

# SD<sup>2</sup>Actor: Continuous State Decomposition via Diffusion Embeddings for Robotic Manipulation

Jiayi Li

National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,  
 National Engineering Research Center for Visual Information and Applications,  
 Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

jiayili@stu.xjtu.edu.cn

## Abstract

Language-conditioned robot manipulation in the continuous spectrum presents a persistent challenge due to the difficulty of mapping states to target actions. Previous methods face limitations in effectively modeling object states, primarily due to their reliance on executing ambiguous instructions devoid of explicit state information. In response, we present SD<sup>2</sup>Actor, a zero-shot robotic manipulation framework that possesses the capability to generate precise actions in continuous states. Specifically, given the novel instructions, we aim to generate instruction-following and accurate robot manipulation actions. Instead of time-consuming optimization and finetuning, our zero-shot method generalizes to any object state with a wide range of translations and versatile rotations. At its core, we quantify multiple base states in the training set and utilize their combination to refine the target action generated by the diffusion model. To obtain novel state representations, we initially employ LLMs to extract the novel state from the instruction and decompose it into multiple learned base states. We then employ the linear combination of base state embeddings to produce novel state features. Moreover, we introduce the orthogonalization loss to constrain the state embedding space, which ensures the validity of linear interpolation. Experiments demonstrate that SD<sup>2</sup>Actor outperforms state-of-the-art methods across a diverse range of manipulation tasks in ARNOLD Benchmark. Moreover, SD<sup>2</sup>Actor can effectively learn generalizable policies from a limited number of human demonstrations, achieving promising accuracy in a variety of real-world manipulation tasks.

## 1. Introduction

Learning generalizable robotics policies under the guidance of instruction has attracted increasing attention in the field of embodied AI and robotics. Recent language-conditioned

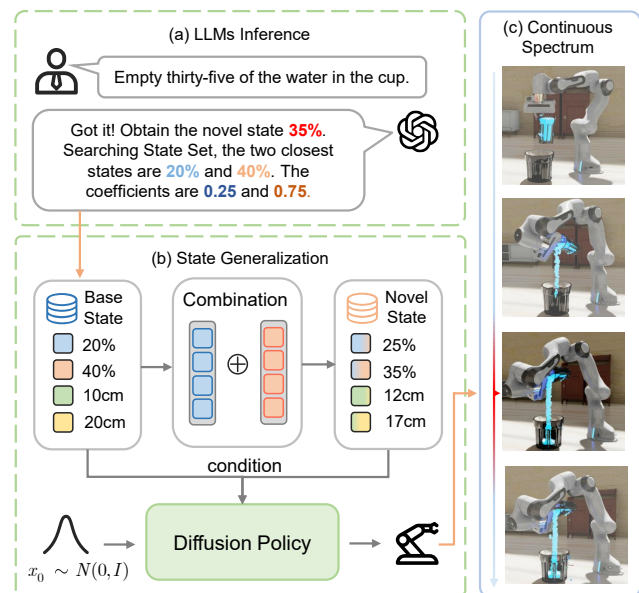


Figure 1. **Zero-shot generalization of SD<sup>2</sup>Actor.** SD<sup>2</sup>Actor facilitates language-conditional robot manipulation driven by continuous goal states. **(a)** Our model employs LLMs to analyze and decompose object states in instructions. **(b)** The model uses linear combination to represent learned embeddings as novel state features and as conditions to guide diffusion to generate actions **(c)** where reflects the continuous and previously unseen object state.

robot manipulation [7, 11, 15, 55] have showcased great potential for end-to-end imitation learning that directly maps goal states to robot actions. However, these methods require training with a large number of demonstrations and exhibit poor state generalization.

Recently, previous robot manipulation approaches [11, 14, 18, 28, 46, 49, 55] mainly focus on tasks with discrete object states, such as “open the cabinet”, where states are binary (*i.e.*, open and close). But these approaches ignore continuity of the states of target object (*e.g.*, cabinet) which ranges from 0% to 100%. Meanwhile, ARNOLD [12] di-

rectly inputs continuous state values into PerAct [32] to refine the actions, validating the essentiality of state modeling in language-conditioned manipulation, specifically in terms of grounding. However, the state information can be overwritten by a lot of redundant information, resulting in inaccurate generated actions. Thus, there is an urgent need for robot systems to learn a mapping from precise goal states to robot actions in a continuous world. Concurrently, given that states are continuous, only a finite number of states exist during training. Therefore, robots have to learn generalizable policies from finite states to handle novel states. Although there are some zero-shot robot manipulation methods [10, 19, 39, 42, 51] that have grounded language in static object properties such as colors and shapes to achieve generalizability. But they do not attempt in-depth physical state generalization, neglecting the problem of grounding language to continuous object states. Therefore, it is crucial for the robot manipulation system to understand the physical state and translate it into precise actual actions.

To address these critical challenges of language-conditioned robot task learning, we propose SD<sup>2</sup>Actor, a novel zero-shot state generalization approach that performs state decomposition for generating the precise action responding to the instruction. As depicted in Figure 1, our key insight is to use embeddings to represent different states to enhance the understanding of the physical state. Subsequently, LLMs is used to decompose the novel state value into multiple learned states in the same task. To generalize to the new states, we design the linear combination of the decomposition coefficients with the appropriate embeddings. Considering the continuity of the action distribution in the continuous state, we leverages the inherent modeling capabilities of diffusion model [5, 13, 24, 31] to represent continuous action distribution [9, 27, 33] and generate the new action consistent with the state.

The workflow of SD<sup>2</sup>Actor is as follows: 1) During training, we utilize diffusion model to model the distribution of the action space. Meanwhile, we incorporate learnable embeddings into instructions to separately represent different object states. These embeddings are used as conditions to generate actions. This approach effectively supplements the information available about the object state, which assists the model in understanding the state and mapping it to the spatial action. 2) During inference, the comprehensive reasoning capabilities of the LLMs is employed to analyze and decompose the novel state value mentioned in the instructions. By the linear combination among learned embeddings in same task, the model can generalize and generate the action corresponding to state, which empowers robot to generalize to the new state in the physical space. To summarize, we have three main contributions:

1) We introduce a novel state modeling framework that addresses the challenge of understanding continuous object

states and generating the corresponding accurate actions in language-conditioned robot task learning.

- 2) We leverage the comprehensive reasoning capability of LLMs to decompose the novel state and implement linear combination approach to obtain the new state representation that assist the agent in generalizing to the new state.
- 3) We demonstrate state-of-the-art performance and of the proposed method in a continuous state spectrum.

## 2. Related Work

### 2.1. Diffusion policy for Robotic Manipulation

Diffusion models, a powerful class of generative models that learn to approximate the data distribution by iterative denoising processes, make great breakthroughs in image generation [16, 30, 34, 35] recent years. Owing to the impressive performance of the diffusion model, it is also used in robotics applications including robot manipulation [5, 25, 46, 48], visual navigation [6, 36, 50], dexterous hand [43, 44, 52]. 3D Diffuser Actor [20] proposes to predict the next 3D keypose for the robot’s end-effector alongside the linking trajectory. We adopt the model structure of 3D Diffuser Actor, and enhance it by incorporating learnable state embeddings. These embeddings aid the diffusion process, enabling the model to better differentiate between continuous goal states with greater precision.

### 2.2. Zero-Shot Robotic manipulation

Zero-shot learning methods [41, 45] leverage extensive pre-training data to facilitate the rapid adaptation of models to new samples during the testing phase, even without the presence of any training examples. This technique has been used in the fields of computer vision [2, 22, 41], natural language processing [26, 29] and embodied AI [17, 38]. In robot manipulation, most zero-shot methods [39, 51] tend to achieve effective manipulation of objects and trajectory generation by capturing geometric invariance and trajectory similarity features for generalizing new objects, scenes, and different tasks. Nevertheless, they lack precise and comprehensive metrics for novel state, resulting in limited effectiveness in the physical world. In contrast, our approach differs as it focuses on addressing the challenge of generalization across various states in a continuous spectrum.

### 2.3. LLMs for Robotic Manipulation

LLMs have demonstrated exceptional capabilities in generalization [1, 40], as well as commonsense reasoning [37, 54]. Recently, there has been a growing demand to incorporate LLMs into robotics systems [8, 23]. LLMs are typically used mostly for instruction decomposition in robotics, such as ManipLLM [23], which combines instructions with image embeddings to gradually refine and decompose instructions into the form of coordinates and directions, maintain-

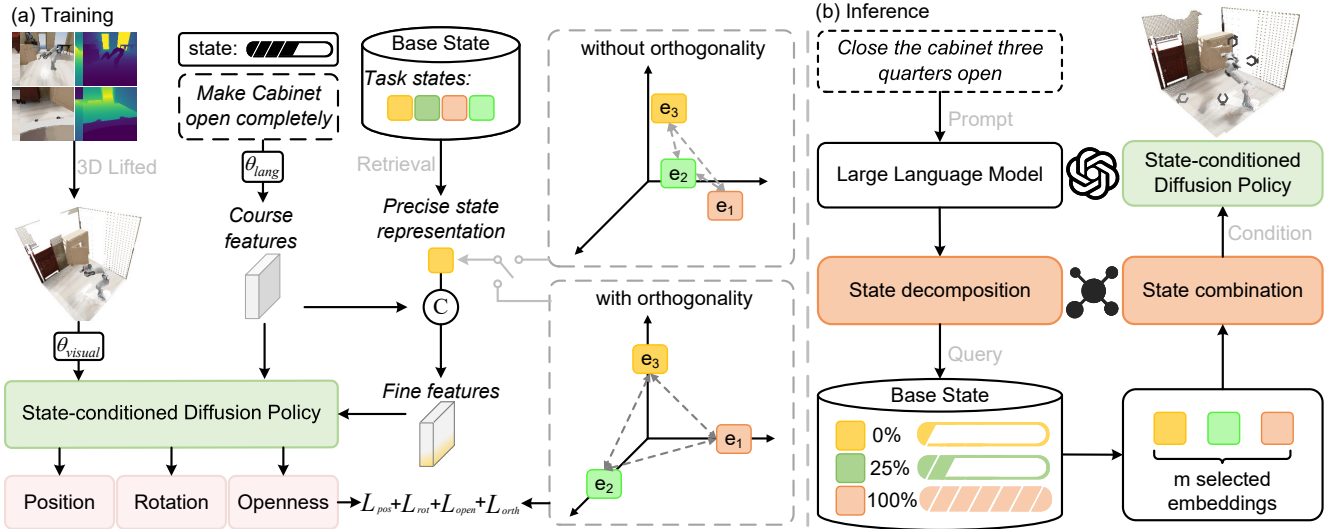


Figure 2. **Overview of our SD<sup>2</sup> Actor.** (a) During training, we utilize DDPM to generate actions. Simultaneously leveraging the Base State  $B$  for retrieving fine-grained state representation, the state-conditioned diffusion model would generate the accurate and stable action. Additionally, we introduce the orthogonalization loss which helps to create a distinct separation between embeddings. (b) During inference, we employ LLMs to decompose states in the instruction into previously observed states during training. By performing the state linear combination of  $m$  learned embeddings, the diffusion process can be guided to generalize the actions in the novel state.

ing a good generalization performance in precisely manipulating different objects. The other category is to decompose a long-horizon task instruction into simple sub-tasks in a sequential manner, and the model is able to execute the task sequentially. However, our approach leverages the robust knowledge-based reasoning of LLMs to analyze and decompose the goal states specified in the same task. This enables us to ensure the stability of previously unseen states and ultimately achieve improved generalization.

### 3. Methodology

Given a set of manipulation tasks demonstrations, which includes language instructions  $l$ , scene observation  $o$ , and agent proprioception  $c$ , we aim to generate the action  $a$  based on a full understanding of the state  $s$ . The state information is included in the instructions  $l$ . The action  $a$  should ensure both the stability and reachability of the target position of the object in the 3D scene, while adhering to the specified degree of variation required by the instruction.

Our proposed approach, SD<sup>2</sup> Actor, leverages Denoising Diffusion Probabilistic Models (DDPM) as the component of our action generation framework, as depicted in Figure 2 (a). Meanwhile, we introduce a fine-grained action control branch where we utilize learnable embeddings to represent different goal state features in the same task and we refer to as Base State  $B$ . During the training phase, the model retrieves the corresponding embedding from  $B$  based on the state. The embeddings in  $B$  are not shared across tasks. To enable effective novel state  $s_{novel}$  control in inference,

we design state decomposition and combination module, as shown in Figure 2 (b), which explicitly interpolates novel state embedding based on multiple embeddings in  $B$  and injects it to the diffusion process as an additional condition. Lastly, we discuss state generalization through the proposed orthogonal constraint.

#### 3.1. Preliminary

**Diffusion models** are innovative generative techniques that iteratively transform a simplistic noise distribution  $\pi_1$  into a complex data distribution  $\pi_0$ , gradually reversing a diffusion process and adding noise at each step. During the forward diffusion process, the sample  $x_0$  adds corresponding noise as the time step  $t$  increases, which is represented by a discrete-time stochastic process  $\{x_t\}_{t=0}^T$  where  $x_0 \sim \pi_0$ , and  $x_t \sim \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_t/\bar{\alpha}_{t-1}}x_{t-1}, (1 - \bar{\alpha}_t/\bar{\alpha}_{t-1}))$ . The decreasing scalar function  $\bar{\alpha}_t$ , with constraints that  $\bar{\alpha}_0 = 1$  and  $\bar{\alpha}_T \approx 0$ , controls the noise level through time. The forward process can be formulated as:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, 1), \quad (1)$$

where  $\alpha = 1 - \beta$ ,  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ .  $\beta_t \in (0, 1)$  represents variance schedule which defines how much noise is added at each time step. For the reverse process, a noise-predictor  $\epsilon_\theta(x_t; t)$  is trained to predict the noise component  $\epsilon$  at each step takes from the noise-added sample  $x_t$  with the guidance of condition  $c_t$  until the original distribution  $\pi_0$  is recovered. The usual loss can be simplified as:

$$\mathcal{L} = \mathbb{E}_{x_t, c_t} \|\epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, c_t, t) - \epsilon\|. \quad (2)$$

On account of the above theoretical foundation, we choose to model the continuous action distribution with the diffusion model. Furthermore, we leverage generalizable state embeddings to guide the diffusion process during inference.

### 3.2. Refined Action Generation

In order to ensure accurate action generation, we adopt the structure of the 3D Diffuser Actor [20], which enables the direct encoding of point cloud features and facilitates the modeling of continuous action distributions. However, providing only a textual description for diffusion model to comprehensively recover all its components is often a challenging undertaking. Inspired by [39], we opt for explicitly integrating state embeddings of the object into the diffusion to empower the state understanding of the model.

In this work, we initialize a set of learnable embeddings  $\{e_1, e_2 \dots e_n\}$  for each task to represent the base states  $\{s_1, s_2 \dots s_n\}$  appearing in the training set, where  $n$  denotes the number of goal states of the object. Embeddings can represent state features and the set that these base state embeddings formed is referred to as the Base State  $B$ . During training, we retrieve the state embeddings  $e_i$  from  $B$  according to state information  $s_i$  in the current instruction  $l$ . To strengthen the state understanding capability of the model, we enhance the attention of the model to the state by feature fusion between language  $l$  and embedding  $e_i$ . We use self-attention, where  $K$  and  $V$  are defined as follows:

$$K = W_K \cdot (\varphi(l) \oplus \varphi_{state}(e_i)), \quad (3)$$

$$V = W_V \cdot (\varphi(l) \oplus \varphi_{state}(e_i)), \quad (4)$$

where  $\oplus$  denotes concatenation operation,  $W_K, W_V$  are projection matrices,  $\varphi(\cdot)$  represents CLIP text encoder and  $\varphi_{state}(\cdot)$  denotes state encoder consisting of two MLP layers and one ReLU nonlinear layer. The attention result is then interacted with scene information to guide the denoising process during diffusion.

Ultimately, the model predicts action  $a$  of the gripper at the next keyframe. The SD<sup>2</sup>Actor is optimized by sampling a random diffusion step and calculate the  $t$ -step the  $\mathcal{L}_1$  loss of translation  $a_t^{pos}$ , rotation  $a_t^{rot}$  and the openness  $a_t^{open}$ . The  $t$ -step corresponding objective function can be expressed as:

$$\mathcal{L}_{pos} = \|\epsilon_\theta^{pos}(o, l, e_i, a_t^{pos}, a_t^{rot}) - \epsilon_t^{pos}\|, \quad (5)$$

$$\mathcal{L}_{rot} = \|\epsilon_\theta^{rot}(o, l, e_i, a_t^{pos}, a_t^{rot}) - \epsilon_t^{rot}\|, \quad (6)$$

$$\mathcal{L}_{open} = \text{BCE}(f_\theta^{open}(o, l, e_i, a_t^{pos}, a_t^{rot}), a^{open}), \quad (7)$$

where  $\epsilon_\theta$  and  $\epsilon_t$  denote the predicted noise and ground truth, respectively.  $f_\theta^{open}$  represent the openness of gripper, BCE is binary cross-entropy loss function.

In this way, at the end of training, each trained embedding is capable of representing a distinct object state feature,

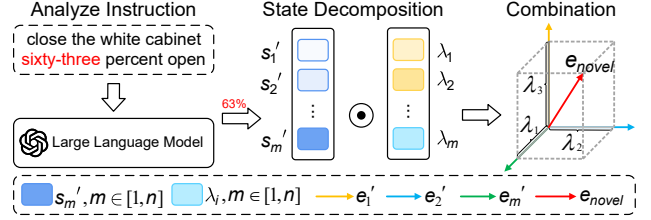


Figure 3. **State decomposition and combination Module.** LLMs utilize  $\{s'_1, s'_2 \dots s'_n\}$  to decompose novel state  $S_{novel}$  to obtain the coefficient  $\{\lambda_1 \dots \lambda_m\}$ , and then a linear combination with the embedding  $\{e'_1 \dots e'_m\}$  is performed to get a new embedding  $e_{novel}$ .

which gradually improves through the diffusion denoising process. The trained diffusion model effectively maps states to spatial locations, enabling us to obtain refined actions by inputting state embeddings into the diffusion model.

### 3.3. Generalizable State Decomposition

In the inference phase, our aim is to achieve control over generated actions in the novel state without influencing the accurate generative capabilities of the trained diffusion model. The fact is that the novel embedding  $e_{novel}$  is not in the Base State  $B$ , so  $e_{novel}$  cannot be retrieved directly. The straightforward design would be to employ the state feature extracted from MLPs as diffusion conditions. However, we argue that such condition states learned from few demonstrations often suffer from a problem similar to identity shortcut [47], where the model simply matches learned states. Herein, the actions generated within the continuous spectrum are even worse.

Instead, considering that the state values under the same task are within a continuous spectrum, then the state embeddings of the same task should also be correlated. Therefore, we can utilize the different state embeddings of same task in  $B$  to combine into novel state embeddings. Further, we can obtain the combination coefficients by analyzing the numerical correlation between  $s_{novel}$  and the states  $\{s_1, s_2 \dots s_n\}$  in the  $B$ . However, due to the diverse forms of expressions and the lack of direct clarity regarding continuous state values, manually extracting state values  $s_{novel}$  from instructions is laborious. Therefore, we leverage the knowledge generalization capability of LLMs to obtain object states in the language. Especially, as shown in Figure 3, we input the instruction  $l$  into LLMs to assist the agent to obtain the state value  $s_{novel}$  included in the  $l$ . LLMs can retrieve the  $m$  learned states closest to  $s_{novel}$  of this task and decompose  $s_{novel}$  into linear combinations of  $\{s'_1, s'_2 \dots s'_m\}$ . The intrinsic form is as follows:

$$s_{novel} = \lambda_1 s'_1 + \lambda_2 s'_2 + \dots + \lambda_m s'_m, \quad (8)$$

$$\lambda_1 + \lambda_2 + \dots + \lambda_m = 1,$$

where  $m \leq n$ ,  $s'_i \in \{s_1, s_2 \dots s_n\}$  and  $\lambda_i$  denote  $i$ th state

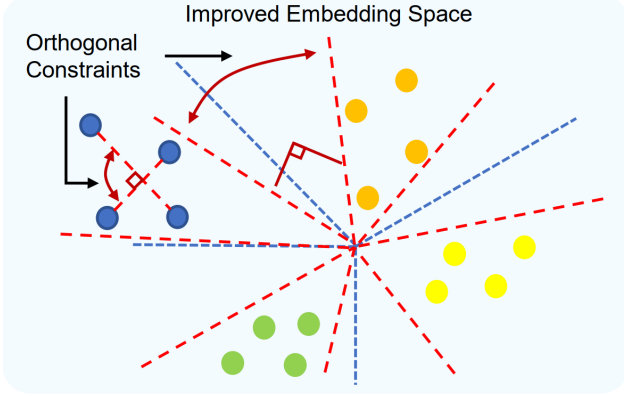


Figure 4. **Orthogonalized embedding space.** Scatters in different colors represent distinct state embeddings, and a certain distance between them is necessary.

value and decomposition coefficient obtained from the decomposition. In order to generalize to novel state embedding  $e_{novel}$ , we utilize the decomposition coefficients  $\{\lambda_1, \lambda_2 \dots \lambda_m\}$  and the coherent embeddings  $\{e'_1, e'_2 \dots e'_m\}$  to obtain the novel state embedding  $e_{novel}$  using linear combinations. The calculations are as follows:

$$e_{novel} = \lambda_1 e'_1 + \lambda_2 e'_2 + \dots + \lambda_m e'_m. \quad (9)$$

In this way, diffusion model can be guided with a novel state embedding  $e_{novel}$  to generate actions  $a_{novel}$  responding to the novel state  $s_{novel}$ . Our design unifies the state modeling setting in training and inference, and facilitates the natural refinement of state representation. So that, our state decomposition and linear combination module allow for the seamless implementation of action control in arbitrary states, thereby significantly enhancing the generalization of language-conditional robot manipulation.

### 3.4. Orthogonal Embedding Regularization

Although we apply a linear combination of learned embeddings to realize the generalization of the novel state. Nevertheless, performing effective linear interpolation in the embedding space requires satisfying two fundamental prerequisites: 1) The state embeddings space should satisfy closure. 2) The state embeddings under each task should be unique. Inspired by [4], the orthogonality ensures that the embedding space satisfies closure, preventing interpolation from being out of distribution. It also imposes constraints between embeddings so that they will not be identical. Therefore, to ensure the validity of the linear combinations, we adopt the Spectral Restricted Isometry Property Regularization [3] during training. Subsequently, to achieve better separation of tasks, we impose larger constraints between tasks and smaller constraints within tasks. As illustrated in Figure 4, the orthogonal loss is divided into two parts: inter-task loss and intra-task loss. For inter-task

loss, we calculate the average  $\mathbf{E}_{mean}$  of the embeddings in  $B$  within each task, and thus compute the orthogonal loss  $\mathcal{L}_{inter}$  among tasks. For intra-task loss, we directly compute the orthogonal loss  $\mathcal{L}_{intra}$  within the task. The orthogonal  $\mathcal{L}_{orth}$  is given by:

$$\mathcal{L}_{inter} = \gamma_1 \cdot \sigma(\mathbf{E}_{mean}^T \mathbf{E}_{mean} - \mathbf{I}), \quad (10)$$

$$\mathcal{L}_{intra} = \gamma_2 \cdot \sum_{i=1}^K \sigma(\mathbf{B}_i^T \mathbf{B}_i - \mathbf{I}), \quad (11)$$

$$\mathcal{L}_{orth} = \mathcal{L}_{inter} + \mathcal{L}_{intra}, \quad (12)$$

where  $\mathbf{B}_i^T$  represents Base State for the  $i$ -th task.  $\gamma_1$  and  $\gamma_2$  are hyperparameters, we set the default values as 5.0 and 1.0.  $\sigma(E) = \sup_{z \in \mathbb{R}^n, z \neq \mathbf{0}} \frac{\|Ez\|}{\|z\|}$  is spectral norm of  $E$ .  $\mathbf{I}$  is a unit vector. The orthogonality allows us to achieve a more stable and robust nonlinear space, resulting in improved accuracy of the linear combination. For a detailed derivation procedure, please refer to [3], where we replace the network parameters with those of the embeddings. Therefore, the total loss  $\mathcal{L}_{total}$  can be expressed as:

$$\mathcal{L}_{total} = w_1 \mathcal{L}_{pos} + w_2 \mathcal{L}_{rot} + \mathcal{L}_{open} + \mathcal{L}_{orth}, \quad (13)$$

where  $w_1, w_2$  are hyperparameters. By optimizing  $\mathcal{L}_{total}$ , we can generalize the robot's actions for any state. The features of novel state are obtained through linear combination of embeddings, and trained diffusion is used to predict the target action in a continuous spectrum.

At inference time, the next keyframe action is sampled via a reverse diffusion process following DDPM:

$$a_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( a_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(o, l, e_{novel}, a_t) \right) + \sigma_t \delta, \quad (14)$$

where  $\delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and  $\sigma_t$  is the noise level at time step  $t$ . The initial action  $a_T$  is sampled from Gaussian distribution  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ . A more detailed example is provided in the supplementary materials.

## 4. Experiments

In this section, we conduct our experiments, which include a comprehensive evaluation on the ARNOLD [12] and then make comparisons with other methods to illustrate the efficacy of our method in the real world. Finally, we demonstrate that SD<sup>2</sup>Actor is capable of solving challenging tasks which involve generalizable continuous state manipulation and apply effectively on highly precise manipulation.

### 4.1. Datasets

In this work, we evaluate our state decomposition model on ARNOLD. It is a multi-task robot manipulation dataset specifically designed for continuous states. There are 40

	P.OBJECT		R.OBJECT		O.DRAWER		C.DRAWER		O.CABINET		C.CABINET		P.WATER		T.WATER		Average	
6D-CLIPort[53]	6.72	25.37	0.00	0.00	0.00	0.00	0.00	2.70	0.00	0.00	0.00	5.83	0.00	0.00	0.00	7.14	0.84	5.13
PerAct[32]	94.78	95.52	24.68	28.57	36.13	52.94	60.14	68.24	23.19	49.28	30.10	48.54	49.25	85.07	28.57	53.57	43.36	60.22
PerAct(MT)[32]	90.30	92.54	14.29	20.78	25.21	47.90	33.78	56.76	20.29	39.13	19.42	37.86	26.87	64.18	17.86	30.36	31.00	48.69
3DA[20]	89.55	91.79	19.48	20.78	35.29	56.30	61.49	71.62	20.29	42.03	27.18	44.66	52.24	86.57	32.14	58.93	42.21	59.08
<b>SD<sup>2</sup>Actor</b>	<b>95.52</b>	<b>97.76</b>	<b>25.97</b>	<b>29.87</b>	<b>50.42</b>	<b>73.95</b>	<b>70.27</b>	<b>81.76</b>	<b>31.88</b>	<b>60.87</b>	<b>36.89</b>	<b>55.34</b>	<b>55.22</b>	<b>92.54</b>	<b>35.71</b>	<b>66.07</b>	<b>50.23</b>	<b>69.77</b>
<b>SD<sup>2</sup>Actor(MT)</b>	<b>91.79</b>	<b>97.01</b>	<b>15.58</b>	<b>24.68</b>	<b>38.66</b>	<b>67.23</b>	<b>45.27</b>	<b>68.92</b>	<b>27.54</b>	<b>52.17</b>	<b>25.24</b>	<b>43.69</b>	<b>29.85</b>	<b>67.16</b>	<b>16.07</b>	<b>32.14</b>	<b>36.25</b>	<b>56.62</b>

Table 1. **Test set Performance on ARNOLD.** We evaluate learned states on 8 challenging tasks from ARNOLD and report mean success rates (in %) for 20 episodes per task across 3 random seeds. MT denotes the model for multi-task learning. The black figures represent test performances from scratch. The gray figures indicate performances with the first-phase ground truth to better demonstrate how well models understand goal states. The table below is similarly applicable.

	P.OBJECT		R.OBJECT		O.DRAWER		C.DRAWER		O.CABINET		C.CABINET		P.WATER		T.WATER		Average	
6D-CLIPort[53]	0.00	0.00	0.00	0.00	0.00	0.57	0.75	1.13	0.00	0.83	0.00	2.78	0.00	0.00	0.00	16.81	0.09	2.77
PerAct[32]	0.68	2.38	0.48	0.00	10.06	12.93	13.58	18.11	0.00	6.22	0.00	8.33	2.15	1.61	5.88	2.52	4.10	6.51
PerAct(MT)[32]	2.04	3.06	0.95	2.38	9.20	18.68	6.98	11.13	0.00	3.73	2.78	11.11	6.45	9.14	1.68	4.20	3.76	7.93
3DA[20]	1.02	1.70	1.43	0.48	10.34	13.79	13.96	18.30	1.24	7.47	1.39	12.50	2.69	1.61	6.72	4.20	4.85	7.51
<b>SD<sup>2</sup>Actor</b>	<b>3.40</b>	<b>5.44</b>	<b>10.95</b>	<b>16.19</b>	<b>25.86</b>	<b>32.47</b>	<b>26.04</b>	<b>35.85</b>	<b>8.71</b>	<b>17.01</b>	<b>5.56</b>	<b>16.67</b>	<b>7.53</b>	<b>9.68</b>	<b>12.61</b>	<b>17.65</b>	<b>12.58</b>	<b>18.87</b>
<b>SD<sup>2</sup>Actor(MT)</b>	<b>1.02</b>	<b>2.38</b>	<b>7.14</b>	<b>13.81</b>	<b>16.67</b>	<b>29.31</b>	<b>20.75</b>	<b>26.60</b>	<b>6.64</b>	<b>12.45</b>	<b>1.39</b>	<b>9.72</b>	<b>1.61</b>	<b>1.61</b>	<b>8.40</b>	<b>15.13</b>	<b>7.95</b>	<b>13.88</b>

Table 2. **Novel State set Performance on ARNOLD.** We evaluate the unique novel state per task on 8 challenging tasks from ARNOLD and report mean success rates(%) across 3 random seeds.

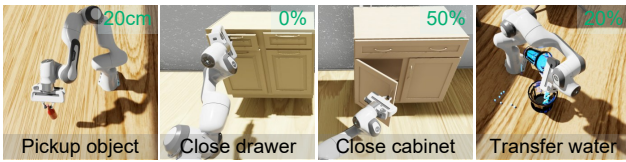


Figure 5. **Qualitative result of our SD<sup>2</sup>Actor on test set.** We visualized four tasks. Green figures are the state value.

distinct objects and 20 diverse scenes which include eight tasks with various goal state variations. We divide each task following ARNOLD into train set, test set, novel and any state set. The test set includes the 773 demos with goal states that have been learned during training. The novel state set contains 2000 demos with only one unseen goal state. The any state set contains 2000 goal states (learned and unlearned) covering the entire manipulation state space. Notably, our method is based on base states, not base skills.

**Metrics.** We have adopted success rate as the evaluation metric in the ARNOLD. A task instance is regarded as a success when the success condition is satisfied continually for 2 seconds, which requires the current state to be within a tolerance threshold of the goal state.

**Implementation Details.** During training stage, we use AdamW [21] optimizer with an initial learning rate of  $1 \times 10^{-5}$  and a cosine scheduler with warmup in the first 2k steps. In practice, we initial 3 embeddings for each task, which represent 3 different goal states. At every iteration, we use Farthest Point Sampling(FPS) to sample 4096 points to reduce computational effort. We train our diffusion model for 800K iterations with a batch size of 96 on  $2 \times$  NVIDIA GeForce RTX 3090 GPUs.

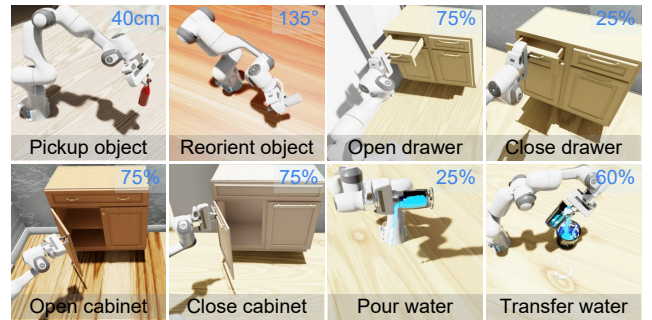


Figure 6. **Qualitative result of our SD<sup>2</sup>Actor on novel state set.** Blue figures are the state value.

## 4.2. Quantitative Evaluations

We evaluate our method against the previous state-of-the-art methods 6D-CLIPort [53] and PerAct [32] on ARNOLD. For a fair comparison, we reproduced the baseline 3D Diffuser Actor [20](3DA) in ARNOLD. We show quantitative results in test set, novel state set and any state set. The detailed results with mean and standard deviation of success rates (%) across 3 random seeds in the supplementary.

**State Modeling Performance.** As shown in Table 1, On test set, our method achieves the best performance with an average success rate of 50.23% and 69.77% on the single task, an absolute improvement of 8.02% and 10.69% over 3D Diffuser Actor. Especially in tasks such as open drawer, close drawer, open cabinet and close cabinet, where the agent is required to have comprehensive state understanding and accurate action generation, state modeling yields better results. It indicates that our model possesses the ability to generate actions with

	P.OBJECT		R.OBJECT		O.DRAWER		C.DRAWER		O.CABINET		C.CABINET		P.WATER		T.WATER		Average	
6D-CLIPort[53]	10.45	29.10	1.30	2.60	0.84	0.00	0.68	1.35	0.00	5.80	0.00	2.91	0.00	1.49	0.00	7.14	1.66	6.30
PerAct[32]	48.51	47.76	14.29	14.29	21.01	33.61	23.65	28.38	4.35	24.64	6.80	13.59	26.87	31.34	14.29	19.64	19.97	26.66
PerAct(MT)[32]	46.27	47.01	12.99	12.99	12.61	23.53	14.86	26.35	4.35	5.80	4.85	8.74	16.42	25.37	3.57	5.36	14.49	19.39
3DA[20]	47.01	47.76	15.58	16.88	22.69	35.29	25.00	31.08	8.70	21.74	8.74	15.53	28.36	34.33	16.07	21.43	21.52	28.01
<b>SD<sup>2</sup>Actor</b>	<b>50.75</b>	<b>51.49</b>	<b>19.48</b>	<b>24.68</b>	<b>36.97</b>	<b>52.94</b>	<b>37.84</b>	<b>50.00</b>	<b>14.49</b>	<b>36.23</b>	<b>12.62</b>	<b>23.30</b>	<b>31.34</b>	<b>38.81</b>	<b>19.64</b>	<b>30.36</b>	<b>27.89</b>	<b>38.48</b>
<b>SD<sup>2</sup>Actor(MT)</b>	49.25	49.25	16.88	20.78	29.41	46.22	29.05	46.62	13.04	28.99	9.71	18.45	25.37	28.36	16.07	25.00	23.60	32.96

Table 3. **Any State set Performance on ARNOLD.** We evaluate the continuous states per task on 8 challenging tasks from ARNOLD and report mean success rates (%) across 3 random seeds.

State Repr.	State Decom.	Ortho.	Novel State	Any State
✗	✗	✗	4.85±0.81	21.52±0.98
✓	✗	✗	5.73±0.78	22.61±0.56
✓	✓	✗	10.84±0.78	25.19±0.64
✓	✗	✓	6.26±0.18	24.87±0.38
✓	✓	✓	<b>12.58±0.38</b>	<b>27.89±0.41</b>

Table 4. **Ablation on each component of our pipeline.** Note the absence of state representation during ablation results in the non-existence of the subsequent two parts as well.

higher accuracy based on the goal states.

**Novel State Generalization Performance.** To valid the effectiveness of SD<sup>2</sup>Actor for state generalization, we analyze it on the novel state. The results in Table 2 demonstrate that SD<sup>2</sup>Actor achieves the highest average success rate of 12.58% and outperforms all previous methods by a wide margin across all tasks. Moreover, we observed that in novel state set, in tasks like open drawer, SD<sup>2</sup>Actor is superior to prior work by a large margin, demonstrating that the model can be generalized to unlearned states.

**Continuous State Generalization Performance.** Furthermore, as shown in Table 3, SD<sup>2</sup>Actor establishes a new state-of-the-art with an average success rate of 27.89% across all 8 tasks, achieving the highest success rate in most tasks. Specifically, on the open drawer and close drawer tasks, SD<sup>2</sup>Actor shows the most performance improvement compared to the other methods, indicating that SD<sup>2</sup>Actor can be applied in generalizing to arbitrary states of the continuous spectrum. A more in-depth analysis on all experiment indicates that the model achieves better performance when ground truth is provided in the first stage, suggesting its effectiveness in capturing the correspondence between states and actions. For example, in close drawer task, The model improves the performance by 12.16% after providing the ground truth in the first stage, while the Peract is only 4.73%. It shows that the generalization of the model mainly works in the second phase of grasping, *i.e.* controlling the object to reach the target state.

### 4.3. Qualitative Evaluations

In the test set setting, we show the four manipulation results of the model in the learned states. As shown in Figure 5, it

Num	Novel State	Any State
0	4.85	21.52
1	6.74	22.27
2	<b>12.58</b>	<b>27.89</b>
3	11.62	26.44

Table 5. **Ablation on the number of combination embeddings.**

indicates that the model is capable of accurately matching target state instructions with actual manipulation actions. In the zero-shot setting, as illustrated in Figure 6, our method demonstrates stable and precise manipulation action in the novel state. While only being trained on a few demonstrations, our method shows superior generalization capability to novel states. Subsequently, as evidenced in Figure 8 in the supplementary, we visualize the experimental results for 8 tasks in any state split. We observed that our model can learn a mapping from language instructions to precise goal states in a continuous spectrum.

Model	1		3		5	
	Novel	Any	Novel	Any	Novel	Any
PerAct [32]	1.29	5.24	2.14	7.30	4.10	19.97
3DA [20]	1.73	7.24	2.87	10.30	4.85	21.52
<b>SD<sup>2</sup>Actor</b>	<b>6.84</b>	<b>11.83</b>	<b>8.93</b>	<b>18.47</b>	<b>12.58</b>	<b>27.89</b>

Table 6. **Model Accuracy.** We evaluate the novel and any state success rate with different success ranges in comparison to PerAct and 3D Diffuser Actor(3DA). Number 1, 3, and 5 indicate that the range is 1/5, 3/5, and 1 times the original range, respectively.

### 4.4. Ablation Study & Model Analysis

**Effectiveness of different components.** We explore three key variations: *Fine-grained state representation*, *Orthogonalization loss* and *State decomposition and combination*. The embeddings preserves the state features of the manipulation tasks, which assist in generating accurate actions. The orthogonalization loss further facilitates the separation of embeddings, which we use to assess the effectiveness of linear combinations. The state decomposition and combination module aids the model in generalizing to new states.

As shown in Table 4, compared with SD<sup>2</sup>Actor, the performance of vanilla diffusion model dropped by 7.48% and 6.37%, which demonstrates the effectiveness of the state modeling. Most importantly, state decomposition module

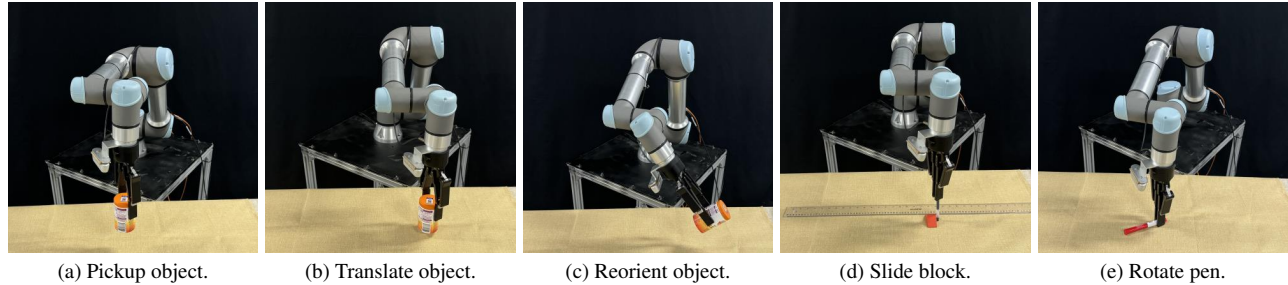


Figure 7. A presentation of 5 tasks in the real world.

is a critical part of generalization, which yields superior results. Lastly, we can analyze that the orthogonalization term enhances the embedding separability and improves the accuracy of the linear combination.

**Effectiveness of the number of linear combination.** We use the different numbers of embeddings as combination vectors to represent novel states for validating the effect of the number of embeddings on the generalization capability of the model. Since there are only 3 different states in  $B$  within each task in ARNOLD, the number of embeddings is also at most 3. The result is reported in Table 5, which shows that using 2 embeddings yields the best performance in the ARNOLD environment. When the number is set to 0,  $SD^2$  Actor degrades to the 3DA. We observe that the performance is already higher than the 3DA when using 1 closest embedding, indicating the importance of embeddings, which is consistent with the conclusions above.

**Model Accuracy.** We show the single-task novel and any state results by reducing the success range in ARNOLD and ultimate success rate in Table 6. The original success range of ARNOLD is in the supplementary material. The data indicates that the success rate improves as the range increases, demonstrating the accuracy of the proposed method. It is apparent that  $SD^2$  Actor demonstrates more stable inference results, with reduced performance degradation as the ranges decrease. Similar to the aforementioned experimental results, our approach exhibits higher performance on novel and any states, showing the relative accuracy of  $SD^2$  Actor even in the context of generalization.

Task	Train States	Test States	3DA	$SD^2$ Actor
Pickup	10, 20, 30 (cm)	0 ~ 30	7/20	<b>12/20</b>
Translate	10, 20, 30 (cm)	0 ~ 30	8/20	<b>14/20</b>
Reorient	10, 90, 135 ( $^\circ$ )	0 ~ 180	4/20	<b>9/20</b>
Slide	0, 70, 100 (%)	0 ~ 100	6/20	<b>13/20</b>
Rotate	0, 45, 180 ( $^\circ$ )	0 ~ 180	3/20	<b>10/20</b>

Table 7. **Real-world results.** A brief description of the collected data and single-task state generalization performance on 5 real-world tasks compared with 3D Diffuser Actor(3DA).

## 4.5. Real Robot Experiment

We further evaluate our model on five real-world tasks: pickup object (pickup), translate object (translate), reorient object (reorient), slide block (slide) and rotate pen (rotate). We illustrate these tasks in Figure 7, and the novel state is extracted offline. We employ a Universal Robot arm with a 2F-140 gripper and utilize a single Azure Kinect RGB-D sensor at a front view. The captured RGB-D images are downsampled from a resolution of  $2048 \times 1536$  to  $128 \times 128$ . We calibrate the robot camera extrinsic and transform point cloud to robot base frame.

For each task, we collect 20 demonstrations with three essential states for training, and subsequently uniformly sample 20 continuous states corresponding to the positions at other target locations in the space for inference. Each task is evaluated across 20 episodes. Table 7 reports success rates of  $SD^2$  Actor and 3D Diffuser Actor in real-world experiments. Results show our method is 30% higher than the baseline, which shows that  $SD^2$  Actor learns state generalization with noisy and limited real-world demonstrations.

## 5. Conclusion

This paper presents  $SD^2$  Actor, a diffusion policy that enables robots to perform novel states from language instructions by state decomposition. It employs learnable embeddings to learn continuous and comprehensible state representations directly from visual and language demonstrations. Through integrating hierarchical state decomposition and embeddings combination with conditional actions generation,  $SD^2$  Actor can comprehend and execute arbitrary-state action for robot manipulation. Extensive experiments on the continuous state manipulation benchmark demonstrate state-of-the-art performance, highlighting its effectiveness for state composition tasks and ability to generalize to any arbitrary state within a continuous spectrum. We envision our work will inspire future research on state generalization policies and feature fusion in robotics.

## References

- [1] Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. *NeurIPS*, 35:38546–38556, 2022. 2
- [2] Stanislaw Antol, C Lawrence Zitnick, and Devi Parikh. Zero-shot learning via visual abstraction. In *ECCV*, pages 401–416. Springer, 2014. 2
- [3] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *NeurIPS*, 31, 2018. 5
- [4] Stephen Boyd. Convex optimization. *Cambridge UP*, 2004. 5
- [5] 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. 2
- [6] Hao-Tien Lewis Chiang, Zhuo Xu, Zipeng Fu, Mithun George Jacob, Tingnan Zhang, Tsang-Wei Edward Lee, Wenhao Yu, Connor Schenck, David Rendleman, Dhruv Shah, et al. Mobility v1a: Multimodal instruction navigation with long-context vlms and topological graphs. *arXiv preprint arXiv:2407.07775*, 2024. 2
- [7] Yuhong Deng, Kai Mo, Chongkun Xia, and Xueqian Wang. Learning language-conditioned deformable object manipulation with graph dynamics. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7508–7514. IEEE, 2024. 1
- [8] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12462–12469. IEEE, 2024. 2
- [9] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *CVPR*, pages 10021–10030, 2023. 2
- [10] Irene Garcia-Camacho, Alberta Longhini, Michael Welle, Guillem Alenyà, Danica Kragic, and Júlia Borràs. Standardization of cloth objects and its relevance in robotic manipulation. *arXiv preprint arXiv:2403.04608*, 2024. 2
- [11] Ran Gong, Xiaofeng Gao, Qiaozi Gao, Suhaila Shakiah, Govind Thattai, and Gaurav S Sukhatme. Lemma: Learning language-conditioned multi-robot manipulation. *IEEE Robotics and Automation Letters*, 2023. 1
- [12] Ran Gong, Huang, et al. Arnold: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In *ICCV*, 2023. 1, 5
- [13] Lawrence W Green, Judith M Ottoson, Cesar Garcia, and Robert A Hiatt. Diffusion theory and knowledge dissemination, utilization, and integration in public health. *Annual review of public health*, 30(1):151–174, 2009. 2
- [14] Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. A survey on deep reinforcement learning algorithms for robotic manipulation. *Sensors*, 23(7):3762, 2023. 1
- [15] Ce Hao, Kelvin Lin, Siyuan Luo, and Harold Soh. Language-guided manipulation with diffusion policies and constrained inpainting. *arXiv preprint arXiv:2406.09767*, 2024. 1
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2
- [17] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022. 2
- [18] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2(3):6, 2022. 1
- [19] Chuhao Jin, Wenhui Tan, Jiange Yang, Bei Liu, Ruihua Song, Limin Wang, and Jianlong Fu. Alphablock: Embodied finetuning for vision-language reasoning in robot manipulation. *arXiv preprint arXiv:2305.18898*, 2023. 2
- [20] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 2, 4, 6, 7, 3
- [21] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [22] Kai Li, Martin Renqiang Min, and Yun Fu. Rethinking zero-shot learning: A conditional visual classification perspective. In *ICCV*, pages 3583–3592, 2019. 2
- [23] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *CVPR*, pages 18061–18070, 2024. 2
- [24] Siao Liu, Yang Liu, Linqiang Hu, Ziqing Zhou, Yi Xie, Zhile Zhao, Wei Li, and Zhongxue Gan. Diffskill: Improving reinforcement learning through diffusion-based skill denoiser for robotic manipulation. *Knowledge-Based Systems*, 300: 112190, 2024. 2
- [25] Xiao Ma, Sumit Patidar, Iain Houghton, and Stephen James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In *CVPR*, pages 18081–18090, 2024. 2
- [26] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *NeurIPS*, 35:462–477, 2022. 2
- [27] Javier R Movellan and James L McClelland. Learning continuous probability distributions with symmetric diffusion networks. *Cognitive Science*, 17(4):463–496, 1993. 2
- [28] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 1
- [29] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013. 2

- [30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2
- [31] Hyunwoo Ryu, Jiwoo Kim, Hyunseok An, Junwoo Chang, Joohwan Seo, Taehan Kim, Yubin Kim, Chaewon Hwang, Jongeun Choi, and Roberto Horowitz. Diffusion-edfs: Biequivariant denoising generative modeling on se (3) for visual robotic manipulation. In *CVPR*, pages 18007–18018, 2024. 2
- [32] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023. 2, 6, 7, 3, 4
- [33] Philip L Smith. Diffusion theory of decision making in continuous report. *Psychological Review*, 123(4):425, 2016. 2
- [34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2
- [35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2
- [36] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024. 2
- [37] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*, 2018. 2
- [38] Jake Varley, Sumeet Singh, Deepali Jain, Krzysztof Choromanski, Andy Zeng, Somnath Basu Roy Chowdhury, Avinava Dubey, and Vikas Sindhwani. Embodied ai with two arms: Zero-shot learning, safety and modularity. *arXiv preprint arXiv:2404.03570*, 2024. 2
- [39] Renhao Wang, Jiayuan Mao, Joy Hsu, Hang Zhao, Jiajun Wu, and Yang Gao. Programmatically grounded, compositionally generalizable robotic manipulation. *arXiv preprint arXiv:2304.13826*, 2023. 2, 4
- [40] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pre-training objective works best for zero-shot generalization? In *International Conference on Machine Learning*, pages 22964–22984. PMLR, 2022. 2
- [41] Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–37, 2019. 2
- [42] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024. 2
- [43] Zehang Weng, Haofei Lu, Danica Kragic, and Jens Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models. *arXiv preprint arXiv:2402.02989*, 2024. 2
- [44] 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. 2
- [45] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE TPAMI*, 41(9):2251–2265, 2018. 2
- [46] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023. 1, 2
- [47] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. *NeurIPS*, 35:4571–4584, 2022. 4
- [48] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024. 2
- [49] Lihan Zha, Yuchen Cui, Li-Heng Lin, Minae Kwon, Montserrat Gonzalez Arenas, Andy Zeng, Fei Xia, and Dorsa Sadigh. Distilling and retrieving generalizable knowledge for robot manipulation via language corrections. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15172–15179. IEEE, 2024. 1
- [50] Gengyu Zhang, Hao Tang, and Yan Yan. Versatile navigation under partial observability via value-guided diffusion policy. In *CVPR*, pages 17943–17951, 2024. 2
- [51] Tong Zhang, Yingdong Hu, Hanchen Cui, Hang Zhao, and Yang Gao. A universal semantic-geometric representation for robotic manipulation. *arXiv preprint arXiv:2306.10474*, 2023. 2
- [52] Zhengshen Zhang, Lei Zhou, Chenchen Liu, Zhiyang Liu, Chengran Yuan, Sheng Guo, Ruiteng Zhao, Marcelo H Ang Jr, and Francis EH Tay. Dexgrasp-diffusion: Diffusion-based unified functional grasp synthesis pipeline for multi-dexterous robotic hands. *arXiv preprint arXiv:2407.09899*, 2024. 2
- [53] Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Wang. VImbench: A compositional benchmark for vision-and-language manipulation. *NeurIPS*, 35:665–678, 2022. 6, 7, 3, 4
- [54] Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. Evaluating commonsense in pre-trained language models. In *AAAI*, pages 9733–9740, 2020. 2
- [55] Minjie Zhu, Yichen Zhu, Jinming Li, Junjie Wen, Zhiyuan Xu, Zhengping Che, Chaomin Shen, Yaxin Peng, Dong Liu, Feifei Feng, et al. Language-conditioned robotic manipulation with fast and slow thinking. *arXiv preprint arXiv:2401.04181*, 2024. 1