is Gaussian noise sampled at diffusion step k, and $\mathcal{F}$ is the geometric conditional representation. This objective ensures that the model learns to reconstruct physically meaningful actions from noisy inputs.

b) Physical consistency loss.: To further enhance the realism and safety of generated trajectories, the physical consistency loss $\mathcal{L}_{\text{phys}}$ combines continuity, goal accuracy, and collision regularization:

$$\mathcal{L}_{\text{phys}} = \lambda_s \mathcal{L}_{\text{cont}} + \lambda_g \mathcal{L}_{\text{goal}} + \lambda_c \mathcal{L}_{\text{col}}, \tag{14}$$

where $\mathcal{L}_{\text{cont}}$ enforces local smoothness, $\mathcal{L}_{\text{goal}}$ drives the final action toward the target, and $\mathcal{L}_{\text{col}}$ penalizes proximity to obstacles.

The detailed definitions are:

$$\mathcal{L}_{\text{cont}} = \kappa(\tau)^2, \quad \text{(defined in Section III-B)} \tag{15}$$

$$\mathcal{L}_{\text{goal}} = \|a_{\text{tran},T} - c_{target}\|^2, \tag{16}$$

$$\mathcal{L}_{\text{col}} = \frac{1}{T} \sum_{t=1}^{T} \big[\min\big(0,\, D(a_{\text{tran},t})\big)\big]^2. \tag{17}$$

where $D(a_{\text{tran},t})$ is the ESDF value at the translational component of the end-effector. Negative values indicate penetration into obstacles and are penalized accordingly. Note that $\kappa(\tau)$ is squared here to provide smoother gradients during optimization, while the evaluation metric in Section III-B uses the non-squared form for stable and interpretable scoring.

The overall training procedure of GeoDiff is summarized in Algorithm 2, which jointly optimizes the reconstruction and physical consistency objectives.

Algorithm 2 Training Procedure of GeoDiff

Input: Training dataset $\mathcal{D}$ with expert trajectories
for each batch $(P, s_r, l, \tau^{(0)}) \in \mathcal{D}$ **do**
    Sample diffusion step $k \sim \text{Uniform}\{1, \ldots, K\}$
    Sample noise $\epsilon_k \sim \mathcal{N}(0, I)$
    Add noise: $\tau^{(k)} = \sqrt{\bar{\alpha}_k}\,\tau^{(0)} + \sqrt{1-\bar{\alpha}_k}\,\epsilon_k$
    Construct conditional features $\mathcal{F} = f(c_{\text{target}}, s_r, d, P, l)$
    Predict noise: $\hat{\epsilon}_\theta = \epsilon_\theta(\tau^{(k)}, k, \mathcal{F})$
    Compute $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{phys}}$
    Update $\theta \leftarrow \theta - \eta \nabla_\theta \big(\mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{phys}}\big)$
end for
Output: Trained model parameters θ

D. Implementation Details

We apply the DBSCAN algorithm for point cloud clustering, which adaptively determines the number of object instances and remains robust to noise. The diffusion policy follows a convolutional network-based architecture with DDIM sampling used as the noise scheduler. The model is trained for 1000 epochs on simple MetaWorld tasks and 3000 epochs for more complex simulated environments, using a batch size of 128. Training involves 100 diffusion steps, while inference is performed with 10 denoising steps and $N_{sample} = 5$ trajectory candidates are generated per trial.

IV. Experiments

In this section, we evaluate the proposed **GeoDiff** framework across diverse simulation environments and manipulation tasks. We first describe the simulation setup, baselines, and evaluation metrics, followed by quantitative and qualitative analyses.

A. Benchmarks and Environments

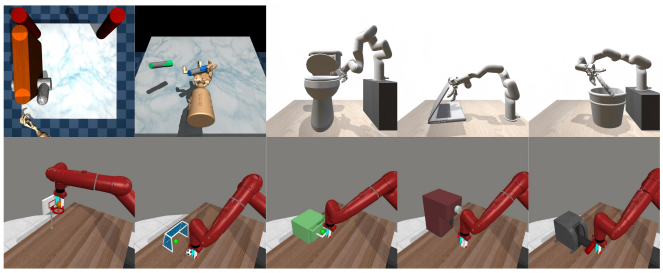

Fig. 2. Simulation environments for implementing and evaluating **GeoDiff**, covering three task categories: *Push&Pull*, *Placement*, and *Reaching*.

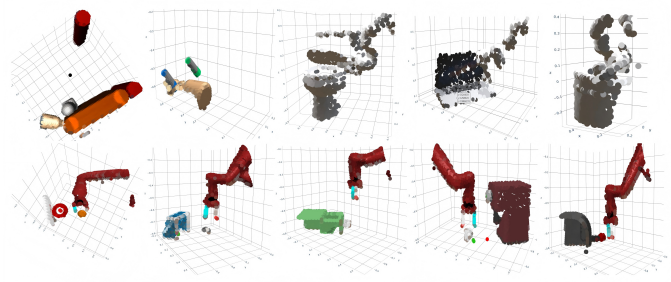

Fig. 3. Point cloud observations used in **GeoDiff** across representative tasks from Adroit, DexArt, and Meta-World. Each point cloud captures the spatial geometry of the robot, manipulated objects, and environment, providing the pose information for diffusion-based policy learning.

We evaluate **GeoDiff** on three representative robot manipulation benchmarks: **Adroit** [19], **DexArt** [18], and **Meta-World** [17], covering dexterous hand control [48]–[50], articulated-object [51], [52], and general tabletop tasks [53] respectively, as illustrated in Fig. 2.

Adroit. Includes three dexterous-hand tasks: *Door*, *Hammer*, and *Pen*. The action space ranges from 23 to 27 dimensions (finger joints and wrist rotations), and the observation space is 38 to 45 dimensions, containing the angular positions of the finger joints, the pose of the palm, and the state of the manipulated object.

DexArt. Consists of four articulated-object tasks: *Bucket*, *Faucet*, *Laptop*, and *Toilet*. Actions are 22-dimensional, representing both arm and hand joint positions. Observations combine proprioception (joint angles, pose, and velocity) with point cloud inputs for geometric perception.

Meta-World. Comprises 50 tabletop manipulation tasks, which can be grouped into three categories: *Placement* (e.g., peg-insert-side, pick-place), *Reaching* (e.g., reach, reach-wall), and *Push&Pull* (e.g., push, stick-push, coffee-pull). Its 4-dimensional action space controls end-effector movement (3D position + gripper), with 18-dimensional observations describing object positions and poses.

These benchmarks jointly test **GeoDiff** across varying control complexities and geometric conditions, under a unified simulation setup built on the MuJoCo and SAPIEN physics engines.

B. Data Generation and Configuration

Expert demonstration data are generated differently across the three benchmarks, reflecting their varying task structures and control complexities. For the **Meta-World** benchmark, expert trajectories are collected using built-in *scripted policies*, ensuring stable and consistent demonstrations for each task. In the **Adroit** domain, expert data is obtained from agents trained by the *VRL3* [46] algorithm, which learns dexterous manipulation through reinforcement learning with vision-based observations. For **DexArt**, demonstrations are collected from agents trained via the *PPO* [47] algorithm after convergence, providing high-quality trajectories for articulated-object control.

GeoDiff learns from these expert-generated trajectories to model the distribution and continuity of successful actions. For Meta-World and Adroit, we generate 10 expert trajectories per task, while DexArt provides 100 expert trajectories per task. To improve data quality and reduce randomness during collection, we first generate $5N$ trajectories when N samples are required for training, then select the top N trajectories with the highest success scores as the final training set. A summary of the benchmarks and their data generation configurations is provided in **Table I**.

TABLE I
BENCHMARKS AND EXPERT DATA CONFIGURATIONS IN **GEODIFF**. TRAJ./SEL. DENOTES THE NUMBER OF EXPERT TRAJECTORIES PER TASK AND THE CORRESPONDING SELECTION STRATEGY.

Benchmark	Tasks	Alg.	Traj./Sel.
Meta-World	Push&Pull, Placement, Reaching	Scripted	10 / 50
Adroit	Door, Hammer, Pen	VRL3	10 / 50
DexArt	Bucket, Faucet, Laptop, Toilet	PPO	100 / 500

This setup ensures consistent supervision quality across environments while maintaining diversity in control styles and motion distributions, which is crucial for learning the physically and geometrically consistent diffusion behaviors of **GeoDiff**.

C. Baselines

We compare **GeoDiff** with two representative diffusion-based visuomotor policy methods.

Diffusion Policy (DP) [6]: A visuomotor policy trained via action-space denoising diffusion, which refines noisy actions into executable trajectories conditioned on visual and proprioceptive observations.

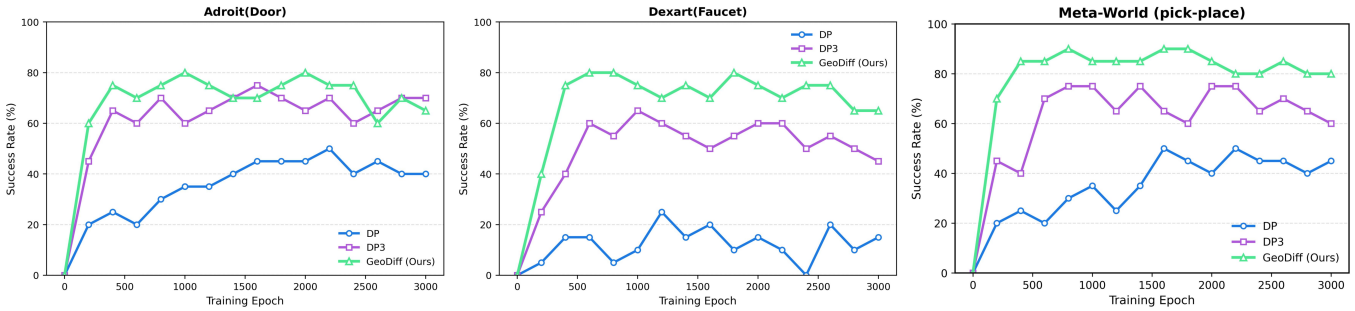

Fig. 4. **Convergence comparison on DexArt, Adroit, and Meta-World.** GeoDiff converges substantially faster (200–400 epochs) and reaches higher stable success rates than DP3 (400–600) and DP (1000–1200), showing improved stability and reduced oscillation.

3D Diffusion Policy (DP3) [7]: An extension of DP that incorporates 3D spatial representations from point clouds or depth maps, enhancing generalization to unseen objects and scenes.

Both serve as strong diffusion-policy baselines. Unlike them, **GeoDiff** explicitly models geometric correspondence and spatial diffusion priors, enabling physically consistent control across diverse manipulation tasks.

TABLE II
OVERALL TASK SUCCESS RATE (SR) ACROSS THREE BENCHMARKS. HIGHER VALUES INDICATE BETTER TASK PERFORMANCE.

Method	Meta-World(50) ↑	Adroit ↑	DexArt ↑	Avg. ↑
DP [6]	55.4	36.67	47.0	53.82
DP3 [7]	72.5	67.22	63.5	70.39
GeoDiff (Ours)	**88.65**	**70.78**	**71.08**	**86.48**

D. Analysis of Geometric Consistency and Efficiency

We conduct a comprehensive quantitative analysis of **GeoDiff**, emphasizing its geometric consistency, physical feasibility, and computational efficiency across three representative benchmarks: Adroit, DexArt, and Meta-World. This section jointly discusses evaluation indicators and results, demonstrating how the proposed geometry-aware diffusion framework achieves accurate, smooth, and efficient manipulation policies.

1) Task success and accuracy. Table II summarizes the overall success rate (SR) across three representative benchmarks. **GeoDiff** achieves the highest performance in all settings, reaching **88.65%** on Meta-World, **70.78%** on Adroit, and **71.08%** on DexArt, and surpassing diffusion-based baselines DP [6] and DP3 [7] by large margins. On average, **GeoDiff improves task performance by 16.1% over DP3 and 32.7% over DP**, demonstrating the effectiveness of incorporating explicit geometric priors into the diffusion process.

Tables III further validate these improvements. On Meta-World, GeoDiff achieves substantial gains across *Placement*, *Push&Pull*, and *Reaching* categories. For instance, SR improves from **12.3%** → **85.33%** on *pick-place* and from **5.0%** → **72.0%** on *shelf-place*, reflecting stronger spatial alignment and fine-grained contact reasoning. Similarly, on dexterous and articulated-object tasks, GeoDiff increases performance from **85.67%** → **90.0%** on *Hammer* (Adroit) and from **58.33%** → **74.33%** on *Faucet* (DexArt), indicating improved robustness under high-DoF motion and complex geometric constraints.

TABLE III
TASK SUCCESS RATE (SR%) COMPARISON ACROSS META-WORLD, ADROIT, AND DEXART BENCHMARKS. GEODIFF CONSISTENTLY OUTPERFORMS DIFFUSION-BASED BASELINES ACROSS DIVERSE MANIPULATION CATEGORIES AND CONTROL COMPLEXITIES.

Benchmark / Category	Task	DP	DP3	GeoDiff
Meta-World				
Placement	pick-place	12.3	63.0	**85.33**
	shelf-place	5.0	43.33	**72.0**
	pick-out-of-hole	11.0	31.67	**66.67**
	sweep-into	10.0	25.0	**55.33**
Push&Pull	handle-pull	8.33	29.33	**36.0**
	push-wall	25.0	50.0	**60.67**
	stick-pull	45.0	66.0	**72.0**
Reaching	hand-insert	4.33	13.0	**40.0**
	soccer	12.33	27.67	**56.67**
	bin-picking	15.67	74.0	**80.0**
Adroit				
Dexterous Hand	Door	45.0	70.0	**73.33**
	Hammer	42.33	85.67	**90.0**
	Pen	22.67	46.0	**49.0**
DexArt				
Articulated Objects	Bucket	45.0	46.67	**61.0**
	Faucet	19.67	58.33	**74.33**
	Laptop	71.0	**82.0**	81.0
	Toilet	52.33	67.0	**68.0**

Overall, these results confirm that **geometry-aware conditioning provides powerful structural cues** that enable more accurate, reliable, and generalizable policy generation across diverse manipulation types.

2) Convergence efficiency. Fig. 4 compares the training convergence behavior of GeoDiff, DP, and DP3 on representative tasks from Meta-World (*pick-place*), Adroit (*door*), and DexArt (*faucet*). Across all benchmarks, **GeoDiff converges substantially faster and reaches higher stable performance**,

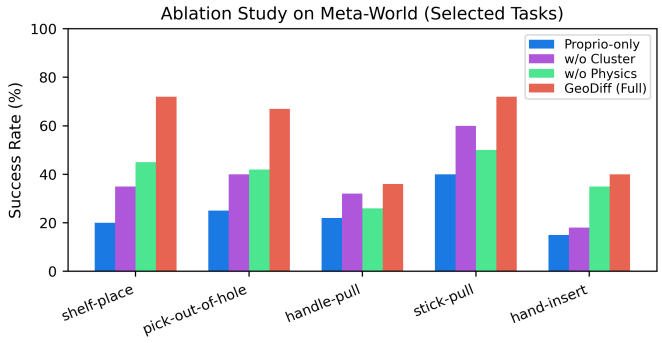

Fig. 5. Ablation study results highlighting the effect of geometric clustering and physics-based evaluation.

achieving a performance plateau within **200–400 epochs**, while DP3 requires **400–600 epochs** and DP does not converge until after **1000–1200 epochs**. This verifies that geometry-aware conditioning greatly improves learning efficiency.

Moreover, GeoDiff presents smoother optimization curves with less oscillation and only mild late-stage decay, whereas DP and DP3 show stronger fluctuations and lower peak performance. For example, on *pick-place*, GeoDiff reaches **80%+ SR** within 400 epochs, while DP3 remains below 60% and DP below 30%. Similar trends appear on Adroit and DexArt, demonstrating improved training stability and sample efficiency.

E. Ablation Study

To validate the effectiveness of each component in **GeoDiff**, we conduct a comprehensive ablation study on representative tasks from Meta-World, Adroit, and DexArt. The proposed system is decomposed into three key modules: (1) geometric representation and clustering, (2) physics-aware trajectory evaluation, and (3) physical consistency loss design. For clarity, all ablation results report the task success rate (SR), and where applicable, collision rate (CR) and trajectory smoothness (TS).

1) Effect of geometric representation. We conduct an ablation study on five representative Meta-World tasks to assess the contribution of explicit geometry-aware features, comparing GeoDiff with two reduced variants: *Proprio-only* (removing point cloud input) and *w/o Cluster* (removing object-level segmentation). As shown in Fig. 5, eliminating structured geometric representation leads to clear performance degradation, especially in **Placement** and **Reaching** tasks that require accurate spatial localization. These results indicate that object-level geometric cues play a key role in enabling reliable and effective manipulation behavior.

2) Effect of physics-aware evaluation. For **Push&Pull** tasks such as *handle-pull* and *stick-pull*, where the goal is implicitly defined by environment interaction rather than a fixed spatial target, geometric perception alone provides limited improvement. As shown in Fig. 5, performance gains mainly result from the *physics-aware trajectory evaluation* module, which filters collision-prone and unstable motions

TABLE IV

EFFECT OF PHYSICS-AWARE EVALUATION ON TRAJECTORY SMOOTHNESS $\mathcal{S}(\tau)$ AND COLLISION RATE (CR) ON SELECTED META-WORLD TASKS. HIGHER $\mathcal{S}$ AND LOWER CR INDICATE BETTER TRAJECTORY QUALITY.

Task	$\mathcal{S}(\tau)$ (w/o / ours)	CR (%) (w/o / ours)
pick-out-of-hole	0.79 / **0.97**	55.0 / **20.0**
hand-insert	0.76 / **0.96**	78.0 / **55.0**
handle-pull	0.83 / **0.98**	5.0 / 5.0
shelf-place	0.87 / **0.96**	32.0 / **8.0**

to produce smoother and more feasible trajectories. This highlights the importance of physical feasibility in interaction-dominant manipulation tasks.

3) Effect of physical consistency loss. We further evaluate the impact of the physics-aware objective on tasks with high collision sensitivity or strong smoothness requirements in Meta-World. As shown in Table IV, GeoDiff substantially reduces collision rates on constrained insertion tasks such as *pick-out-of-hole* and *shelf-place*, and markedly improves trajectory smoothness on motion-continuity–critical tasks such as *hand-insert* and *handle-pull*. These results confirm that enforcing physical feasibility is essential for generating safe and stable trajectories beyond geometric perception alone.

V. CONCLUSION

We presented **GeoDiff**, a geometry-conditioned diffusion framework for robotic manipulation. The method leverages object-centric geometric representation and a physics-aware trajectory evaluation module to generate smooth, accurate, and collision-free motion plans. A differentiable physical-consistency loss further improves training stability and feasibility.

Experiments on Meta-World, Adroit, and DexArt demonstrate that GeoDiff achieves state-of-the-art performance, significantly improving success rate, convergence efficiency, and trajectory quality over diffusion-based baselines. Ablation results confirm the effectiveness of geometric conditioning and physics-driven refinement.

However, GeoDiff provides smaller gains in tasks requiring strong environment feedback (e.g., complex Push&Pull behaviors), where static geometric priors and offline physical constraints are insufficient. Future work will integrate real-time interaction perception and dynamic contact modeling to enhance adaptability.

REFERENCES

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, pp. 1334–1373, 2016.

[2] M. Janner, Q. Li, and S. Levine, "Diffuser: Planning with diffusion for control," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022.

[3] A. Ajay, M. Xu, S. Xu, D. Pathak, and A. Gupta, "Trajectory diffusion for motion planning," in *Proc. Robot. Sci. Syst. (RSS)*, 2023.

[4] C. Wang, X. Ma, J. Zhang, and Y. Zhu, "MimicPlay: Long-horizon imitation learning by watching human play," in *Proc. Conf. Robot. Learn. (CoRL)*, 2023.

[5] Z. Chen and K. Fan, "An online trajectory guidance framework via imitation learning and interactive feedback in robot-assisted surgery," *arXiv preprint*, 2025.

[6] L. Chi, H. Jiang, X. Ma, and P. Stone, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proc. Conf. Robot. Learn. (CoRL)*, 2023.

[7] Z. Zhao, J. Zhang, and Y. Zhu, "3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations," in *Proc. Conf. Robot. Learn. (CoRL)*, 2024.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[9] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16259–16268.

[10] A. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization for motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009, pp. 489–494.

[11] J. Schulman, J. Ho, A. Lee, I. Awwal, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," *Int. J. Robot. Res.*, vol. 32, no. 9–10, pp. 1155–1178, 2013.

[12] J. Li, L. Wu, and H. Lin, "Trajectory evaluation under physical constraints for robot planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023.

[13] Z. Wu, Y. Liu, X. Ma, and Y. Zhu, "Physics-aware diffusion for motion control," *arXiv preprint*, 2024.

[14] X. Ma, C. Wang, L. Chi, and Y. Zhu, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[15] J. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, "Contact-GraspNet: 6-DoF grasp detection using contact-level geometry," in *Proc. Robot. Sci. Syst. (RSS)*, 2021.

[16] Y. Yuan, L. Wang, and D. Fox, "Energy-guided diffusion policies for robot manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[17] T. Yu, D. Quillen, Z. He, R. Julian, A. Tamar, and S. Levine, "Meta-World: A benchmark for multi-task and meta reinforcement learning," in *Proc. Conf. Robot. Learn. (CoRL)*, 2020.

[18] C. Bao, H. Xu, Y. Qin, and X. Wang, "DexArt: Benchmarking generalizable dexterous manipulation with articulated objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[19] V. Kumar, Z. Xu, and E. Todorov, "Fast, strong and compliant pneumatic actuation for dexterous tendon-driven hands," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2013.

[20] G. Tevet, S. Atzmon, A. Shafir, and Y. Lipman, "Human motion diffusion model (MDM)," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[21] X. Li, J. Xu, and F. Gao, "Guided motion diffusion for controllable human motion synthesis," *arXiv preprint*, 2023.

[22] J. Zhang, W. Yang, and Y. Wang, "InterDiff: Generating 3D human-object interactions with physics-informed diffusion," *arXiv preprint*, 2023.

[23] L. Sun, H. Luo, and Z. Lin, "TransFusion: A transformer-based diffusion model for 3D human motion prediction," *arXiv preprint*, 2023.

[24] X. Zhou, J. Chen, and H. Wu, "RDT-1B: A diffusion foundation model for bimanual manipulation," *arXiv preprint*, 2024.

[25] J. Fang, Y. Huang, and H. Liang, "Language-guided manipulation with diffusion policies," *arXiv preprint*, 2024.

[26] L. Wang, Q. Sun, and X. Zhu, "UniPhys: Unified planner and controller with diffusion for physics-based character control," *arXiv preprint*, 2024.

[27] Y. Zhang, F. Yu, and T. Wang, "Adaptive diffusion policy optimization for robotic manipulation," *arXiv preprint*, 2024.

[28] P. Chen, R. Gao, and X. Wu, "SceneDiffuser: Diffusion-based generation, optimization and planning in 3D scenes," *arXiv preprint*, 2024.

[29] Y. Yuan, Z. Lin, and X. Xu, "PhysDiff: Physics-guided human motion diffusion model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[30] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[31] Y. Wang, Y. Sun, Z. Liu, S. Sarma, M. Bronstein, and J. Solomon, "Dynamic graph CNN for learning on point clouds (DGCNN)," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[32] H. Thomas, C. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019.

[33] Y. Li, R. Bu, M. Sun, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018.

[34] X. Yu, R. Liu, and J. Li, "Point-BERT: Pre-training 3D point cloud transformers with masked point modeling," *arXiv preprint*, 2021.

[35] Z. Qian, H. Wang, and Y. Zhao, "PointNeXt: Revisiting PointNet++ with improved training and scaling," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022.

[36] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "VoteNet: Deep Hough voting for 3D object detection in point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019.

[37] F. Fuchs, D. Worrall, V. Fischer, and M. Welling, "SE(3)-Transformers: 3D roto-translation equivariant attention," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[38] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2011.

[39] R. Wolf, Y. Shi, S. Liu, and R. Rayyes, "Diffusion Models for Robotic Manipulation: A Survey," *Frontiers in Robotics and AI*, 2025.

[40] M. Mukadam, X. Yan, and B. Boots, "Gaussian process motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016.

[41] M. Mukadam, X. Yan, and B. Boots, "Continuous-time Gaussian process motion planning via probabilistic inference," *Int. J. Robot. Res.*, vol. 37, no. 11, pp. 1311–1330, 2018.

[42] P. Nadeau, M. Rogel, I. Bilić, I. Petrović, and J. Kelly, "Generating Stable Placements via Physics-guided Diffusion Models," *arXiv preprint*, 2025.

[43] A. D. Vuong, M. N. Vu, and I. Reid, "Improving Robotic Manipulation with Efficient Geometry-Aware Vision Encoder," *arXiv preprint*, 2025.

[44] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "BIT*: Batch informed trees," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 892–914, 2015.

[45] T. Posa, S. Kuindersma, and R. Tedrake, "Contact-implicit trajectory optimization using variational contact-implicit dynamics," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 69–81, 2014.

[46] C. Wang, X. Luo, K. Ross, and D. Li, "Vrl3: A data-driven framework for visual deep reinforcement learning," *Advances in Neural Information Processing Systems*, 2022.

[47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint*, 2017.

[48] M. Andrychowicz *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[49] A. Handa *et al.*, "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020.

[50] Y. Chen *et al.*, "Bi-dexhands: Towards human-level bimanual dexterous manipulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2804–2818, 2023.

[51] B. Eisner, H. Zhang, and D. Held, "Flowbot3D: Learning 3D articulation flow to manipulate articulated objects," *arXiv preprint arXiv:2205.04382*, 2022.

[52] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 1647–1654, 2022.

[53] A. M. Wells, N. T. Dantam, A. Shrivastava, and L. Kavraki, "Learning feasibility for task and motion planning in tabletop environments," *IEEE Robotics and Automation Letters*, 2019.

[54] R. Xu, J. Zhang, M. Guo, Y. Wen, H. Yang, M. Lin, J. Huang, Z. Li, K. Zhang, L. Wang, Y. Kuang, M. Cao, and X. Liang, "A0: An Affordance-Aware Hierarchical Model for General Robotic Manipulation," *arXiv preprint*, 2025.

[55] H. Tong, Y. Zhang, S. Lueth, and G. Chalvatzaki, "Adaptive Diffusion Constrained Sampling for Bimanual Robot Manipulation," *arXiv preprint*, 2025.

[56] M. G. Tamizi, H. Honari, A. M. S. Enayati, and H. Najjaran, "A Cross-Environment and Cross-Embodiment Path Planning Framework via a Conditional Diffusion Model (GADGET)," *arXiv preprint*, 2025.

[57] L. Bai, C. Du, L. Shao, X. Yang, and X. Chen, "Diffusion Model in Robotics: A Comprehensive Review," *SSRN preprint*, 2025.