DexHandDiff: Interaction-aware Diffusion Planning for Adaptive Dexterous Manipulation

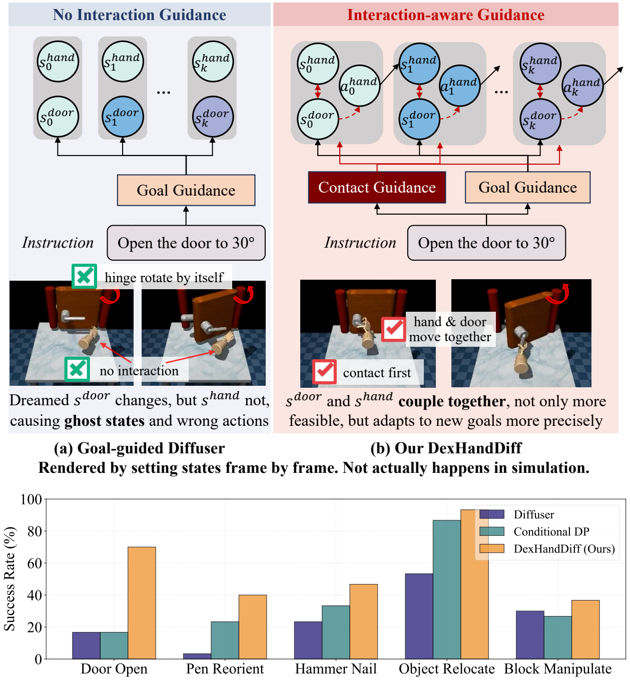
Zhixuan Liang[1,2†] Yao Mu[1] Yixiao Wang[2] Tianxing Chen[1] Wenqi Shao[1]
Wei Zhan[2] Masayoshi Tomizuka[2‡] Ping Luo[1‡] Mingyu Ding[2]
[1]The University of Hong Kong [2]University of California, Berkeley
{zxliang, ymu, pluo}@cs.hku.hk {yixiao_wang, wzhan, tomizuka, myding}@berkeley.edu
https://dexdiffuser.github.io/

Abstract

Dexterous manipulation with contact-rich interactions is crucial for advanced robotics. While recent diffusion-based planning approaches show promise for simple manipulation tasks, they often produce unrealistic ghost states (e.g., the object automatically moves without hand contact) or lack adaptability when handling complex sequential interactions. In this work, we introduce DexHandDiff, an interaction-aware diffusion planning framework for adaptive dexterous manipulation. DexHandDiff models joint state-action dynamics through a dual-phase diffusion process which consists of pre-interaction contact alignment and post-contact goal-directed control, enabling goal-adaptive generalizable dexterous manipulation. Additionally, we incorporate dynamics model-based dual guidance and leverage large language models for automated guidance function generation, enhancing generalizability for physical interactions and facilitating diverse goal adaptation through language cues. Experiments on physical interaction tasks such as door opening, pen and block reorientation, object relocation, and hammer striking demonstrate DexHandDiff's effectiveness on goals outside training distributions, achieving over twice the average success rate (59.2% vs. 29.5%) compared to existing methods. Our framework achieves an average of 70.7% success rate on goal adaptive dexterous tasks, highlighting its robustness and flexibility in contact-rich manipulation.

1. Introduction

Dexterous manipulation, a cornerstone of advanced robotics with applications from service robotics to industrial automation, remains a challenging problem despite advances in reinforcement learning (RL) [2, 4, 8, 47, 52] and imitation learning [23, 34]. Recently, diffusion-based planning [1, 14, 24, 28] has emerged as a promising new representative of imitation learning, capable of learning intricate motion trajectories from demonstration data for smoother and more adaptable control. However, current diffusion approaches are primarily designed for simpler gripper-based



(a) Goal-guided Diffuser **(b) Our DexHandDiff**
Rendered by setting states frame by frame. Not actually happens in simulation.

(c) Performance Comparison on Physical Interaction Tasks

Figure 1. (a) Previous diffusers directly apply goal guidance to object states, which causes ghost states, where objects appear to move independently without hand contact-a physically impossible scenario. (b) DexHandDiff introduces contact guidance that jointly influences both hand/object states and hand actions, while maintaining tight state-action coupling. It prevents ghost states, and enables precise goal adaptation. (c) Quantitative comparisons with previous methods on goal-shifted interaction tasks.

[†]This work was done during Zhixuan's visit to UC Berkeley.
[‡]Corresponding authors.

(one Degree of Freedom) manipulation tasks, focusing on either trajectory completion or action replay by reaching target positions sequentially. They fall short in dexterous hand manipulation requiring part-aware precise interaction and exhibiting rich contact dynamics through multi-finger control and in-hand adjustment.

More specifically, existing diffusion on action models [14, 56] (*i.e.* models generating actions) excel in well-defined tasks but often lack generalizability in adapting to complex or new tasks with flexible interaction requirements. They necessitate continual data collection for new goal configurations even within the same dynamics, limiting their effectiveness in contact-rich interactions. In contrast, diffusion on state methods [1, 24, 37], including those adapted from video diffusion models for imitation learning [6, 16], will produce unrealistic "ghost states" in interaction tasks. As shown in Fig. 1 and Fig. 2, the visualizations are rendered by setting states frame by frame with predicted output from state-based methods, and show objects react independently of physical contact (*e.g.* drawers opening on their own before the manipulator reaches them), which cannot actually happen and would result in failure. This issue arises because the object states can't be directly controlled. Actions must first influence dexterous hand's states before impacting the object, revealing the importance of modeling state transitions for physics-driven interactions.

Thus, we propose DexHandDiff, an interaction-aware diffusion model tailored for adaptive dexterous manipulation that exhibits goal shifts or cost function variations while maintaining similar dynamics. DexHandDiff models joint state-action dynamics that takes the state output to guide and constrain the action output with realistic physical behavior. A dynamics model-based dual guide is incorporated to maintain coherence with dynamics observed in training data. It addresses the action-state consistency challenge first identified in Diffuser [24] which however prioritized generated state over action, as shown in Fig. 1.

Specifically, DexHandDiff adopts a goal-adaptive diffusion mechanism with dual-phase process. 1) At first, pre-contact phase, it guides the manipulator to align with the object's key contact point, such as a handle or the center of object, ensuring stable alignment before initiating physical interaction. 2) In the subsequent post-contact phase, it introduces joint guidance over both the manipulator and the object states, enabling fine-grained control to achieve the target state for the object. This sequential approach integrates both action diffusion that prevents premature influence on the object's state before contact, and state diffusion that ensures effective goal alignment throughout. By generating states and actions in an interaction-aware manner, DexHandDiff produces more coherent and realistic trajectories suited to complex tasks like tool using. Furthermore, to automate guidance function design, DexHandDiff introduces an approach using large language models in the text-to-reward paradigm, that can generalize across diverse goals and cost functions via language cues.

We conduct experiments on multiple dexterous manipulation tasks to evaluate DexHandDiff's effectiveness, covering both in-domain and goal-adaptability challenges, *e.g.*, adapting to new goal "door closing" from "90-degree door opening" training data. Results with up to 70.0% success rate on the 30-degree door task (vs. the next best 16.7% for Diffusion Policy) and 46.7% on the hammer nail half-drive task (vs. the next best 33.3% for Decision Diffuser), confirm DexHandDiff's robustness and adaptability in capturing complex hand-object-environment interactions.

In summary, DexHandDiff advances adaptive dexterous manipulation by: 1) We propose the first interaction-aware, goal-adaptive diffusion planner for dexterous manipulation, modeling manipulator-object-environment dependencies to handle sequential tasks with complex state transitions. 2) By jointly modeling state-action behaviors with dynamics-based dual guidance and LLM-based interaction guidance, DexHandDiff sets a new standard for adaptive planning in dexterous manipulation and for the first time extends text-to-reward concepts to diffusers. 3) Experimental validation on diverse dexterous manipulation tasks, demonstrating its robustness and adaptability. DexHandDiff achieves over twice the average success rate of the next best method (59.2% vs. 29.5%) across goal-directed tasks.

2. Related Works

Dexterous Manipulation. Dexterous manipulation [12, 13, 19, 20, 32, 40, 42, 45, 48, 50] with multi-fingered hands enables complex tasks in unstructured environments by mimicking human hand flexibility. Initially, traditional methods using trajectory optimization and precise dynamics models [36, 41], struggled with high-dimensional action spaces and contact-rich dynamics. This led to the adoption of reinforcement learning (RL) [10, 41, 52, 58] for handling complex, high-DOF (degree of freedom) interactions. However, RL requires extensive online exploration and carefully designed reward functions [11, 36] where inadequate reward shaping can hardly learn and it limits adaptability [55, 57]. Demonstration-based methods [57] reduce sample complexity, but they struggle to generalize across sequential, contact-rich tasks. Our DexHandDiff addresses these challenges by explicitly modeling hand-object-environment interactions, enabling goal-adaptive planning without intricate reward shaping, thus allowing for more efficient learning in complex, sequential tasks.

Diffusion-based Planning Methods. Planning with diffusion models has become prominent in imitation learning for robotic manipulation [1, 9, 14, 24, 28–30, 37]. Classifier-guided methods [24, 28] used task-specific classifiers to condition policies, while classifier-free ones integrated task variations within diffusion model [1]. However, classifier-free methods lack flexibility for zero-shot explicit conditioning tasks due to reliance on training data configurations. DexHandDiff addresses this by performing classifier-guided diffusion over both state and action spaces, enabling precise interaction and rich-contact dynamics planning for more realistic, complex and adaptable manipulation.

LLM-based Robot Policy Code Generation. Recent works [7, 27, 35, 51] have demonstrated the potential of LLMs in generating executable code for robotics tasks. Code as Policies [27] showed LLMs can effectively translate high-level task descriptions into functional robot control programs. Eureka [33] and Text2Reward [54] further advanced this direction by generating crucial parameters or complete reward functions from language descriptions, demonstrating well-structured prompts with comprehensive environment information can enable reliable reward shaping. Our work extends this text-to-code paradigm to imitation learning through diffusers. DexHandDiff provides a natural interface for LLM code generation through its guidance function formulation, bridging the gap between task specification and behavioral policies to learn.

3. Preliminary

3.1. Diffusion Model as Policy

We formulate the dexterous manipulation planning problem within the Markov Decision Process (MDP) framework [39], defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$. The objective is to find an optimal action sequence $\boldsymbol{a}^*_{0:T}$ that satisfies:

$$\boldsymbol{a}^*_{0:T} = \arg\max_{\boldsymbol{a}_{0:T}} \mathcal{J}(\boldsymbol{s}_0, \boldsymbol{a}_{0:T}) = \arg\max_{\boldsymbol{a}_{0:T}} \sum_{t=0}^{T} \gamma^t R(\boldsymbol{s}_t, \boldsymbol{a}_t),$$

where state transitions follow $\boldsymbol{s}_{t+1} = \mathcal{T}(\boldsymbol{s}_t, \boldsymbol{a}_t)$. (1)

Following [1, 24], we leverage diffusion models to address this planning problem by treating state and action trajectories $\boldsymbol{\tau}$ as sequential data. The reverse process of diffusion learns to denoise trajectories from a standard normal distribution through conditional probability $p_\theta(\boldsymbol{\tau}^{i-1} \mid \boldsymbol{\tau}^i)$. The model is trained to maximize the likelihood:

$$p_\theta\left(\boldsymbol{\tau}^0\right) = \int p\left(\boldsymbol{\tau}^N\right) \prod_{i=1}^{N} p_\theta\left(\boldsymbol{\tau}^{i-1} \mid \boldsymbol{\tau}^i\right) \mathrm{d}\boldsymbol{\tau}^{1:N}, \tag{2}$$

with the optimization objective inspired by ELBO,

$$\theta^* = \arg\min_\theta -\mathbb{E}_{\boldsymbol{\tau}^0}\left[\log p_\theta\left(\boldsymbol{\tau}^0\right)\right]. \tag{3}$$

For practical implementation, we adopt the simplified surrogate loss [22] that focuses on predicting the noise term:

$$\mathcal{L}_{\text{denoise}}(\theta) = \mathbb{E}_{i, \boldsymbol{\tau}^0 \sim q, \epsilon \sim \mathcal{N}}[||\epsilon - \epsilon_\theta(\boldsymbol{\tau}^i, i)||^2]. \tag{4}$$

3.2. Classifier-free Conditional Diffusion Policy

To generate high-reward trajectories, classifier-free guidance [15] has been transferred from image to trajectory generation [1]. This approach incorporates guidance signals $\boldsymbol{y}(\boldsymbol{\tau})$ directly in the noise prediction model by:

$$\hat{\epsilon} = \epsilon_\theta(\boldsymbol{\tau}^i, \varnothing, i) + \omega(\epsilon_\theta(\boldsymbol{\tau}^i, \boldsymbol{y}, i) - \epsilon_\theta(\boldsymbol{\tau}^i, \varnothing, i)), \tag{5}$$

where ω controls the guidance strength, and $\varnothing$ denotes the absence of conditioning. During sampling, trajectories are generated with the predicted modified noise $\hat{\epsilon}$.

3.3. Classifier-guided Diffusion Policy

Different from classifier-free diffusion models that condition relying solely on implicit representations within the training data, classifier-guided approach, enables direct reward or goal conditioning through gradient-based guidance.

For reward maximization, it introduces trajectory optimality $\mathcal{O}_t$ at timestep t, following a Bernoulli distribution where $p(\mathcal{O}_t = 1) = \exp(\gamma^t \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t))$. The diffusion process can be naturally extended to incorporate conditioning by sampling from perturbed distributions:

$$\tilde{p}_\theta(\boldsymbol{\tau}) = p(\boldsymbol{\tau} \mid \mathcal{O}_{1:T} = 1) \propto p_\theta(\boldsymbol{\tau}) p(\mathcal{O}_{1:T} = 1 \mid \boldsymbol{\tau}) \tag{6}$$

Under Lipschitz conditions on $p(\mathcal{O}_{1:T} \mid \boldsymbol{\tau}^i)$ [17], the reverse diffusion process follows:

$$p_\theta(\boldsymbol{\tau}^{i-1} \mid \boldsymbol{\tau}^i, \mathcal{O}_{1:T}) \approx \mathcal{N}(\boldsymbol{\tau}^{i-1}; \mu_\theta + \alpha\Sigma g, \Sigma), \tag{7}$$

where the guidance gradient g is:

$$\begin{aligned} g &= \nabla_{\boldsymbol{\tau}} \log p(\mathcal{O}_{1:T} \mid \boldsymbol{\tau})|_{\boldsymbol{\tau}=\mu_\theta} \\ &= \sum_{t=0}^{T} \gamma^t \nabla_{\boldsymbol{s}_t, \boldsymbol{a}_t} \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)|_{(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mu_t} = \nabla_{\boldsymbol{\tau}} \mathcal{J}(\mu_\theta). \end{aligned} \tag{8}$$

For discrete goal conditioned tasks, the constraint can be simplified by directly substituting conditional values at each diffusion timestep $i \in \{0, 1, ..., N\}$.

4. Analysis of Diffusion-based Planning Methods for Interaction-intensive Tasks

Current diffusion-based methods are widely adopted for robotic manipulation but reveal significant limitations when applied to dexterous, sequential interaction tasks. Table 1 provides an overview of prominent diffusion-based methods (including Diffuser [24], Decision Diffuser [1], Diffusion Policy [14] and our DexHandDiff), categorizing each by their conditioning approach, action generation method, and goal adaptability. In this section, we analyze these challenges across three key dimensions.

Action-only Diffusion is Limited in Explicit State Conditioning. Existing diffusion on action models like Diffusion Policy (DP) [14], excel in providing precise, consistent action control, benefiting from extensive training data and bypassing errors from inverse kinematics. They yield high performance when training data is sufficient and diverse. However, for tasks requiring variant multi-stage goals, action-only diffusion lacks the flexibility to perform explicit state guidance at intermediate stages, like aligning hand and object state at pre-grasp stage, hurting the adaptability of the whole planner. For example, DP trained on

Method	Diffusion on State or Action	Diffusion Condition Type	Action Gen Method	Goal Adaptability	No Ghost States	Interaction Aware
Diffuser [24]	State	Classifier-Guided	Inverse Dyn	✓	×	×
Decision Diffuser [1]	State	Classifier-Free	Inverse Dyn	× (if diverse data, then ✓)	×	×
Diffusion Policy [14]	Action	Classifier-Free	Direct	× (if diverse data, then ✓)	✓	×
DexHandDiff (Ours)	State & Action	Classifier-Guided	Direct	✓	✓	✓

Table 1. **Comparison of diffusion-based approaches for robot manipulation.** Quantitative results on door-opening are shown in Sec. 6.

data with opening the door to 90 degrees hardly adapt to open 30 or 60 degrees.

Ghost States in State-only Diffusion for Sequential Interaction. While state-based diffusion models offer the advantage of flexible goal specification, it is only effective in fully actuated tasks where all degrees of freedom (DoF) are directly controllable, such as MuJoCo [24, 46], and gripper pick-and-place (requiring only end-effector position control) [1, 14] tasks. In such scenarios, all states of the system can be manipulated directly. However, in contact-rich interaction task where indirect control exists, such as striking a nail with a hammer using a dexterous hand, additional uncontrollable DoFs, like the hammer head and nail positions, must be changed through transitions from the states of the hand. Applying generation across all states, including those of objects beyond the hand, will result in unrealistic "ghost states" where objects appear to move independently of contact but actually cannot, as illustrated in Fig. 1 and Fig. 2.

Classifier-free vs. Classifier-guided Adaptability. Classifier free diffusion models, valued for not requiring external classifiers, encode task variations directly within the model. This structure is effective for tasks with constrains in observed configurations, but with limited goal adaptability in zero-shot or new-task scenarios. For instance, in the push-T task, DP cannot directly adapt to new target positions due to the fixed goal in training data. In contrast, classifier-guided methods, such as ours, mitigate this limitation by offering adaptable, gradient-based guidance, enabling direct conditioning on new goals or rewards, enhancing flexibility across a range of tasks.

5. Method

5.1. Interaction-aware Diffusion-based Planning

To address these limitations, we propose DexHandDiff, an interaction-aware diffusion planning framework (Fig. 3), maintaining physical consistency and enabling flexible goal adaptation for dexterous manipulation.

Joint State-Action Diffusion Model. Our approach builds upon classifier-guided diffusion policies. But we jointly diffuse over the concatenated state-action space $\boldsymbol{\tau} = [(\boldsymbol{a}_0, \boldsymbol{s}_0), (\boldsymbol{a}_1, \boldsymbol{s}_1), ..., (\boldsymbol{a}_T, \boldsymbol{s}_T)]$, where state $\boldsymbol{s}$ includes both hand (24 joint angles and 3 position offsets) and task-specific object states (*i.e.* door hinge angle, pen pose *etc.*), and action $\boldsymbol{a}$ represents changes in controllable states (only hand joints and positions).

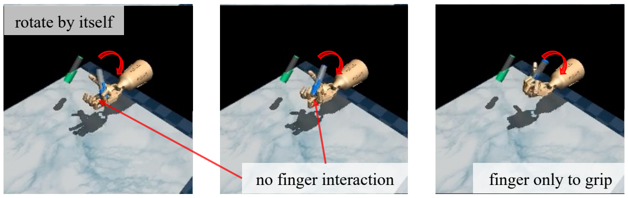


Figure 2. **Demonstration of ghost states on the pen reorientation task.** The visualizations are rendered by setting predicted states frame by frame, which cannot actually happen and will lead to failure. The pen appears to autonomously rotate to the desired pose without any hand manipulation, and the fingers look like moving to grasp the pen at the last frame.

This design choice directly addresses the above mentioned limitations: (1) By including states in the diffusion process, we enable explicit state conditioning and goal specification, overcoming the limitations of action-only approaches; (2) By classifier-guided diffusion, we allow flexible goal adaptation without exhaustive training data; (3) By jointly modeling states and actions, we maintain their physical coupling and prevent ghost states through carefully designed guidance. With denoised states guiding the generated actions, we effectively balance the state conditioning and action precision.

Extended Behavior Model and Energy Function. According to Eq. 6, the standard conditional diffusion follows:

$$\tilde{p}_\theta(\boldsymbol{\tau}) \propto p_\theta(\boldsymbol{\tau}) p(\mathcal{O}_{1:T} = 1 \mid \boldsymbol{\tau}) \propto p_\theta(\boldsymbol{\tau}) h(\boldsymbol{\tau}), \tag{9}$$

where we generalize $p(\mathcal{O}_{1:T} = 1 \mid \boldsymbol{\tau})$ as a behavior model $h(\boldsymbol{\tau})$. Then we further generalize this formulation through a product of experts framework [21], where each expert represents a specific behavior model:

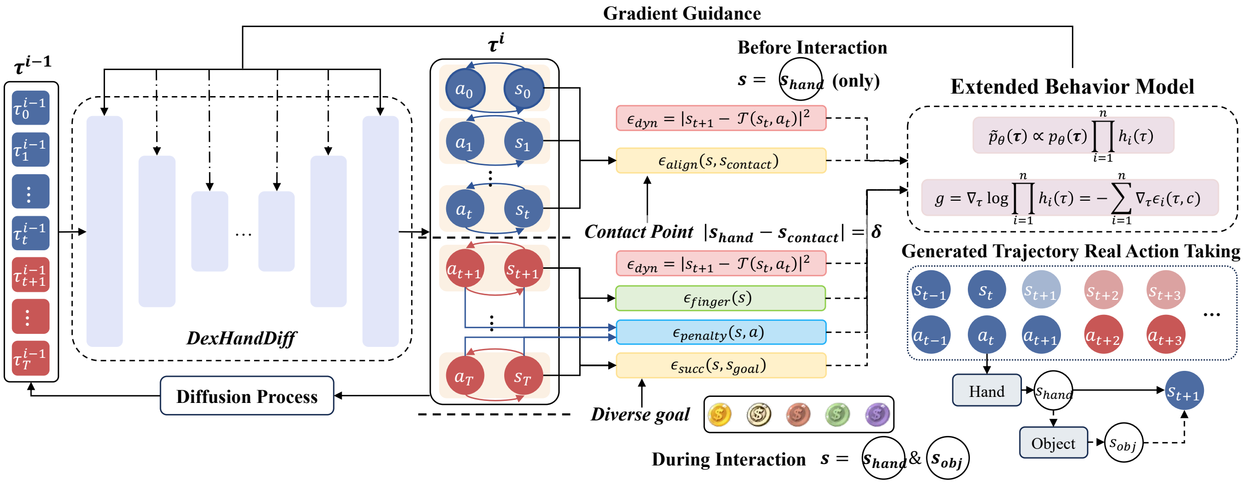
$$\tilde{p}_\theta(\boldsymbol{\tau}) \propto p_\theta(\boldsymbol{\tau}) \prod_{i=1}^{n} h_i(\boldsymbol{\tau}). \tag{10}$$

From the energy function perspective, each behavior model encoding task-specific objectives or constraints is:

$$h_i(\boldsymbol{\tau}, c) = \frac{1}{\int e^{-\varepsilon_i(\boldsymbol{\tau}, c)} d\boldsymbol{\tau}} e^{-\varepsilon_i(\boldsymbol{\tau}, c)}, \tag{11}$$

where $\varepsilon_i(\boldsymbol{\tau}, c)$ represents the energy function for the i-th guidance objective, with c denoting task-specific conditions. This formulation allows combining multiple objectives (*e.g.*, reaching the target state while maintaining physical consistency) via their respective guidance functions.

Under appropriate smoothness conditions, the guidance gradient g in the reverse diffusion process (Eq. 7) can be decomposed as the sum of individual guidance gradients:



Gradient Guidance
τ^{i-1}
τ_0^{i-1}
τ_1^{i-1}
τ_t^{i-1}
τ_{t+1}^{i-1}
τ_T^{i-1}
DexHandDiff
Diffusion Process
τ^i
a_0 s_0
a_1 s_1
a_t s_t
a_{t+1} s_{t+1}
a_T s_T
Before Interaction
$s =$ s_{hand} (only)
$\epsilon_{dyn} = |s_{t+1} - \mathcal{T}(s_t, a_t)|^2$
$\epsilon_{align}(s, s_{contact})$
Contact Point $|s_{hand} - s_{contact}| = \delta$
$\epsilon_{dyn} = |s_{t+1} - \mathcal{T}(s_t, a_t)|^2$
$\epsilon_{finger}(s)$
$\epsilon_{penalty}(s, a)$
$\epsilon_{succ}(s, s_{goal})$
Diverse goal
During Interaction $s =$ s_{hand} & s_{obj}
Extended Behavior Model
$\tilde{p}_\theta(\boldsymbol{\tau}) \propto p_\theta(\boldsymbol{\tau}) \prod_{i=1}^{n} h_i(\tau)$
$g = \nabla_\tau \log \prod_{i=1}^{n} h_i(\tau) = -\sum_{i=1}^{n} \nabla_\tau \epsilon_i(\tau, c)$
Generated Trajectory Real Action Taking
s_{t-1} s_t s_{t+1} s_{t+2} s_{t+3} ...
a_{t-1} a_t a_{t+1} a_{t+2} a_{t+3}
Hand
s_{hand}
s_{t+1}
Object
s_{obj}

Figure 3. **Framework of DexHandDiff.** DexHandDiff employs joint state-action diffusion with interaction-aware guidance. Before interaction (top middle), guidance aligns the hand to the object contact point. Upon contact (bottom middle), additional guidance steers both hand and object states towards the goal ("&" means state concatenation at input level), enforcing physical constraints and avoiding ghost states. A learned dynamics model further ensures consistency between states and actions. Our DexHandDiff utilizes extended behavior model to aggregate multiple condition terms to guide the diffusion process.

$$g = \nabla_{\boldsymbol{\tau}} \log \prod_{i=1}^{n} h_i(\boldsymbol{\tau}) = \sum_{i=1}^{n} \nabla_{\boldsymbol{\tau}} \log h_i(\boldsymbol{\tau}) = -\sum_{i=1}^{n} \nabla_{\boldsymbol{\tau}} \varepsilon_i(\boldsymbol{\tau}, c).$$

This enables integration of multiple guidance signals, each addressing different aspects of the interaction task, while maintaining a coherent optimization objective.

Dynamics-aware Generation. A key challenge in joint state-action diffusion is maintaining consistency between generated states and actions [24]. Our method addresses this through a learned dynamics model trained on demonstration data, constraining state-action generation via additional loss in diffusion training and serving as a guide in inference. By penalizing state-action pairs that violate observed dynamics, this guidance ensures our model maintains both state conditioning benefits and action feasibility.

$$\varepsilon_{\text{dyn}}(\boldsymbol{\tau}) = |\boldsymbol{s}_{t+1} - \mathcal{T}(\boldsymbol{s}_t, \boldsymbol{a}_t)|^2, \tag{12}$$

where $\mathcal{T}(\boldsymbol{s}, \boldsymbol{a})$ is a separately trained dynamics model to ensure physically plausible motion patterns.

Manipulation after Contact Task Guidance. For manipulation after contact tasks such as door opening and tool using, DexHandDiff employs a dual-phase interaction approach that acknowledges the fundamentally different nature of interaction before and after contact establishment. The framework automatically determines the phase transition based on the distance between the palm position and the designated contact point on the object, applying a smooth transition mask to blend between phases.

In the pre-grasp phase, our method focuses on guiding the manipulator to stably align with the contact point while preventing premature object movement. We engineer two primary guidance components: 1) Alignment guidance ϵ_{align} that directs the end-effector towards precise contact point while maintaining natural approaching trajectory; 2) Dynamics consistency guidance ϵ_{dyn}.

Upon establishing contact (determined by palm-object proximity), the post-grasp phase activates additional guidance mechanisms: 1) Goal-directed guidance ϵ_{succ} that steers the coupled hand-object system towards target configurations; 2) Physical constraint guidance $\epsilon_{\text{penalty}}$ that prevents unrealistic state changes (*e.g.*, limiting per-step changes in both door hinge and latch angles); 3) Continued dynamics guidance ϵ_{dyn} to maintain motion feasibility.

Therefore, the guidance energy function follows,

$$\epsilon = \begin{cases} \epsilon_{\text{pre}} = \epsilon_{\text{align}} + \epsilon_{\text{dyn}} & \text{if } |\boldsymbol{s}_{\text{hand}} - \boldsymbol{s}_{\text{contact}}| > \delta_1 \\ \epsilon_{\text{post}} = \epsilon_{\text{succ}} + \epsilon_{\text{dyn}} + \epsilon_{\text{penalty}} & \text{otherwise} \end{cases} \tag{13}$$

where $\boldsymbol{s}_{\text{hand}}$ and $\boldsymbol{s}_{\text{contact}}$ represents the states of dexterous hand and object contact point (*e.g.* door latch, hammer handle *etc.*) respectively, and δ_1 is a small threshold. The separated design of grasp proposal guidance (ϵ_{align}) and task achieving guidance (ϵ_{succ}) mirrors successful policies in prior work [47, 52], effective for dexterous manipulation. Besides, the $\epsilon_{\text{penalty}}$ ensures continuous object state transitions, corresponding to

$$h_{penalty} \triangleq 1 - H(|s_{obj}^{t+1} - s_{obj}^{t}| - \delta_2), \tag{14}$$

where δ_2 is another small threshold and $H(\cdot)$ is the Heaviside step function [49]. Then $\epsilon_{\text{penalty}}$ can be obtained by applying Eq. 11, becoming a Dirac delta function that directly sets value when satisfying the constraints.

In-hand Manipulation Task Guidance. For tasks primarily involving in-hand manipulation (e.g., pen spinning, object reorientation), where objects are typically already in hand or quickly transition to in-hand states, we employ a simplified single-phase guidance structure: 1) Goal state guidance ϵ_{succ} for achieving target object configurations; 2) Active finger motion guidance to ensure realistic object manipulation; 3) Dynamics consistency guidance ϵ_{dyn} to maintain physical plausibility; 4) Physical constraint guidance

$\epsilon_{\text{penalty}}$ that prevents unrealistic state changes.

$$\epsilon = \epsilon_{\text{goal}} + \epsilon_{\text{finger}} + \epsilon_{\text{dyn}} + \epsilon_{\text{penalty}}. \tag{15}$$

Specially, we define the behavior model that encourages active finger involvement as,

$$h_{\text{finger}}(\boldsymbol{\tau}, t) = H(|\boldsymbol{s}^{t+1}_{\text{finger-joints}} - \boldsymbol{s}^{t}_{\text{finger-joints}}| - \delta_3), \tag{16}$$

where $\boldsymbol{s}^{t}_{\text{finger-joints}}$ is the state vector of all finger joints at planning step t. δ_3 is the third small threshold. $H(\cdot)$ is also the Heaviside step function. This specialized handling prevents unrealistic “ghost states”, as discussed in Sec. 4.

5.2. LLM-Based Guidance Generation

The design of task-specific guidance functions for diffusion policies traditionally requires significant manual effort, particularly for diverse dexterous manipulation tasks. To address this challenge, we leverage a **two-stage** Large Language Model (LLM) process for automated guidance generation, adopting text-to-reward paradigm [33, 54].

Overall Pipeline. First, we feed the LLM with a 6-part template (including function purpose, guidance structure, environment description, function prototype, task instruction and few-shot hints) and public documents on simulation environments [38, 41] to generate task-specific prompts. Then, the generated task prompts are queried to another LLM to write guidance function code. Only few-shot hints require specific refinement, reducing human trial-and-error times from about 20 (for hand-craft energy function design) to around 5 while maintaining DexHandDiff performance.

Environment Description. Our approach employs a comprehensive *Pythonic* environment abstraction that captures the complete interaction system. It encapsulates detailed robot joint configurations, and object-environment specifications from public documents, enabling LLM to generate precise guidance functions that account for the full complexity of dexterous manipulation tasks.

Other Details. As previous works [54], once the guidance function code is generated, we execute the code in interpreter. This step may give us valuable feedback, *e.g.*, syntax errors and runtime errors. We utilize the feedback from code execution as a tool for ongoing refinement within the LLM. Besides, our approach uses few-shot hints instead of examples to allow the model to access relevant functions and best practices without direct examples. Each guidance component is normalized over the trajectory horizon to ensure balanced contributions across objectives while preserving their temporal structure. Detailed examples of prompts and generated guidance functions are shown in Appx. E.

6. Experiments

We evaluate our DexHandDiff on five challenging dexterous manipulation tasks with four from Adroit Hand [41] and one from Shadow Hand environment [38]. Both environments feature a 24-joint Shadow Hand simulator with up to 30 degrees of freedom, designed to closely match the hardware setting [44]. Detailed explanations of the five tasks are provided in Appendix B. We use the expert demonstrations collected by teleoperation from D4RL [18] for Adroit tasks (Door, Hammer, Pen and Relocate). However, Shadow Hand environment does not provide demonstration data, so we employ TQC+HER [3, 26] to collect **5000** expert trajectories for the Block Rotate-Z task.

6.1. Performance Comparisons on Goal Adaptability in Interaction-Aware Tasks

We evaluate DexHandDiff in the Door environment to test its goal adaptability across various target angles. Specifically, we require the planners to open the door to 30, 50, 70, 90 and 110 degrees, as well as close the door (reversal task). Note that the training data only includes 90-degree door-opening demonstrations. For some of these tasks, we adjust the environment settings, such as expanding the door's range of motion, to satisfy the evaluation requirements.

We compare DexHandDiff with five baselines: two classifier-guided methods (Diffuser [24] with Goal Inpainting that sets discrete goal states, and Diffuser with Guided Sampling that leverages continuous gradients for fine control), two classifier-free methods (Decision Diffuser [1] and Diffusion Policy [14] that apply diffusion on states and actions respectively), and a variant of DexHandDiff (denoted DexHandDiff-disc.) that uses goal inpainting. To enhance classifier-free methods' learning of goal condition, we use *the difference between the current door angle and target angle* as the condition, rather than a fixed 90° target.

The results are shown in Tab. 2. Classifier-free methods perform well on the 90° task, but their success declines sharply on new target angles, indicating limited adaptability to out-of-distribution targets. Classifier-guided methods demonstrate moderate but consistent performance across goal-adaptive tasks yet their overall success rates remain suboptimal due to imprecise state-action relation modeling in the policy. Our DexHandDiff achieves consistently high success rates across nearly all tasks. The slightly lower performance (90.0%) on the training task (90°) compared to classifier-free methods stems from our additional guidance for adaptation. When ablating this, DexHandDiff achieves $96.7 \pm 4.7\%$ success rate on Open 90°, which is a reasonable trade-off for better generalization. Averaging a 59.2% success rate, over twice that of the next best method (29.5%), DexHandDiff demonstrates robust adaptability across both in-domain and goal-adaptive scenarios.

Besides, we also observe a trend that goals closer to the original training data don't have higher success rates than others. We suppose it's because when target angle is close to training, learned dynamics often override guidance. We observed 8 out of 14 failures in 30 tries in 70° task opened to 90° instead, supporting our hypothesis. This learned bias

Method	Condition	Open 30°	Open 50°	Open 70°	Open 90°	Open 110°	Close Door	Average
Diffuser [24]	Goal Inpainting	16.7 ±4.7	16.7 ±12.5	6.7 ±4.7	56.7 ±9.4	10.0 ±8.2	0	17.8
Diffuser [24]	Guided Sampling	10.0 ±8.2	26.7 ±17.0	10.0 ±4.7	63.3 ±18.7	6.7 ±9.4	**60.0** ±8.2	29.5
Decision Diffuser [1]	Embedding	0	3.3 ±4.7	16.7 ±4.7	**100** ±0	**30.0** ±8.2	0	25.0
Diffusion Policy [14]	Embedding	16.7 ±4.7	3.3 ±4.7	13.3 ±12.5	**100** ±0	3.3 ±4.7	0	22.8
DexHandDiff-disc.	Goal Inpainting	46.7 ±4.7	13.3 ±9.4	**53.3** ±4.7	20.0 ±8.2	6.7 ±4.7	0	23.3
DexHandDiff (Ours)	Guided Sampling	**70.0** ±8.2	**56.7** ±4.7	**53.3** ±8.2	90.0 ±8.2	**26.7** ±14.1	**58.3** ±13.4	**59.2**

Table 2. **Success rates (in %) of different diffusion-based approaches in Adroit Hand [41] environment.** All models were trained on the Open 90° task only, and we test their adaptability to other task goals in Adroit Door environment. All results and standard deviation are calculated over 3 tries for 10 random seeds. Best methods and those within 5% of the best are highlighted in **bold**.

Environment	Task	Diffuser [24] (Inpaint)	Conditional DP [1, 14]	DexHandDiff (Ours)
Door	Open 90°	56.7 ±9.4	**100** ±0	90.0 ±8.2
Door	Open 30°	16.7 ±4.7	16.7 ±4.7	**70.0** ±8.2
Pen	Full Re-orientation	10.0 ±0	80.0 ±8.2	**93.3** ±4.7
Pen	Half-side Re-orientation	3.3 ±4.7	23.3 ±9.4	**40.0** ±8.2
Hammer	Nail Full Drive	53.3 ±9.4	76.7 ±9.4	**90.0** ±8.2
Hammer	Nail Half Drive	23.3 ±12.5	33.3 ±4.7	**46.7** ±12.5
Relocate	Full Relocation	56.7 ±4.7	**96.7** ±4.7	**96.7** ±4.7
Relocate	Half-side Relocation	53.3 ±4.7	86.7 ±12.5	**93.3** ±4.7
Manipulate Block	Rotate-Z	36.7 ±12.5	40.0 ±8.2	**50.0** ±8.2
Manipulate Block	Half-side Rotate-Z	30.0 ±0	26.7 ±4.7	**36.7** ±4.7
Average		34.0	58.0	**70.7**

Table 3. **Overall performance of dexterous manipulation with goal adaptability on multiple environments and tasks.** We compare our method with one classifier-guided baseline and one classifier-free baseline. The results are calculated over 3 tries for 10 random seeds.

is harder to correct than for more distant angles.

6.2. Evaluation on Various Dexterous Tasks

To evaluate the cross-task adaptability and goal-oriented performance of DexHandDiff, we test it across multiple dexterous manipulation tasks in Door, Pen, Hammer, Relocate and Block environments, as summarized in Tab. 3. In addition to the Door task, the Pen task involve aligning a pen to the specified orientation, with a particularly challenging goal-adaptability variant, Half-side Re-orientation, where training data includes only right-hemisphere orientations while test goals require left-hemisphere rotations. Similarly, the Block Rotate-Z and Object Relocation tasks have block's half-side variant trained on positive goal yaw angles but tested on negative ones and object's target right-half table training but left-half testing. The Nail Half Drive requires the hand to drive a nail and stop halfway before retracting, testing control precision for partial goals.

We compare DexHandDiff with two baselines: Diffuser [24] (Inpainting), using classifier-guided goal inpainting as in the previous section, and Conditional DP [1, 14], a classifier-free approach with state diffusion for Door, Hammer and Relocate tasks while action diffusion for Pen and Block tasks, as modeling dynamics for these tasks are particularly challenging, making direct action generation more effective than state-based diffusion. As shown in Tab. 3, DexHandDiff consistently achieves superior results across both in-domain and goal-adaptive tasks. Although conditional DP demonstrates 23.3% on the challenging pen half-side re-orientation, leveraging the inherent multi-modality and anisotropy of diffusion models, DexHandDiff still performs better (40.0%). These results underscore DexHandDiff's robustness and adaptability across a range of tasks, demonstrating generalization on familiar goals and novel configuration challenges.

Adapt Tasks	Door 30°	Door 70°	Pen Half	Hammer Half	Relocate Half
Diffuser [24]	4.19	4.03	5.23	4.01	5.48
DexHandDiff	**2.92**	**2.38**	**2.76**	**2.41**	**3.22**

Table 4. **Quantitative results for preventing ghost states over 3 tries.** (Conditional DP is not included due to its action-only.)

6.3. Validation for Preventing Ghost States

We measured *L2 distance* between predicted and simulated hand-object states (normalized per dimension for fair comparison) in Tab. 4. DexHandDiff nearly halves baseline's gap across tasks, illustrating its ghost-state reduction effect.

6.4. Ablation on LLM-based Guidance Generation

Table 5 presents results for different guidance methods on goal adaptability tasks. All three methods are based on the same joint state-action diffusion model. The Human Craft approach reflects our above results with manually de-

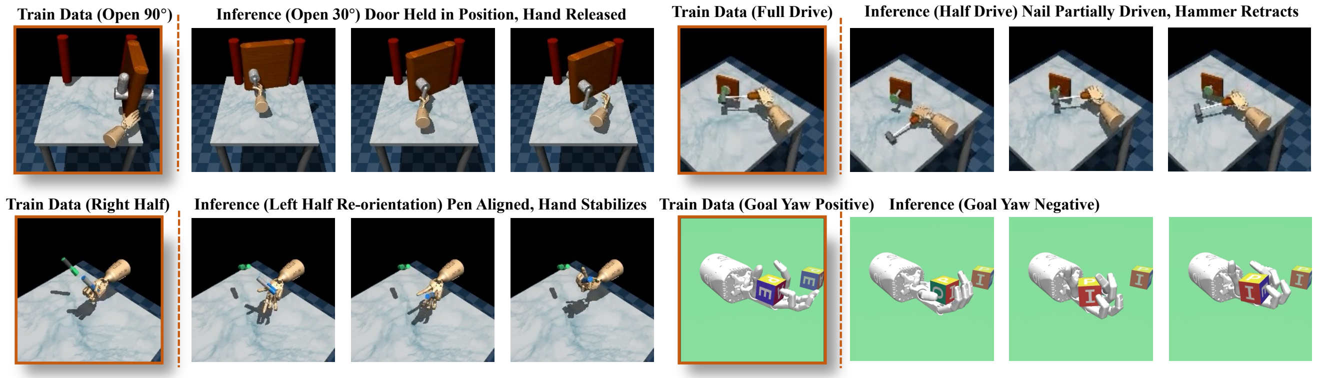

Train Data (Open 90°)
Inference (Open 30°) Door Held in Position, Hand Released
Train Data (Full Drive)
Inference (Half Drive) Nail Partially Driven, Hammer Retracts
Train Data (Right Half)
Inference (Left Half Re-orientation) Pen Aligned, Hand Stabilizes
Train Data (Goal Yaw Positive)
Inference (Goal Yaw Negative)

Figure 4. **Visualization results of goal-adaptive tasks by DexHandDiff.** For each task, training data sample (with orange stroke) is followed by inference on novel goals beyond the training data. In the Door task, DexHandDiff guides the door to new target angle (30°) and holds the door in position when the hand releases, *which cannot be attained by simply truncating actions from 90° training data*. DexHandDiff avoids ghost states and achieves better goal adaptability.

Task	Naïve Guide	Human Craft	LLM Gen
Door Open 30°	0	70.0 ±8.2	40.0 ±8.2
Pen Half-side Re-orien	20.0 ±8.2	40.0 ±8.2	26.7 ±4.7
Hammer Half Nail	20.0 ±8.2	46.7 ±12.5	43.3 ±9.4

Table 5. **Ablation study on LLM-based guidance generation.**

signed guidance. LLM Gen generate guidance functions with Claude Sonnet 3.5 [5]. And Naïve Guide directly guides the object to the goal, corresponding to ghost-state existing baseline. Results indicate that both Human Craft and LLM Gen significantly outperform Naïve Guide across tasks, with Human Craft achieving the highest success rates.

6.5. Ablation Study of DexHandDiff Framework

We analyze the contribution of each component in DexHandDiff through ablation studies (Tab. 6), across multiple door-opening tasks (open 30°, 50°, 70°, and 90°), using the same training checkpoint for fair comparison. The baseline Diffuser[24] uses a basic goal-guidance strategy, while Dyn-guide enhances it with dynamics guidance for better state-action consistency. Joint S&A adopts joint state-action denoising like DexHandDiff but retains naive goal guidance. DexHandDiff incorporates all components and achieves the highest success rate of 67.5%, significantly outperforming the other configurations and demonstrating the effectiveness of our full design.

6.6. Visualizations

We visualize the behavior of DexHandDiff across various goal-adaptive dexterous tasks in Fig. 4. DexHandDiff ensures realistic contact by aligning hands with contact points first using joint dynamics modeling, eliminating ghost states. Notably, for example, DexHandDiff guides the door to new target angle and holds the door steady when the hand releases , which cannot be achieved by policies trained with slicing 90° data. These results underscore DexHandDiff's ability to maintain physically realistic interactions while adapting to novel goals.

Method	Goal Guidance	Dynamics Guide	Joint State Action	Interact Mechanism	Overall SR
No-guide	×	×	×	×	24.1
Diffuser [24]	✓	×	×	×	27.5
Dyn-guide	✓	✓	×	×	27.5
Joint S&A	✓	×	✓	×	30.8
Dyn+Joint	✓	✓	✓	×	31.7
DexHandDiff	✓	✓	✓	✓	**67.5**

Table 6. **Ablation study on DexHandDiff framework.** We report the average success rates (overall SR) on Adroit Door environment over open 30°, 50°, 70° and 90° tasks.

6.7. Efficiency

We test the control frequency of DexHandDiff on an RTX 3090 with receding horizon set as 8 for all tasks except Door (32 instead). The control frequency are reported below.

Task	Door	Pen	Hammer	Relocate	Block
Freq.	5.04 Hz	5.88 Hz	5.86 Hz	5.78 Hz	6.92 Hz

Table 7. **Control command frequency over 10 tries.**

Besides, our lightweight model (3.96M params, 3.27 GFLOPS) can be further accelerated via DPM Solver++ [31] (4x speedup) and command interpolation (reaching 36 Hz), sufficient for real robot control.

7. Conclusion

This work presents DexHandDiff, an interaction-aware diffusion planner for adaptive dexterous manipulation. By modeling joint state-action dynamics and incorporating a dual-phase diffusion mechanism, it addresses action-state consistency issues, including the "ghost state" and generalization problems observed in previous diffusion methods. DexHandDiff's design enables it to handle intricate multi-contact interactions through a pre-contact alignment and a post-contact control. We believe its potential to advance the field toward diverse dexterous tasks while remaining grounded in real physics and dynamics.

Future Work can investigate deployment with hand states sensed and object poses estimated by vision models.

Acknowledgements

This paper is partially supported by the General Research Fund of Hong Kong No.17200622 and 17209324.

References

[1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B Tenenbaum, Tommi S Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 3, 4, 6, 7

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019. 1

[3] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017. 6

[4] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 1

[5] Anthropic. Claude 3.5 sonnet, 2024. Available at: https://www.anthropic.com/news/claude-3-5-sonnet. 8

[6] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 2

[7] Junting Chen, Yao Mu, Qiaojun Yu, Tianming Wei, Silang Wu, Zhecheng Yuan, Zhixuan Liang, Chao Yang, Kaipeng Zhang, Wenqi Shao, et al. Roboscript: Code generation for free-form manipulation tasks across real and simulation. *arXiv preprint arXiv:2402.14623*, 2024. 3

[8] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pages 297–307. PMLR, 2022. 1

[9] Tianxing Chen, Yao Mu, Zhixuan Liang, Zanxin Chen, Shijia Peng, Qiangyu Chen, Mingkun Xu, Ruizhen Hu, Hongyuan Zhang, Xuelong Li, et al. G3flow: Generative 3d semantic flow for pose-aware and generalizable object manipulation. *arXiv preprint arXiv:2411.18369*, 2024. 2

[10] Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Zongqing Lu, Stephen McAleer, Hao Dong, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5150–5163, 2022. 2

[11] Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[12] Zerui Chen, Shizhe Chen, Cordelia Schmid, and Ivan Laptev. Vividex: Learning vision-based dexterous manipulation from human videos. *arXiv preprint arXiv:2404.15709*, 2024. 2

[13] Zoey Qiuyu Chen, Karl Van Wyk, Yu-Wei Chao, Wei Yang, Arsalan Mousavian, Abhishek Gupta, and Dieter Fox. Dextransfer: Real world multi-fingered dexterous grasping with minimal human demonstrations. *arXiv preprint arXiv:2209.14284*, 2022. 2

[14] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 1, 2, 3, 4, 6, 7

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3, 1

[16] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[17] William Feller. On the theory of stochastic processes, with particular reference to applications. In *Selected Papers I*, pages 769–798. Springer, 2015. 3

[18] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. 6

[19] Abhishek Gupta, Justin Yu, Tony Z Zhao, Vikash Kumar, Aaron Rovinsky, Kelvin Xu, Thomas Devlin, and Sergey Levine. Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6664–6671. IEEE, 2021. 2

[20] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024. 2

[21] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8): 1771–1800, 2002. 4

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[23] Wenlong Huang, Igor Mordatch, Pieter Abbeel, and Deepak Pathak. Generalization in dexterous manipulation via geometry-aware multi-task learning. *arXiv preprint arXiv:2111.03062*, 2021. 1

[24] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pages 9902–9915. PMLR, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 3

[26] Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, pages 5556–5566. PMLR, 2020. 6

[27] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023. 3

[28] Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. In *International Conference on Machine Learning*, pages 20725–20745. PMLR, 2023. 1, 2

[29] Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16467–16476, 2024.

[30] Zhixuan Liang, Yao Mu, Yixiao Wang, Fei Ni, Tianxing Chen, Wenqi Shao, Wei Zhan, Masayoshi Tomizuka, Ping Luo, and Mingyu Ding. Dexdiffuser: Interaction-aware diffusion planning for adaptive dexterous manipulation. *arXiv preprint arXiv:2411.18562*, 2024. 2

[31] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 8

[32] Zhengyi Luo, Jinkun Cao, Sammy Christen, Alexander Winkler, Kris Kitani, and Weipeng Xu. Grasping diverse objects with simulated humanoids. *arXiv preprint arXiv:2407.11385*, 2024. 2

[33] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023. 3, 6

[34] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning*, pages 651–661. PMLR, 2022. 1

[35] Yao Mu, Junting Chen, Qing-Long Zhang, Shoufa Chen, Qiaojun Yu, GE Chongjian, Runjian Chen, Zhixuan Liang, Mengkang Hu, Chaofan Tao, et al. Robocodex: Multimodal code generation for robotic behavior synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 3

[36] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020. 2

[37] Fei Ni, Jianye Hao, Yao Mu, Yifu Yuan, Yan Zheng, Bin Wang, and Zhixuan Liang. Metadiffuser: Diffusion model as conditional planner for offline meta-rl. In *International Conference on Machine Learning*, pages 26087–26105. PMLR, 2023. 2

[38] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research, 2018. 6

[39] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994. 3

[40] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022. 2

[41] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 2, 6, 7

[42] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 2

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[44] Shadow Robot Company. Shadow robot. https://www.shadowrobot.com/, 2024. Accessed: 2024-11-14. 6

[45] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. *arXiv preprint arXiv:2202.10448*, 2022. 2

[46] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012. 4

[47] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3891–3902, 2023. 1, 5

[48] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024. 2

[49] Eric W Weisstein. Heaviside step function. *https://mathworld. wolfram. com/*, 2002. 5

[50] Zehang Weng, Haofei Lu, Danica Kragic, and Jens Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models. *arXiv preprint arXiv:2402.02989*, 2024. 2, 1

[51] Chengyue Wu, Yixiao Ge, Qiushan Guo, Jiahao Wang, Zhixuan Liang, Zeyu Lu, Ying Shan, and Ping Luo. Plot2code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots. *arXiv preprint arXiv:2405.07990*, 2024. 3

[52] Tianhao Wu, Yunchong Gan, Mingdong Wu, Jingbo Cheng, Yaodong Yang, Yixin Zhu, and Hao Dong. Unidexfpm: Universal dexterous functional pre-grasp manipulation via diffusion policy. *arXiv preprint arXiv:2403.12421*, 2024. 1, 2, 5

[53] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3

[54] Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 6

[55] Chunmiao Yu and Peng Wang. Dexterous manipulation for multi-fingered robotic hands with reinforcement learning: A review. *Frontiers in Neurorobotics*, 16:861825, 2022. 2

[56] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 2

[57] Bohan Zhou, Haoqi Yuan, Yuhui Fu, and Zongqing Lu. Learning diverse bimanual dexterous manipulation skills from human demonstrations. *arXiv preprint arXiv:2410.02477*, 2024. 2

[58] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019. 2