Learning Diffusion Policy from Primitive Skills for Robot Manipulation

Zhihao Gu[1*], **Ming Yang**[2], **Difan Zou**[1], **Dong Xu**[1†]

[1]Department of Computer Science, School of Computing and Data Science, The University of Hong Kong
[2]School of Software, Beihang University
zhihao.gu@ntu.edu.sg, viv@buaa.edu.cn, {dzou, dongxu}@cs.hku.hk

Abstract

Diffusion policies (DP) have recently shown great promise for generating actions in robotic manipulation. However, existing approaches often rely on *global* instructions to produce *short-term* control signals, which can result in misalignment in action generation. We conjecture that the primitive skills, referred to as fine-grained, short-horizon manipulations, such as "move up" and "open the gripper", provide a more intuitive and effective interface for robot learning. To bridge this gap, we propose SDP, a skill-conditioned DP that integrates interpretable skill learning with conditional action planning. SDP abstracts eight reusable primitive skills across tasks and employs a vision-language model to extract discrete representations from visual observations and language instructions. Based on them, a lightweight router network is designed to assign a desired primitive skill for each state, which helps construct a single-skill policy to generate skill-aligned actions. By decomposing complex tasks into a sequence of primitive skills and selecting a single-skill policy, SDP ensures skill-consistent behavior across diverse tasks. Extensive experiments on two challenging simulation benchmarks and real-world robot deployments demonstrate that SDP consistently outperforms SOTA methods, providing a new paradigm for skill-based robot learning with diffusion policies.

Introduction

Enabling robots to perform diverse real-world tasks has been a long-standing goal in robotics and artificial intelligence. One promising approach is to teach robots by example, allowing them to learn directly from demonstrations. However, it is uniquely challenging: unlike standard prediction problems, robotic control demands precise, context-aware actions (Chi et al. 2025). To handle it, prior research focused on improved action representations (Mandlekar et al. 2021; Shafiullah et al. 2022) and richer internal models of robot behavior (Florence et al. 2022; Wu et al. 2020).

Recently, diffusion models (Ho, Jain, and Abbeel 2020; Mokady et al. 2023), a class of generative models that learn to reverse a gradual noise-adding process, have achieved remarkable success in high-fidelity image generation (Ho, Jain, and Abbeel 2020; Jo, Lee, and Hwang 2022; Rombach et al. 2022). Building on this success, diffusion models have been explored in robotics for generating action sequences. The Diffusion Policy (DP) (Chi et al. 2025) is a pioneering work that generates robot behavior via the conditional denoising process of diffusion models. Instead of directly outputting an action, it infers the action-score gradient for several denoising iterations, significantly improving the performance. However, conditioned only on visual observations, it is difficult to learn multiple tasks at the same time, severely limiting its deployment in real-world scenarios.

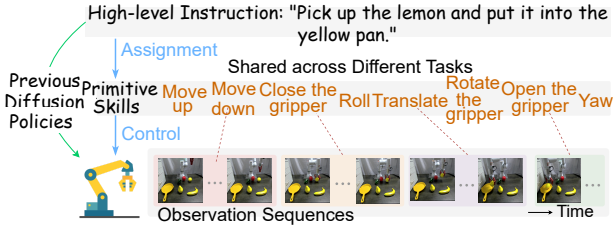
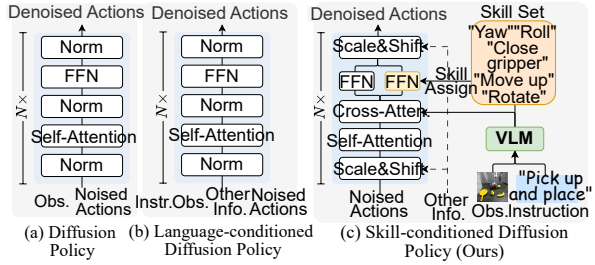


Figure 1: A task consists of a series of short-term manipulations, and we abstract them into eight shared primitive skills, which provide concrete instruction. Previous diffusion policies map high-level instructions to actions directly. In contrast, we learn primitive skills and integrate them into the conditional action generation for more precise control.

To overcome these limitations, recent work has introduced natural language instructions as a condition, enabling robots to perform a broader range of tasks (Ha, Florence, and Song 2023; Reuss et al. 2024b; Liu et al. 2024; Wang et al. 2024b; Reuss et al. 2024a). Typically, a language encoder transforms instructions into embeddings, which, together with noisy action sequences, are sent to the diffusion policy for action generation. Research in this paradigm has advanced along three dimensions (Song et al. 2025): robot data representations (Wang et al. 2024b; Ze et al. 2024; Wang et al. 2024a), model architectures (Team et al. 2024; Reuss et al. 2024b; Ye et al. 2024), and diffusion strategies (Ren et al. 2024; Reuss et al. 2024b; Liang et al. 2024). Despite these advances, most existing methods map high-level instructions directly to short-term actions, which can result in ambiguous or misaligned behaviors. For example, if the robot is going to close the gripper, the high-level task de-

*Work done when Zhihao Gu was a Postdoc of Prof. Dong Xu.
†Corresponding author.



(a) Diffusion Policy (b) Language-conditioned Diffusion Policy (c) Skill-conditioned Diffusion Policy (Ours)

Figure 2: Comparison between (a) diffusion policy (DP), (b) language-conditioned DP, and (c) our skill-conditioned DP. Ours executes abstract instructions with more precise guidance from the assigned primitive skills.

scription, "Pick up the lemon and put it into the pan", would be too abstract to provide an explicit instruction, while a more fine-grained level of instruction should be included. This motivates us to design a method that can *generate concrete short-term instructions (named primitive skills)*, such as "close the gripper", and *learn single-skill diffusion policies* to generate more accurate actions.

To address this gap, we propose SDP, a skill-conditioned diffusion policy that combines fine-grained skill learning with conditional, low-level action generation. Our approach is built on two key ideas: (1) decomposing ambiguous, high-level instructions into learnable short-term skills based on current observations, and (2) training a diffusion policy that generates actions conditioned on these skills. Specifically, we first abstract short-term manipulations across various tasks into eight primitive skills (see Figure 1), which in turn can be composed to form complex tasks, and convert visual observations and high-level instructions into discrete representations by a vision-language model. A lightweight router network then dynamically assigns the appropriate skill for each state. The assigned skill synthesizes parameters of the feed-forward network (FFN) in diffusion policy, while additional information, such as proprioception, is injected using an AdaLN operation. The resulting single-skill diffusion policy is capable of producing coherent and precise behaviors aligned with the skill. Compared to previous approaches (see Figure 2 (a) and (b)), our SDP interprets and executes complex instructions end-to-end with greater accuracy and more precise control. Extensive experiments on two challenging simulation benchmarks and real-world robot deployments demonstrate the superior performance of SDP. In summary, our main contributions are as follows:

- We present SDP, a skill-conditioned diffusion policy that combines fine-grained skill learning and low-level action generation, to mitigate the misalignment in granularity between global instruction and short-term actions.
- We introduce eight reusable primitive skills that generalize across diverse manipulation tasks, providing a structured and interpretable action space for robot learning. Furthermore, to leverage these skills effectively, we design a lightweight router network that dynamically assesses state relevance and selects the optimal skill, ensuring adaptive and task-aligned behavior generation.
- We design a novel single-skill diffusion policy that generates actions precisely aligned with each skill. By dynamically parameterizing the policy's FFN layer from the assigned skill, we can effectively capture the dependency between primitive skills and low-level control signals.
- We demonstrate better multi-task and generalization capabilities than baselines across simulated and real-world tasks. The visualization of skills also reveals its ability to decompose abstract instructions and compose primitive skills, validating its effectiveness and interpretability.

Related Work

Diffusion policy in robot Manipulation. Diffusion models (Ho, Jain, and Abbeel 2020; Mokady et al. 2023) have recently achieved remarkable success in a variety of fields, and their potential for robotic manipulation has attracted growing interest. In the robotics community, researchers have focused on developing diffusion-based policies that enable robots to follow language instructions and perform diverse tasks (Ha, Florence, and Song 2023; Reuss et al. 2024b; Liu et al. 2024; Wang et al. 2024b; Reuss et al. 2024a). These efforts span three main areas: robot data representations, model architectures, and diffusion strategies. The data representations include 2D trajectories (Wang et al. 2024b), 3D point clouds (Ze et al. 2024), and combinations of sensory inputs (Wang et al. 2024a). Model architectures often combine diffusion models with large language models (Team et al. 2024), transformers (Reuss et al. 2024b), or variational autoencoders (Ye et al. 2024). Additionally, different training strategies have been explored, such as integrating reinforcement learning (Ren et al. 2024), self-supervised learning (Reuss et al. 2024b), and classifier guidance (Liang et al. 2024; Mete et al. 2024). In contrast, our SDP emphasizes learning executable skills and training a skill-conditioned diffusion policy for more precise and coherent robot control.

Planning by VLM. Decomposing complex instructions into manageable sub-goals helps robots complete sophisticated tasks more reliably (Dalal, Pathak, and Salakhutdinov 2021; Dhakan et al. 2022; Hiranaka et al. 2023; Liu et al. 2025). However, manually annotating these sub-goals is labor-intensive and does not scale well. To overcome this, recent methods (Singh et al. 2022; Zhang et al. 2023; Ni et al. 2024) leverage VLMs rich in real-world knowledge to automatically generate task plans for robot learning. Other methods like (Garg et al. 2022), learn codebooks of sub-goals from latent variables, where each code may correspond to multiple states. Our SDP differs in two key aspects. First, we propose a set of human-understandable primitive skills that can be flexibly combined to complete a wide range of tasks. Second, unlike prior work that models these skills *implicitly*, we *explicitly* assign a skill to each state using a lightweight neural network guided by VLM outputs, leading to more transparent and controllable behavior.

Parameter synthesis. Hypernetworks (Ha, Dai, and Le 2016) are neural networks designed to generate the parameters of other networks, using context information as input. This approach provides an efficient way to model the depen-

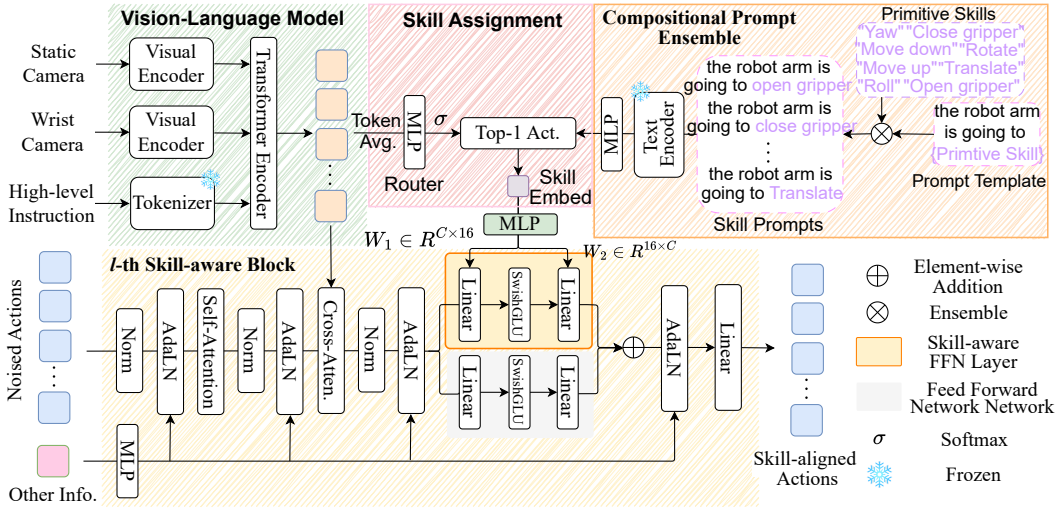


Figure 3: Overview of the proposed skill-conditioned diffusion policy (SDP). SDP abstracts short-term manipulations across different tasks into eight primitive skills and introduces a unified prompt template to specify the upcoming manipulations. A router is then designed to assign importance scores for all candidate skills based on the embedding $\boldsymbol{z}_{vl}$, generated from visual observations and language instructions by a VLM. Furthermore, a skill with the highest score is selected, which parameterizes an additional FFN layer in the diffusion policy. Other information, such as proprioception, is encoded by an MLP and further injected via an AdaLN operation, resulting in a single-skill policy that predicts skill-aligned actions for precise control.

dency between the task and the optimal control policy (Ren et al. 2025). For instance, HyperDistill (Xiong et al. 2024) uses a hypernetwork to learn policies for robots with different physical structures, achieving strong performance with minimal computational cost. Inspired by it, SDP establishes the dependency between skills and action predictions by parameterizing FFN layers in the diffusion policy.

Preliminaries

The robotic manipulation learns a general policy that performs diverse tasks. Assume we have a set of robotic demonstrations $\mathcal{T} = \{\boldsymbol{\tau}_i\}_{i=1}^{|\mathcal{T}|}$, where each trajectory $\boldsymbol{\tau}_i = \{(\boldsymbol{s}_n, \overline{\boldsymbol{a}}_{n,k}, \boldsymbol{l}_i)\}_{n=1}^{N}$ contains the state $\boldsymbol{s}_n \in \mathbb{R}^{d_s}$, the action sequence $\overline{\boldsymbol{a}}_{n,k} \in \mathbb{R}^{7\times k}$ of length k starting at timestep n and the high-level language instruction $\boldsymbol{l}_i$ specifying the task. The language-conditioned policy aims to train a policy $\pi_\theta(\overline{\boldsymbol{a}}|\boldsymbol{s}, \boldsymbol{l}) : (\boldsymbol{s}_n, \boldsymbol{l}_i) \mapsto \overline{\boldsymbol{a}}_{n,k}$ that maps state $\boldsymbol{s}_n$ at timestep n and the instruction $\boldsymbol{l}_i$ to a sequence of future actions.

Language-conditioned diffusion policy leverages the diffusion model to obtain the policy $\pi_\theta(\overline{\boldsymbol{a}}|\boldsymbol{s}, \boldsymbol{l})$. To generate new samples from noise, based on historical state embeddings $\bar{\boldsymbol{s}}$ and the instructions $\boldsymbol{l}$, it trains a neural network D_θ to approximate the score function of the diffusion process by Denoising Score Matching (Vincent 2011):

$$\mathcal{L}_{\text{SM}}(\theta; \boldsymbol{s}, \boldsymbol{l}) = \mathbb{E}_{\sigma, \bar{\boldsymbol{a}}, \boldsymbol{\epsilon}} \left[\frac{1}{\sigma_t} \left\| D_\theta(\bar{\boldsymbol{a}} + \boldsymbol{\epsilon}, \bar{\boldsymbol{s}}, \boldsymbol{l}, \sigma_t) - \bar{\boldsymbol{a}} \right\|_2^2 \right], \quad (1)$$

where $\boldsymbol{\epsilon}$ is the noise and σ_t is the density at step t. The diffusion model is trained by minimizing the average loss over state-action-instruction tuples from $\mathcal{T}$. Once D_θ is trained, the DDIM (Zhang, Tao, and Chen 2022) is adopted to sample the desired actions within N_d denoising steps. We refer readers to MoDE (Reuss et al. 2024a) for more details.

The Equation (1) directly maps the global instruction $\boldsymbol{l}$ to the local actions. We argue that the task specification is too abstract to provide an explicit instruction that guides the diffusion policy to generate precise short-term actions.

Proposed Approach

Approach Overview

This paper proposes the skill-conditioned diffusion policy to address the issue of imprecise executions from high-level instructions. The diagram is illustrated in Figure 3: the upper part predicts a primitive skill that describes the upcoming manipulations, and the lower part provides a single-skill policy that integrates the state information and generates skill-aligned actions. Notably, we use the vision-language representations to assign skills, thereby rendering the execution of tasks both interpretable and comprehensible to humans.

Primitive Skill Assignment

To perform a task based on a language instruction, a policy predicts short-term actions for each state. However, the coarse granularity of the high-level instruction may introduce ambiguity into fine-grained action generation. Instead, we propose to decompose tasks into fundamental, irreducible manipulation primitives, called *primitive skills*. These skills provide precise, actionable guidance for generating accurate short-term controls.

Compositional prompt ensemble (CPE). Following the intuition, we learn such primitive skills explicitly. Specifically, we abstract basic manipulations into eight reusable primitive skills, denoted as P, *i.e.,* "roll", "yaw", "open

the gripper", "move up", "translate", "close the gripper", "move down", and "rotate". To better describe the state of the robot, we specially design a unified text template *"the robot arm is going to {skill}."*. Inspired by the prompt ensemble in CLIP (Radford et al. 2021), we further propose the Compositional Prompt Ensemble to generate prompts for each skill, formulated as follows:

$$P_{En} := \text{"the robot arm is going to } \{skill\}\text{."} \otimes P \quad (2)$$

where $\otimes$ denotes the ensemble operation. After that, a frozen CLIP text encoder $\mathrm{CLIP}_{\mathrm{text}}(\cdot)$, followed by a MLP f, encodes the ensemble P_{En} into the prompt embedding $\boldsymbol{p} = f(\mathrm{CLIP}_{\mathrm{text}}(P_{En})) \in \mathbb{R}^{8\times C_{\mathrm{img}}}$, where C_{img} is the dimension of joint space for the skill assignment. Finally, one of $\{\boldsymbol{p}_i\}_{i=1}^{8}$ will be selected to guide the action generation.

Note that almost all tasks can be composed of those primitive skills, and the text template provides the general prompt for the robot's state. Thus, texts from CPE are reusable, and we pre-compute and store them for efficiency in inference.

Vision-language model. Since CPE provides concrete prompts for each skill, we need to identify which skill will be performed. In particular, we utilize vision-language representations from visual observations and the high-level instruction as guidance for the assignment. Formally, let $\boldsymbol{I}_s, \boldsymbol{I}_w \in \mathbb{R}^{3\times H\times W}$ be the visual observations from the static and wrist cameras, respectively. They are encoded into visual embedding $f_{\mathrm{img}}(\boldsymbol{I}_s)$ and $f_{\mathrm{img}}(\boldsymbol{I}_w)$ by a shared image encoder $f_{\mathrm{img}}(\cdot) : \mathbb{R}^{3\times H\times W} \to \mathbb{R}^{N_{\mathrm{img}}\times C_{\mathrm{img}}}$, where N_{img} and C_{img} represent the number and dimensionality of vision tokens, respectively. At the same time, the high-level instruction $\boldsymbol{l}$ is processed by the tokenizer and word embedding layer from (Xiao et al. 2024), resulting in the text embeddings $f_{\mathrm{t}}(\boldsymbol{l}) \in \mathbb{R}^{N_t\times C_{\mathrm{text}}}$. Then vision and text tokens are concatenated and sent to a transformer Φ to obtain the vision-language representations $\boldsymbol{z}_{vl} = \Phi([f_{\mathrm{t}}(\boldsymbol{l}), f_{\mathrm{img}}(\boldsymbol{I}_s), f_{\mathrm{img}}(\boldsymbol{I}_w)]) \in \mathbb{R}^{(N_t+2N_{\mathrm{img}})\times C_{\mathrm{img}}}$. We omit the extra projection on $f_{\mathrm{t}}(\boldsymbol{l})$ for dimension alignment.

Primitive skill selection. Based on the embedding of skill prompts $\boldsymbol{p}$ and the vision-language representations $\boldsymbol{z}_{vl}$, we further devise the skill assignment module that employs a lightweight router network to select a skill for each state. In detail, we first average the token dimension of $\boldsymbol{z}_{vl}$ and obtian the variable $\boldsymbol{z}_{\mathrm{avg}} \in \mathbb{R}^{C_{\mathrm{img}}}$. Then an MLP layer maps $\boldsymbol{z}_{\mathrm{avg}}$ into a logits to reflect the importance of each skill, followed by a $\mathrm{Softmax}$ function $\sigma(\cdot)$ and the $\text{top-1}(\cdot)$ operation to narrow down all skills to the most suitable one:

$$R(\boldsymbol{z}_{\boldsymbol{vl}}) = \text{top-1}(\sigma(\mathrm{MLP}(\mathrm{Avg}(\boldsymbol{z}_{vl})))), \quad (3)$$

A skill with the highest score is subsequently selected based on its importance in $R(\boldsymbol{z}_{vl}) \in \mathbb{R}^8$. Finally, the skill embedding for each state is selected by $\boldsymbol{z} = \sum_{i=1}^{8} R(\boldsymbol{z}_{vl})_i \cdot \boldsymbol{p}_i$.

Analysis. The VQ (Van Den Oord, Vinyals et al. 2017) is utilized to *implicitly* learn discrete latent codes (Garg et al. 2022; Liang et al. 2024). Differently, our SDP *explicitly* abstracts shared primitive skills across varying tasks, and the skill assignment is more human-understandable. More recently, GSC (Mishra et al. 2023) explicitly parameterizes skills, pedicted by a DP. In contrary, our (finer grained) primitive skills (PS) can be assembled into their skills and express broader tasks. Moreover, ours are learned in a unified model rather than separate ones, which is more efficient.

Skill-conditioned Diffusion Policy Learning

The ultimate goal is to predict skill-aligned actions. We propose to learn single-skill diffusion policies. We first inject state priors and then build a dependency between the assigned primitive skill and the conditional action generation.

Priors injection. For each state, time steps, proprioception, visual observations, and high-level instruction are provided. Following work (Doshi et al. 2024), a small MLP-based encoder is used to handle time steps and proprioception. Later, they are injected by a modified AdaLN (Perez et al. 2017) that generates distinct modulation signals shared across all layers. Instead, for the visual and linguistic information, the output tokens of the VLM are first projected via a linear layer with RMSNorm (Zhang and Sennrich 2019) and then injected by a Cross-Attention in each block, as shown in Figure 3. These operations integrate state priors and achieve conditional injection. Compared to the standard AdaLN that assigns unique parameters to each layer, ours reduces learnable parameters while maintaining performance.

Skill-dependent FFN layer. To build a dependency between the primitive skill and the action generation, we additionally introduce a LoRA-like (Hu et al. 2022) feed-forward (FFN) layer to the original $\mathrm{FFN}_{\mathrm{ori}}$. The new FFN contains a SwishGLU activation and two matrices $\boldsymbol{W}_{\boldsymbol{z}}^1 \in \mathbb{R}^{C\times 16}$ and $\boldsymbol{W}_{\boldsymbol{z}}^2 \in \mathbb{R}^{16\times C}$, generated from the skill embed $\boldsymbol{z}$ by an MLP and mapping an input $\boldsymbol{x}$ from dimension of C to 16 and 16 to C, respectively. The final FFN layer is thus formulated as:

$$\mathrm{FFN}(\boldsymbol{x}) = \boldsymbol{W}_{\boldsymbol{z}}^2(\mathrm{SwishGLU}(\boldsymbol{W}_{\boldsymbol{z}}^1\boldsymbol{x})) + \mathrm{FFN}_{\mathrm{ori}}(\boldsymbol{x}), \quad (4)$$

The LoRA-like FFN explicitly considers skill in feature extraction that saves memory and reduces the total parameters.

Training Objective. We additionally adopt an orthogonal loss $\mathcal{L}_{\mathrm{Orth}}(\theta)$ to reduce pairwise cosine similarity on $\boldsymbol{p}_{i,j}$ with a hyperparameter γ. The loss function thus becomes:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{SM}}(\theta) + \gamma\mathcal{L}_{\mathrm{Orth}}(\theta), \quad (5)$$

where $\mathcal{L}_{\mathrm{Orth}} = \frac{1}{64}\sum_{i=1}^{8}\sum_{j=1}^{8}\mathrm{Cos}(\boldsymbol{p}_i, \boldsymbol{p}_j)$ and $\gamma = 0.01$.

Analysis. Equation (4) can be viewed as a variant of the mixture of experts (Jacobs et al. 1991), where the first term is designed as a skill-dependent expert but the second one is a shared expert. Consequently, it constructs a single-skill diffusion policy that predicts skill-aligned actions. We call it a skill-conditioned diffusion policy in this paper.

Experiments

This section describes details of benchmarks and implementation. Comprehensive evaluations are conducted to study:

- **Performance.** Can our SDP deliver strong performance compared to SOTA competitors across various settings?
- **Effectiveness**. How do the proposed design choices of our architecture impact final performance?
- **Interpretability.** How does SDP complete various tasks?

Train→Test	Method	No. Instructions in a Row (1000 chains)					
		1	2	3	4	5	**Average Length**
ABCD→D	DiffPolicy	86.3%	72.7%	60.1%	51.2%	41.7%	3.16±0.06
	RoboFlamingo	96.4%	89.6%	82.4%	74.0%	66.0%	4.09±0.00
	GR-1	94.9%	89.6%	84.4%	78.9%	73.1%	4.21±0.00
	MDT	98.6%	95.8%	91.6%	86.2%	80.1%	4.52±0.02
	MoDE†	95.4%	89.1%	83.8%	78.5%	73.4%	4.19±0.03
	SDP (Ours)	**99.7%**	**96.7%**	**93.8%**	**90.8%**	**86.5%**	**4.67±0.02**
ABC→D	RT-1	53.3%	22.2%	9.4%	3.8%	1.3%	0.90±0.06
	DiffPolicy	63.5%	35.3%	19.4%	10.7%	6.4%	1.35±0.05
	RoboFlamingo	82.4%	61.9%	46.6%	33.1%	23.5%	2.47±0.00
	OpenVLA†	91.3%	77.8%	62.0%	52.1%	43.5%	3.27±0.00
	GR-1	85.4%	71.2%	59.6%	49.7%	40.1%	3.06±0.00
	UniVLA	95.5%	85.8%	75.4%	66.9%	56.5%	3.80±0.07
	SkillDiffuser	94.4%	82.7%	72.1%	62.4%	55.4%	3.66±0.07
	MoDE†	95.8%	88.1%	79.3%	70.8%	62.4%	3.92±0.05
	SDP (Ours)	**99.3%**	**96.1%**	**90.9%**	**85.3%**	**76.9%**	**4.49±0.04**

Table 1: Performance on the CALVIN. Success rates for each task and average rollout length to complete 5 consecutive instructions are reported. ± 0.00 indicates methods without average performance and † means re-implementation using official codes.

Experiment Setup and Implementation Details

Simulated benchmarks. We evaluate the proposed SDP on the **CALVIN** (Mees et al. 2022) and **LIBERO** (Liu et al. 2023) benchmarks. The **CALVIN** consists of four distinct scene configurations (splits A-D), with 34 distinct tasks of 24,000 language-annotated demonstrations. In our study, we adopt the challenging evaluation setting of ABC→D, wherein policies are trained using demonstrations from environments A, B, and C, and zero-shot evaluated in environment D, and ABCD→D. The evaluation protocol comprises a test set of 1,000 unique instruction chains, each consisting of five consecutive tasks. The performance is measured by success rates on sequences of 1-5 consecutive tasks and the average length of completed task sequences. The **LIBERO** (Liu et al. 2023) comprises multiple task suites reflecting different aspects of robotic manipulation. Our experiments focus on supervised fine-tuning within the target suite, including LIBERO-Spatial for spatial relationships, LIBERO-Object for manipulation on various objects, LIBERO-Goal for varying objectives, and LIBERO-Long for extended task duration, each consisting of 10 tasks with 50 human-teleoperated demonstrations per task.

Real-world evaluation. We design 9 tasks to evaluate the capacities of multi-task learning and visual generalization. 30 trajectories are collected for each task via a 6-DoF Lebai robot arm. The average success rate over 20 trials is reported.

- **Multi-task Learning.** 1) *spatial awareness* (Pick up the lemon and put it into the pan; Open the microwave and put the chips into it). 2) *tool usage* (Sweep the cube into the dustpan; Stir water in the bowl with the spoon in the cup). 3) *semantic understanding* (Pour water from a cup into the bowl; Stack the yellow cube on another cube).
- **Visual Generalization.** It includes two aspects: 1) operating on unseen objects (an apple or a banana). 2) Picking and putting a lemon with *complex distractors*.

Method	Spatial	Object	Goal	Long	Average
DiffPolicy	78.3±0.0	92.5±0.0	68.3±0.0	50.5±0.0	72.4±0.0
Octo	78.9±1.0	85.7±0.9	84.6±0.9	51.1±1.3	75.1±0.6
MDT	78.5±0.0	87.5±0.0	73.5±0.0	64.8±0.0	76.1±0.0
OpenVLA	84.7±0.9	88.4±0.8	79.2±1.0	53.7±1.3	76.5±0.6
MaIL	74.3±0.0	90.1±0.0	81.8±0.0	78.6±0.0	83.5±0.0
UniActions	65.0±0.0	78.0±0.0	68.0±0.0	47.0±0.0	64.5±0.0
UniVLA	95.2±0.0	95.4±0.0	91.9±0.0	87.5±0.0	92.5±0.0
Ours	**98.3±1.3**	**99.8±0.4**	**95.6±0.5**	**93.8±0.8**	**96.9±0.7**

Table 2: Success rate (%) on the LIBERO across four suites. Zero standard deviation indicates no average performance.

Implementation Details. We build our SDP on a 12-block Diffusion Transformers (Peebles and Xie 2023) and pre-trained on the OpenX (Vuong et al. 2023) following (Team et al. 2024). For the simulated tasks, the model is fine-tuned on 4 A100 GPUs for 40 epochs, with AdamW as the optimizer and a learning rate of 10^{-4}. The batch size is set to 64, and images from the static and wrist cameras are resized to 224×224. $N_d = 4$ denoising steps are used to generate actions, and we report the average performance of overall tasks over 3 seeds. For real-world evaluation, we only use images from the static camera and train the model for 200 epochs. All results are averaged over 20 trials. Baselines are fine-tuned on real-world data with default hyperparameters.

Performance on Simulated Robotic Manipulation

Baselines. We adopt state-of-the-art diffusion policies that report results for the CALVIN benchmark as baselines, including diffusion policy (Chi et al. 2025) with CNN backbone (DiffPolicy), Octo (Team et al. 2024), MDT (Reuss et al. 2024b), and MoDE (Reuss et al. 2024a). Octo (Team et al. 2024) employs a unified action representation to handle heterogeneous action spaces. MDT leverages diffusion mod-

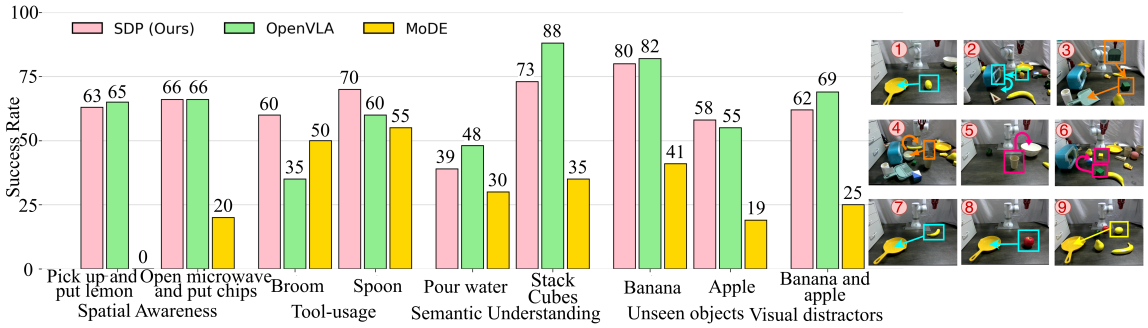


Figure 4: Task success rates (%) on real-world robot manipulation tasks. We specially designed 9 tasks (see the right figure) to evaluate two aspects of policy ability: multi-task learning (the first six tasks) and visual generalization (the last three tasks). In the visual generalization setting, we further investigate the generalization to unseen objects (an apple and a banana) and the robustness to visual distractors. The proposed SDP (pink) consistently outperforms baselines (green and orange), demonstrating better generalization across tasks and objects as well as robustness to distractors.

Method	ABCD→D	ABC→D	LIBERO-Long
Baseline (DP)	1.98±0.09	1.13±0.02	50.5±0.5%
+ Cross Atten.	4.09±0.07	3.59±0.03	81.0±0.8%
+ Prior Injection	4.30±0.07	4.01±0.04	86.0±0.3%
+ Skill Abs.	4.51±0.07	4.32±0.07	91.5±0.7%
+ CPE	**4.67±0.02**	**4.49±0.05**	**93.8±0.8%**

(a) Study on key components.

Strategy	ABCD→D	ABC→D	LIBERO-Long
Addition	4.34±0.02	4.12±0.04	90.9±0.3%
Concatenation	4.41±0.04	4.24±0.06	91.8±0.5%
FiLM	4.49±0.03	4.31±0.02	92.5±0.6%
Eq. (4)	**4.67±0.02**	**4.49±0.05**	**93.8±0.8%**

(b) Study on strategy of skill conditioning.

Table 3: Ablations on the CALVIN and LIBERO-Long.

els to generate flexible action sequences conditioned on multimodal goals. MoDE combines sparse experts with a noise-conditioned self-attention mechanism to achieve more effective denoising across different noise levels. Additional baselines include current state-of-the-art VLA policies. They involve RoboFlamingo (Li et al. 2023), GR-1 (Wu et al. 2023), OpenVLA (Kim et al. 2024), and recent UniVLA (Bu et al. 2025). RoboFlamingo introduces alternative VLAs that use continuous action head predictions instead of discrete ones. GR-1 learns to predict future frames and actions after pre-training. OpenVLA pretrains on large-scale datasets to enable generalist robotic policies. UniVLA derives task-centric action representations from videos with a latent action model. For the LIBERO, MaIL (Jia et al. 2024), and UniActions (Zheng et al. 2025) are additionally compared.

Performance on the CALVIN. Results in Table 1 demonstrate that the proposed SDP consistently outperforms all SOTA policies on both challenges. Additionally, SDP only employs four denoising steps for action generation, significantly fewer than the ten steps in diffusion-based baselines like MDT and MoDE. Specifically, on the ABCD→D setting, SDP surpasses the prior state-of-the-art MDT and MoDE by a considerable margin. On the challenging ABC→D setting, SDP achieves a 76.9% success rate for completing all five tasks in sequence, surpassing the previous best method, MoDE by 14.5%, and recent UniVLA by 20.4%. The average number of consecutively completed tasks increases from UniVLA's 3.80 to 4.49. These results not only confirm that SDP provides strong performance, but also demonstrate its ability to generalize to unseen environment settings and tackle long-horizon manipulation tasks.

Performance on the LIBERO. As shown in Table 2, our SDP demonstrates exceptional performance across all four evaluation suites, achieving high completion rates and significantly outperforming strong baselines, including MaIL and UniVLA. Notably, SDP is the only policy exceeding the success rate of 90% on the LIBERO-Long suite, while other generalist approaches struggle with complex and long-horizon tasks, with only the recent UniVLA achieving competitive performance. What's more, SDP achieves an average performance of 96.9%, surpassing diffusion-based MDT and UniVLA by margins of 13.4% and 4.4%, respectively. Overall, the proposed SDP demonstrates versatility and robustness across a range of robotic manipulation scenarios, leading to a new state-of-the-art on the LIBERO benchmark.

Performance on Real-world Robot Manipulation

Baselines. We compare SDP with the SOTA MoDE, employing the MoE structure, and the representative OpenVLA with a large auto-regressive architecture.

Multi-task learning. The evaluation results are shown in Figure 4. The proposed SDP consistently achieves the best performance, demonstrating a clear advantage in spatial awareness, tool usage, and semantic understanding. Notably, on complex tasks, such as "Open microwave and put chips" and "Pour water", SDP outperforms other methods by a significant margin, indicating its superior ability to learn and generalize across diverse manipulation tasks. This further highlights the effectiveness of our SDP in handling complex and varied tasks within a multi-task learning scenario.

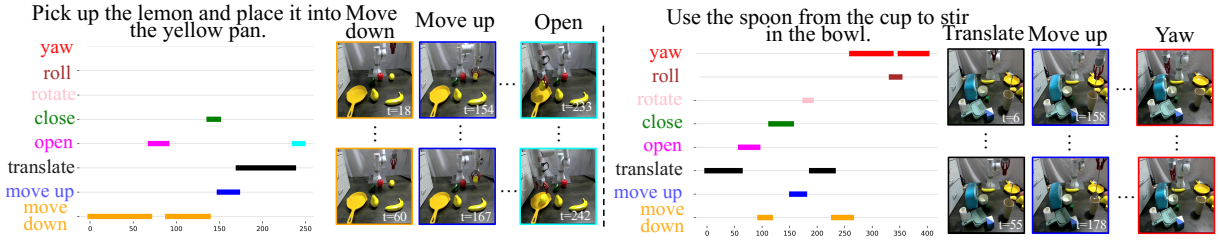


Figure 5: Visualizations on assigned skills. The left plots draw assigned skills at each timestep, where the horizontal axis denotes the timestep, and the vertical axis corresponds to different primitive skills. Images on the right correspond to the observations by performing the skills. SDP learns primitive skills during training and composes them to accomplish complex tasks in inference.

Visual generalizability. This setting evaluates 1) the ability to manipulate objects unseen before, and 2) the robustness to irrelevant objects, known as visual distractors. For the former, we introduce a previously unseen apple and a banana. For the latter, we repeat the task with a lemon, but add more objects nearby to serve as distractors. The corresponding results are shown in Figure 4. The proposed SDP (pink) is capable of manipulating the apple, while baselines (green and orange) struggle with picking and placing it. This is because SDP decomposes a task and produces skill-aware actions aligned with the task on the lemon, which is similar in shape to the apple. However, when faced with a banana, a shape not seen during training, the performance drops, indicating that generalization is more challenging for unfamiliar shapes. More importantly, the presence of visual distractors slightly impacts the success rate (from 75% to 65%), while baselines are confused by visual distractors and perform poorly, which highlights the strong robustness of SDP.

Effectiveness of Design Choices

Study on key components. SDP has several key components: prior injection (including cross-attention and the AdaLN), skill abstraction (used by skill-dependent FFN), and compositional prompt ensemble (CPE). Table 3 (a) evaluates their contribution on the CALVIN and the LIBERO-Long suite. First of all, the baseline has a relatively low performance across all tasks. On the one hand, incorporating vision-language information by cross-attention significantly boosts the results, indicating its effectiveness in feature injection. Further adding other information via AdaLN continues to improve performance, demonstrating the benefit of leveraging prior knowledge. On the other hand, the introduction of skill abstraction leads to additional gains, particularly on the LIBERO-Long suite, where the score increases by 5.5%. Finally, based on the above structure, ensembling the compositional prompts achieves the best performance. All these results validate the importance of each component.

Study on skill conditioning. The skill-dependent FFN helps construct the dependency between the assigned skill and the action prediction. We investigate different strategies for conditioning, including element-wise addition, channel concatenation, and FiLM (Perez et al. 2018). As listed in Table 3 (b), our modeling in Equation (4), parameterizing the FFN layers by the assigned skills, consistently outperforms others across all settings. Specifically, our approach achieves the highest performance on the LIBERO-Long suite, surpassing them by 2.9%, 2.0%, and 1.3%, respectively. Similar findings are observed in the other two. They demonstrate the effectiveness of our strategy for complex tasks.

	#Params	FLOPS	Infer. Time	ABC→D	ABCD→D
Diff-P-T	286M	36.3G	22.1ms	1.13±0.02	1.98±0.09
MoDE	780M	57.4G	30.5ms	3.92±0.07	4.19±0.03
Ours	1017M	74.5G	45.1ms	4.49±0.05	4.67±0.02

Table 4: Complexity analysis on the CALVIN benchmark.

Complexity analysis. We analyze the training and deployment cost in Table 4. Compared to other diffusion-based policies, our SDP has a larger model size and computational cost, but consistently outperforms them by a clear margin across all tasks, with a negligible increase in inference time of 14.6ms. These comparisons demonstrate the effectiveness and efficiency of design choices despite extra overhead.

Visualization Analysis

Figure 5 visualizes the assigned skill at each timestep (left plot) and corresponding observations (right part) from conducting the skill, with the color of borders matching the skill. It is observed that SDP learns to assign reusable skills during training and sequentially composes them to accomplish the overall goals in inference. At the lower level, a diffusion model is conditioned on these skills to produce control signals that enable the desired manipulations, thereby completing complex tasks. Although the skill is assigned without explicit supervision, the visual observations are well aligned with the assigned skills, which demonstrates both the effectiveness and interpretability of our method.

Conclusion

This paper presents SDP, a skill-conditioned diffusion policy that integrates skill learning with conditional diffusion planning. It abstracts primitive skills from different tasks and assigns the appropriate one to guide the action generation. Experiments on both simulated and real-world tasks demonstrate the generalization and robustness, and extensive studies further validate its effectiveness and interpretability.

Acknowledgments

The project is supported in part by the Research Grants Council (RGC) of the Hong Kong SAR through the General Research Fund (17203023), the Collaborative Research Fund (C5052-23G), and the NSFC/RGC Collaborative Research Scheme (CRS_HKU703/24), and in part by UBTECH Robotics. The research work described in this paper was conducted while Zhihao Gu was a Postdoc of Prof. Dong Xu in the JC STEM Lab of Multimedia and Machine Learning, funded by the Hong Kong Jockey Club Charities Trust.

References

Bu, Q.; Yang, Y.; Cai, J.; Gao, S.; Ren, G.; Yao, M.; Luo, P.; and Li, H. 2025. Learning to Act Anywhere with Task-centric Latent Actions. *arXiv:2502.14420*.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2025. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11): 1684–1704.

Dalal, M.; Pathak, D.; and Salakhutdinov, R. R. 2021. Accelerating robotic reinforcement learning via parameterized action primitives. *Advances in Neural Information Processing Systems*, 34: 21847–21859.

Dhakan, P.; Kasmarik, K.; Vance, P.; Rano, I.; and Siddique, N. 2022. Concurrent skill composition using ensemble of primitive skills. *IEEE Transactions on Cognitive and Developmental Systems*, 15(4): 1879–1890.

Doshi, R.; Walke, H.; Mees, O.; Dasari, S.; and Levine, S. 2024. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *arXiv:2408.11812*.

Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O. A.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; and Tompson, J. 2022. Implicit behavioral cloning. In *Conference on robot learning*, 158–168.

Garg, D.; Vaidyanath, S.; Kim, K.; Song, J.; and Ermon, S. 2022. Lisa: Learning interpretable skill abstractions from language. *Advances in Neural Information Processing Systems*, 35: 21711–21724.

Ha, D.; Dai, A.; and Le, Q. V. 2016. Hypernetworks. *arXiv:1609.09106*.

Ha, H.; Florence, P.; and Song, S. 2023. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, 3766–3777. PMLR.

Hiranaka, A.; Hwang, M.; Lee, S.; Wang, C.; Fei-Fei, L.; Wu, J.; and Zhang, R. 2023. Primitive skill-based robot learning from human evaluative feedback. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7817–7824. IEEE.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1): 79–87.

Jia, X.; Wang, Q.; Donat, A.; Xing, B.; Li, G.; Zhou, H.; Celik, O.; Blessing, D.; Lioutikov, R.; and Neumann, G. 2024. Mail: Improving imitation learning with selective state space models. In *8th Annual Conference on Robot Learning*.

Jo, J.; Lee, S.; and Hwang, S. J. 2022. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International conference on machine learning*, 10362–10383. PMLR.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. Openvla: An open-source vision-language-action model. *arXiv:2406.09246*.

Li, X.; Liu, M.; Zhang, H.; Yu, C.; Xu, J.; Wu, H.; Cheang, C.; Jing, Y.; Zhang, W.; Liu, H.; et al. 2023. Vision-language foundation models as effective robot imitators. *arXiv:2311.01378*.

Liang, Z.; Mu, Y.; Ma, H.; Tomizuka, M.; Ding, M.; and Luo, P. 2024. 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*, 16467–16476.

Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36: 44776–44791.

Liu, S.; Wu, L.; Li, B.; Tan, H.; Chen, H.; Wang, Z.; Xu, K.; Su, H.; and Zhu, J. 2024. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv:2410.07864*.

Liu, T.; Li, J.; Zheng, Y.; Niu, H.; Lan, Y.; Xu, X.; and Zhan, X. 2025. Skill expansion and composition in parameter space. *arXiv:2502.05932*.

Mandlekar, A.; Xu, D.; Wong, J.; Nasiriany, S.; Wang, C.; Kulkarni, R.; Fei-Fei, L.; Savarese, S.; Zhu, Y.; and Martín-Martín, R. 2021. What matters in learning from offline human demonstrations for robot manipulation. *arXiv:2108.03298*.

Mees, O.; Hermann, L.; Rosete-Beas, E.; and Burgard, W. 2022. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334.

Mete, A.; Xue, H.; Wilcox, A.; Chen, Y.; and Garg, A. 2024. Quest: Self-supervised skill abstractions for learning continuous control. *Advances in Neural Information Processing Systems*, 37: 4062–4089.

Mishra, U. A.; Xue, S.; Chen, Y.; and Xu, D. 2023. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, 2905–2925. PMLR.

Mokady, R.; Hertz, A.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2023. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6038–6047.

Ni, F.; Hao, J.; Wu, S.; Kou, L.; Liu, J.; Zheng, Y.; Wang, B.; and Zhuang, Y. 2024. Generate subgoal images before act: Unlocking the chain-of-thought reasoning in diffusion model for robot manipulation with multimodal prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13991–14000.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205.

Perez, E.; Strub, F.; De Vries, H.; and Dumoulin, V. 2017. Visual reasoning with a general conditioning layer, Courville. In *In Proceedings of the AAAI Conference on Artificial Intelligence*.

Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Ren, A. Z.; Lidard, J.; Ankile, L. L.; Simeonov, A.; Agrawal, P.; Majumdar, A.; Burchfiel, B.; Dai, H.; and Simchowitz, M. 2024. Diffusion policy policy optimization. *arXiv:2409.00588*.

Ren, H.; Sun, L.; Wang, X.; Zhou, P.; Wu, Z.; Dong, S.; Zou, D.; Zheng, Y.; and Yang, Y. 2025. HyPoGen: Optimization-Biased Hypernetworks for Generalizable Policy Generation. In *The Thirteenth International Conference on Learning Representations*.

Reuss, M.; Pari, J.; Agrawal, P.; and Lioutikov, R. 2024a. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. *arXiv:2412.12953*.

Reuss, M.; Yağmurlu, Ö. E.; Wenzel, F.; and Lioutikov, R. 2024b. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. *arXiv:2407.05996*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Shafiullah, N. M.; Cui, Z.; Altanzaya, A. A.; and Pinto, L. 2022. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35: 22955–22968.

Singh, I.; Blukis, V.; Mousavian, A.; Goyal, A.; Xu, D.; Tremblay, J.; Fox, D.; Thomason, J.; and Garg, A. 2022. Progprompt: Generating situated robot task plans using large language models. *arXiv:2209.11302*.

Song, M.; Deng, X.; Zhou, Z.; Wei, J.; Guan, W.; and Nie, L. 2025. A survey on diffusion policy for robotic manipulation: Taxonomy, analysis, and future directions. *Authorea Preprints*.

Team, O. M.; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; et al. 2024. Octo: An open-source generalist robot policy. *arXiv:2405.12213*.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Vincent, P. 2011. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674.

Vuong, Q.; Levine, S.; Walke, H. R.; Pertsch, K.; Singh, A.; Doshi, R.; Xu, C.; Luo, J.; Tan, L.; Shah, D.; et al. 2023. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*.

Wang, L.; Chen, X.; Zhao, J.; and He, K. 2024a. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Advances in neural information processing systems*, 37: 124420–124450.

Wang, Y.; Zhang, Y.; Huo, M.; Tian, R.; Zhang, X.; Xie, Y.; Xu, C.; Ji, P.; Zhan, W.; Ding, M.; et al. 2024b. Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. *arXiv:2407.01531*.

Wu, H.; Jing, Y.; Cheang, C.; Chen, G.; Xu, J.; Li, X.; Liu, M.; Li, H.; and Kong, T. 2023. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv:2312.13139*.

Wu, J.; Sun, X.; Zeng, A.; Song, S.; Lee, J.; Rusinkiewicz, S.; and Funkhouser, T. 2020. Spatial action maps for mobile manipulation. *arXiv:2004.09141*.

Xiao, B.; Wu, H.; Xu, W.; Dai, X.; Hu, H.; Lu, Y.; Zeng, M.; Liu, C.; and Yuan, L. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4818–4829.

Xiong, Z.; Vuorio, R.; Beck, J.; Zimmer, M.; Shao, K.; and Whiteson, S. 2024. Distilling morphology-conditioned hypernetworks for efficient universal morphology control. *arXiv:2402.06570*.

Ye, S.; Jang, J.; Jeon, B.; Joo, S.; Yang, J.; Peng, B.; Mandlekar, A.; Tan, R.; Chao, Y.-W.; Lin, B. Y.; et al. 2024. Latent action pretraining from videos. *arXiv:2410.11758*.

Ze, Y.; Zhang, G.; Zhang, K.; Hu, C.; Wang, M.; and Xu, H. 2024. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv:2403.03954*.

Zhang, B.; and Sennrich, R. 2019. Root mean square layer normalization. *Advances in neural information processing systems*, 32.

Zhang, J.; Zhang, J.; Pertsch, K.; Liu, Z.; Ren, X.; Chang, M.; Sun, S.-H.; and Lim, J. J. 2023. Bootstrap your own skills: Learning to solve new tasks with large language model guidance. *arXiv:2310.10021*.

Zhang, Q.; Tao, M.; and Chen, Y. 2022. gddim: Generalized denoising diffusion implicit models. *arXiv:2206.05564*.

Zheng, J.; Li, J.; Liu, D.; Zheng, Y.; Wang, Z.; Ou, Z.; Liu, Y.; Liu, J.; Zhang, Y.-Q.; and Zhan, X. 2025. Universal actions for enhanced embodied foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 22508–22519.