# Structural Action Transformer for 3D Dexterous Manipulation

Xiaohan Lei[1] Min Wang[2,*] Bohong Weng[1] Wengang Zhou[1,2,*] Houqiang Li[1,2]

[1]MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

{leixh, bhweng}@mail.ustc.edu.cn wangmin@iai.ustc.edu.cn {zhwg, lihq}@ustc.edu.cn

Project Page: https://xiaohanlei.github.io/projects/SAT

## Abstract

*Achieving human-level dexterity in robots via imitation learning from heterogeneous datasets is hindered by the challenge of cross-embodiment skill transfer, particularly for high-DoF robotic hands. Existing methods, often relying on 2D observations and temporal-centric action representation, struggle to capture 3D spatial relations and fail to handle embodiment heterogeneity. This paper proposes the Structural Action Transformer (SAT), a new 3D dexterous manipulation policy that challenges this paradigm by introducing a structural-centric perspective. We reframe each action chunk not as a temporal sequence, but as a variable-length, unordered sequence of joint-wise trajectories. This structural formulation allows a Transformer to natively handle heterogeneous embodiments, treating the joint count as a variable sequence length. To encode structural priors and resolve ambiguity, we introduce an Embodied Joint Codebook that embeds each joint's functional role and kinematic properties. Our model learns to generate these trajectories from 3D point clouds via a continuous-time flow matching objective. We validate our approach by pre-training on large-scale heterogeneous datasets and fine-tuning on simulation and real-world dexterous manipulation tasks. Our method consistently outperforms all baselines, demonstrating superior sample efficiency and effective cross-embodiment skill transfer. This structural-centric representation offers a new path toward scaling policies for high-DoF, heterogeneous manipulators.*

## 1. Introduction

The quest for robotic systems capable of human-level dexterity represents a grand challenge in Embodied Artificial Intelligence. Dexterous robotic hands, with their high degrees of freedom (DoF), hold the potential to perform a vast array of complex, contact-rich manipulation tasks currently beyond the reach of simpler parallel-jaw grippers. Imitation learning, particularly in its offline formulation, has emerged as a promising paradigm for teaching such skills by leveraging large-scale datasets of human demonstrations [8, 13, 15, 27, 28, 41, 45, 63, 66]. A critical bottleneck, however, lies in the cross-embodiment transfer of these skills: how can a robot effectively learn from heterogeneous demonstrations, given the significant differences in morphology, kinematics, and sensory feedback [9, 68, 75, 85]? While recent Vision-Language-Action (VLA) models have made strides in general-purpose robotics [29, 72, 79, 83, 87], they predominantly rely on 2D visual inputs, which often fail to capture the intricate 3D spatial relationships essential for precise dexterous manipulation [22, 31, 32, 51, 81]. This work tackles the challenge of cross-embodiment imita-

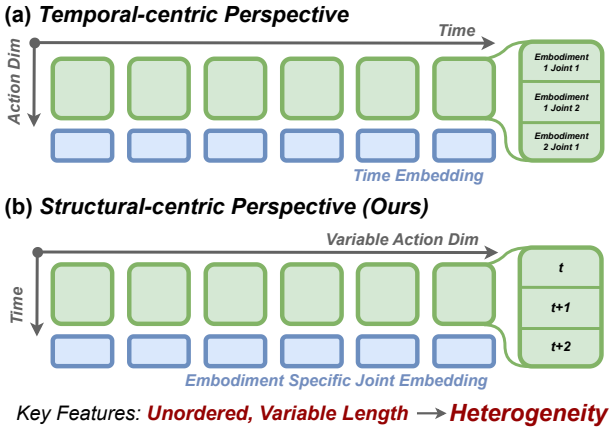


Figure 1. Conceptual illustration of action chunk tokenization. **(a)** The conventional temporal-centric perspective, which structures actions as a sequence of $T$ timesteps (chunk length), with each token having dimension $D_a$ (action dim). **(b)** Our proposed structural-centric perspective, which reframes the action chunk as a sequence of $D_a$ joints, where each token's feature is its temporal trajectory over $T$. This $(D_a, T)$ view naturally handles heterogeneous embodiments as a variable-length, unordered sequence, which is a key feature of our approach.

*Corresponding authors.

tion for dexterous hands by directly learning from 3D point cloud observations, proposing a fundamental shift in how we represent and process robotic actions.

A dominant paradigm in recent policy learning is action chunking, where a model predicts a sequence of future actions $(T, D_a)$ [12, 80]. As in Figure 1 (a), this **temporal-centric perspective**, which treats each $D_a$-dimensional vector as a token in a temporal sequence, is effective for low-dimensional systems. However, this representation faces a fundamental challenge when scaling to high-DoF, heterogeneous manipulators. As the action dimension $D_a$ grows (*e.g.*, from a 7-DoF robot arm to a 24-DoF dexterous hand), the model must learn complex, implicit correlations within a monolithic feature vector. More critically, this fixed-dimensional view provides no natural mechanism for cross-embodiment transfer, as different morphologies cannot be directly compared.

This paper challenges this conventional wisdom by reframing the action representation problem. We propose a **structural-centric perspective**, modeling actions as a sequence of $D_a$ embodied joints, where each joint's feature is its trajectory over the time horizon $T$, *i.e.*, $(D_a, T)$, as shown in Figure 1 (b). This structural view directly addresses the heterogeneity problem: the sequence length $D_a$ can now vary between embodiments, a property that Transformer architectures handle natively [65]. This allows a policy to learn transferable skills by finding functional similarities between corresponding joints. At the same time, this formulation also allows the model to learn a compressed representation of motion primitives for each joint, as time is now treated as the feature dimension.

We introduce the Structural Action Transformer (SAT), a novel 3D dexterous manipulation policy built upon this structural-centric representation. Our model takes raw 3D point cloud observations and natural language instructions as input. The 3D scene is processed by a hierarchical tokenizer that uses Farthest Point Sampling (FPS) and PointNets [50] to extract both local geometric tokens and a single global scene token. These are combined with language features from a T5 encoder [54] to form a multi-modal observation sequence. This sequence conditions a Diffusion Transformer (DiT) [47] that operates directly on our $(D_a, T)$ structural action representation. We treat the $D_a$ joints as a variable-length sequence, where each token represents a compressed temporal trajectory for a single joint. To explicitly encode structural priors and manage heterogeneity, we introduce an Embodied Joint Codebook. This codebook provides learnable embeddings for each joint based on its morphological properties, enabling the model to identify functional correspondences across different embodiments. The policy is trained to generate the entire action chunk by learning a conditional velocity field via a continuous-time flow matching objective [34], with the final action produced by an ODE solver.

We demonstrate the efficacy of this approach by pre-training on large-scale, heterogeneous datasets of both human and robot demonstrations [16, 18, 37, 46, 67] and then fine-tuning it on a suite of challenging simulation benchmarks [2, 10, 55] and real-world bimanual manipulation tasks. Our extensive experiments show that this structural-centric paradigm consistently outperforms strong baselines [12, 68, 74, 77, 82] and achieves superior sample efficiency, proving its effectiveness in cross-embodiment skill transfer. Fundamentally, our work is the first to successfully implement a policy that tokenizes actions along the structural dimension, offering a new and scalable path toward learning generalist policies for a diverse ecosystem of high-DoF, heterogeneous manipulators.

## 2. Related Work

### 2.1. Action Representation in Policy Learning

The representation of actions is a fundamental design choice in learning-based control. While early methods in imitation learning predict a single action at each timestep [49, 59], this autoregressive approach is known to suffer from compounding errors, where small inaccuracies accumulate over long horizons, leading to significant trajectory divergence [58].

To mitigate this issue, a dominant paradigm in recent literature is **action chunking**, where a policy outputs a sequence of future actions at each inference step [12, 61, 80]. This approach has been shown to improve temporal consistency and reduce the impact of compounding errors. For instance, Diffusion Policy [12] and related diffusion-based models [21, 25, 35, 36, 57, 70] generate chunked action trajectories by iteratively denoising a random tensor, producing smooth and effective behaviors. Similarly, transformer-based models [7, 29, 56, 80, 87] predict sequences of discretized action tokens. This temporal-centric view, structuring action chunks as a sequence of feature vectors over time, $(T, D_a)$, is now standard in large-scale robotic policies.

While effective, these methods treat the action vector at each timestep as a monolithic entity. This overlooks the rich kinematic structure of the robot and becomes inefficient as the action dimensionality $D_a$ grows. Our work challenges this conventional representation. Instead of viewing actions as a sequence of temporal snapshots, we propose to model them as a sequence of joint-wise trajectories, $(D_a, T)$. This reframing allows the model to learn compressed temporal primitives for each joint, and more importantly, provides a natural mechanism to handle embodiment heterogeneity.

### 2.2. Dexterous Manipulation

Dexterous robotic hands, with their high degrees of freedom, offer the potential to replicate human-level dexterity but pose significant learning challenges due to their complex dynamics and contact-rich nature [3]. Early successes often relied on precise analytical models [42], but recent progress has been dominated by learning-based methods.

Deep reinforcement learning (RL) has been successfully used to learn complex in-hand manipulation skills from scratch in simulation [1, 39, 43, 55], though transferring these policies to the real world remains a challenge. Imitation learning from human demonstrations offers a more data-efficient alternative. This includes learning from various interfaces such as vision-based motion capture [11, 52, 67, 76], or glove-based controllers [17, 73, 78].

Recent Vision-Language-Action (VLA) models have also been applied to this domain [4, 24, 71, 75]. For example, DexGraspVLA [84] and DexVLG [20] leverage diffusion-based or flow-based policies and large pre-trained models to achieve zero-shot grasping of novel objects. However, nearly all of these methods still rely on the conventional $(T, D_a)$ action representation. This approach treats the action as a monolithic, fixed-dimensional vector at each timestep, which is fundamentally ill-suited for cross-embodiment transfer as it provides no natural mechanism to align or compare manipulators with different kinematic structures or joint counts.

### 2.3. Heterogeneous Learning

A central goal of modern robotics is to create "generalist" agents that can operate across a wide range of tasks, environments, and robot morphologies [6]. This has led to a paradigm shift towards pre-training large policies on massive, heterogeneous datasets. The Open X-Embodiment dataset [45] represents a landmark effort in this direction, aggregating data from dozens of different robots.

Several strategies have emerged to handle the "heterogeneity" problem. Data-centric approaches aim to unify diverse datasets under a common data format and action representation. For instance, some methods [7, 29, 30, 45, 48, 69] discretize the action space into a shared vocabulary of tokens , while others [14, 19, 26, 33, 44, 62, 86, 87] leverage the power of pre-trained vision-language models (VLMs) to perform high-level reasoning, effectively treating robot control as a sequence modeling problem. This paradigm has been successfully scaled by training single, diffusion-based generalist policies on large-scale aggregated datasets [5, 23, 63]. Modular approaches explicitly design architectures to handle heterogeneity. A common technique is to introduce embodiment-specific "stems" that tokenize proprioceptive and visual inputs from different robots into a shared latent space, which is then processed by a large, pre-trained "trunk" [68, 82]

While these methods have achieved impressive generalization, they either enforce a unified action space that may not be optimal for all robots, or rely on separate modules to align inputs before the core policy network. Our work introduces a fundamentally different approach. By re-structuring the action prediction problem as generating a sequence of joint trajectories, $(D_a, T)$, we leverage a core property of the Transformer architecture: its ability to operate on variable-length sequences. In our framework, a robot's embodiment is defined by the length of the sequence, $D_a$. This allows a single, unified policy to naturally handle different robots and enables the self-attention mechanism to learn functional similarities and mappings between the joints of different embodiments directly within the representation space.

## 3. Method

### 3.1. Problem Formulation

We formulate the control problem for dexterous hands as a conditional generative modeling task. At each time step $t$, the policy receives an observation $o_t$ which consists of a history of the last $T_o$ raw 3D point clouds, $\mathcal{P}_t = (\mathbf{P}_{t-T_o+1}, \ldots, \mathbf{P}_t)$, where each $\mathbf{P}_k \in \mathbb{R}^{N\times 3}$ ($N$ is the number of points), and a natural language instruction $L$. The goal is to learn a policy $\pi$ that predicts a chunk of future actions $\mathbf{A}_t$. As motivated in our introduction, we challenge the conventional temporal-centric action representation $(T, D_a)$. Instead, we define the action chunk from a structural-centric perspective as $\mathbf{A}_t \in \mathbb{R}^{D_a \times T}$, where $D_a$ is the dimensionality of the robot's action space and $T$ is the prediction horizon. Each row $j_i \in \mathbb{R}^T$ of $\mathbf{A}_t$ represents the entire future trajectory for the $i$-th joint. The policy $\pi$ thus models the conditional distribution $p(\mathbf{A}_t|o_t)$. During closed-loop control, this action chunk $\mathbf{A}_t$ is generated in a receding-horizon manner. A subset of the predicted actions is executed, and the policy is re-queried with the new observation, enabling continuous feedback and correction.

### 3.2. Policy as a Conditional Normalizing Flow

We model the complex, high-dimensional conditional distribution $p(\mathbf{A}_t|o_t)$ using a continuous-time normalizing flow (CNF) [34]. We frame this as learning a conditional velocity field $v(\mathbf{A}_t^\tau, \tau, o_t)$ that transports a standard Gaussian $\mathcal{N}(0, I)$ to the action distribution. We parameterize the velocity field with a neural network $\epsilon_\theta(\mathbf{A}_t^\tau, \tau, o_t)$, where $\tau \in [0, 1]$ is the flow time. The network is trained to minimize the following objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau\sim\mathcal{U}(0,1),\mathbf{A}_t^0\sim\mathcal{N}(0,I),\mathbf{A}_t^1\sim\mathcal{D}}\left[\left\|\epsilon_\theta(\mathbf{A}_t^\tau, \tau, o_t) - (\mathbf{A}_t^1 - \mathbf{A}_t^0)\right\|^2\right], \tag{1}$$

where $\mathcal{D}$ is the dataset of ground-truth action chunks, $\mathbf{A}_t^1 \sim \mathcal{D}$ is a ground-truth action, $\mathbf{A}_t^0 \sim \mathcal{N}(0, I)$ is a noise sample, and $\mathbf{A}_t^\tau = (1-\tau)\mathbf{A}_t^0 + \tau\mathbf{A}_t^1$. The conditioning observation $o_t$ is the one associated with the target action $\mathbf{A}_t^1$.

At inference time, we sample a noise $\mathbf{A}_t^0 \sim \mathcal{N}(0, I)$ and generate the final action chunk $\mathbf{A}_t^1$ by solving the ordinary differential equation (ODE) $\frac{d\mathbf{A}_t^\tau}{d\tau} = \epsilon_\theta(\mathbf{A}_t^\tau, \tau, o_t)$ from $\tau = 0$ to $\tau = 1$. This is done with a numerical solver, such as a single Euler integration step, which recovers the probability flow in one function evaluation (1-NFE).

### 3.3. Policy Network Architecture

As shown in Figure 2, we design our velocity field model $\epsilon_\theta$ as a Transformer [65] architected to ingest our novel struc-

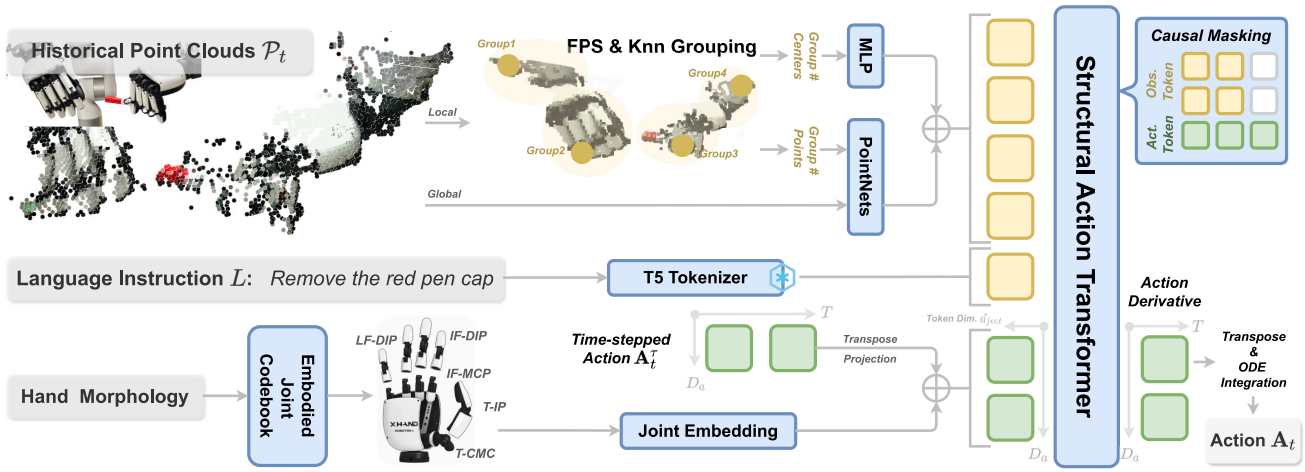


Figure 2. Our proposed model architecture. The policy takes a history of $T_o$ raw 3D point clouds $\mathcal{P}_t = (\mathbf{P}_{t-T_o+1}, \ldots, \mathbf{P}_t)$ and a language instruction $L$ as input. *Observation Tokenizer*: Each point cloud $\mathbf{P}_k$ in the history is processed via Farthest Point Sampling (FPS) and PointNets to extract local geometric tokens and a global scene context. The tokens from each time step are concatenated to form the final observation token sequence. Language is encoded by a T5 tokenizer [53]. *Structural Action Tokenizer*: Guided by the manipulator's morphology, the Embodied Joint Codebook produces structural-centric embeddings aligned with the action dimension $D_a$, which are added to the time-stepped noisy tokens $\mathbf{A}_t^\tau$. *Structural Action Transformer:* A DiT [47] with causal masking predicts the action velocity field. This field is then integrated via an ODE solver to produce the final action chunk $\mathbf{A}_t$.

tural action tokens and the multi-modal conditioning observation tokens. The architecture is divided into three components: an **Observation Tokenizer** that tokenizes the historical 3D point clouds and language inputs, a **Structural Action Tokenizer** that embed structural priors into action tokens, and a **Structural Action Transformer** that predicts the action velocity field conditioned on the observations.

### 3.3.1. Observation Tokenizer

Our hierarchical point cloud tokenizer is designed to capture both local geometric details and the global scene context from the entire observation history $\mathcal{P}_t$. We apply a shared encoding module to each of the $T_o$ point clouds $\mathbf{P}_k \in \mathcal{P}_t$. For a given $\mathbf{P}_k$, we first use Farthest Point Sampling (FPS) to select $M$ local group centers $C_k = \{c_1, ..., c_M\} \subset \mathbf{P}_k$. For each center $c_i$, we query its $K$ nearest neighboring points to form a local group $\mathbf{P}_{k,i} \subset \mathbf{P}_k$. Each local group $\mathbf{P}_{k,i}$ is processed by a shared PointNet [50] to extract a local geometric feature $f_{k,i} \in \mathbb{R}^{d_{feat}/T_o}$. The 3D coordinates of the group centers $C_k$ are passed through an MLP to create positional embeddings $p_{k,i} \in \mathbb{R}^{d_{feat}/T_o}$. These are combined ($f'_{k,i} = f_{k,i} + p_{k,i}$) as $M$ local point tokens, $tok_{l,k} \in \mathbb{R}^{M \times d_{feat}/T_o}$. To leverage the permutation-invariant nature of these local patches, we apply a random shuffle to the $tok_{l,k}$ sequence during training as data augmentation. In parallel, to capture holistic scene understanding, we feed the entire raw point cloud $\mathbf{P}_k$ into a separate PointNet-based encoder [50]. This module produces a single global token, $tok_{g,k} \in \mathbb{R}^{d_{feat}/T_o}$, which represents the overall scene context for that timestep. This global token does not receive any positional embedding.

This process yields $T_o$ global tokens and $T_o$ sets of $M$ local tokens. These are concatenated to form the full point cloud history sequence:

$$tok_{hist} = \text{Cat}\big(\text{Cat}(tok_{g,\,t-T_o+1}, \ldots, tok_{g,\,t}), \\ \text{Cat}(tok_{l,\,t-T_o+1}, \ldots, tok_{l,\,t})\big) \in \mathbb{R}^{(1+M)\times d_{feat}}, \tag{2}$$

where Cat means concatenation. The language instruction $L$ is tokenized and encoded using a pre-trained T5 encoder [53] to produce a sequence of language tokens $tok_{lang} \in \mathbb{R}^{L_{lang} \times d_{feat}}$. The final observation tokens are formed by concatenating the historical tokens and the language tokens: $tok_{obs} = \text{Cat}(tok_{hist}, tok_{lang})$. This sequence serves as the conditioning prefix for the model.

### 3.3.2. Structural Action Tokenizer

The noisy action $\mathbf{A}_t^\tau \in \mathbb{R}^{D_a \times T}$ is treated as a sequence of $D_a$ tokens, where each token represents the entire temporal trajectory for a single joint. This high-dimensional temporal vector (dim $T$) contains significant redundancy. We compress it by passing each of the $D_a$ joint trajectories through a shared MLP, projecting it from $T$ dimensions to a lower-dimensional embedding $d_{feat}$ (*e.g.*, from 64 to 16). This results in an embedded action sequence $tok_{act} \in \mathbb{R}^{D_a \times d_{feat}}$.

To resolve the ambiguity between the $D_a$ joint tokens and explicitly embed structural priors, we introduce an **Embodied Joint Codebook**. This codebook is derived from the manipulator's morphology. For any given hand, we define each joint $j$ as a three-part triplet $J_j = (e, f, r)$:

- $e \in \mathbb{Z}$: The **Embodiment ID**, a unique identifier for the manipulator (*e.g.*, ShadowHand, XHand).
- $f \in \mathbb{Z}$: The **Functional Category**. Inspired by human hand anatomy, we classify joints by their functional role, such as Carpometacarpal (CMC), Metacarpophalangeal (MCP), Proximal Interphalangeal (PIP), or Distal Interphalangeal (DIP) joints.
- $r \in \mathbb{Z}$: The **Rotation Axis**, describing the joint's primary

motion, *e.g.*, Flexion/Extension, Abduction/Adduction, or Pronation/Supination.

The complete mapping table covering embodiment, function category and rotation axis used in our experiments is provided in Appendix. Each element in the triplet $(e, f, r)$ indexes a separate learnable embedding table. The final codebook embedding $C_j \in \mathbb{R}^{d_{feat}}$ for joint $j$, is the sum of its three component embeddings. This design is central to handling heterogeneity: two different hands (different $e$) may share the same functional joint (same $f$) and rotation (same $r$), resulting in similar codebook embeddings that prime the model for transfer learning. The final input sequence for the action tokens is formed by adding the codebook embeddings to the compressed action trajectories: $tok_{input\_act} = tok_{act} + \mathbf{E}$, where $\mathbf{E} \in \mathbb{R}^{D_a \times d_{feat}}$ is the matrix of codebook embeddings for the manipulator.

#### 3.3.3. Structural Action Transformer

The $tok_{input\_act}$ sequence is then concatenated with the observation tokens $tok_{obs}$. This combined sequence is fed into the DiT [47]. We modify the DiT's self-attention mask to enforce causal masking, where $tok_{obs}$ only attends to other observation tokens, and $tok_{input\_act}$ attends to all observation tokens and all other action tokens. The output tokens from the DiT corresponding to the action sequence are passed through a final MLP to produce the predicted action velocity field $\epsilon_\theta(\mathbf{A}_t^\tau, \tau, o_t)$.

## 4. Experiments

### 4.1. Datasets and Experimental Setup

**Offline Pre-training Datasets.** Our approach leverages large-scale, heterogeneous datasets for pre-training. The full dataset is a mixture from three distinct sources, as detailed in Figure 3. For **human demonstrations**, we use HOI4D [37], Ego-Exo4D [18] and Aria Digital Twin (ADT) [46], which include large-scale 4D egocentric dataset of human-object interactions. We leverage its 3D point cloud observations and corresponding MANO hand poses. To integrate this data, we process the MANO model parameters to generate our structural action representation $\mathbf{A} \in \mathbb{R}^{D_a \times T}$ and tag each joint according to our Embodied Joint Codebook (same for Robot and Simulation). For **robot demonstrations**, we incorporate 3D dexterous manipulation data from existing robot datasets, including Fourier ActionNet [16] and DexCap [67], which provide diverse 3D interaction trajectories. For the **simulation domains**, we collect expert demonstrations using policies trained via reinforcement learning (RL), such as VRL3-trained policies [55] for Adroit [55] and PPO-trained policies [60] for DexArt [2] and Bi-DexHands [10]. We filter these datasets to retain only successful trajectories.

**Implementation Details.** We pre-train our model using the AdamW optimizer [38] with hyperparameters $\beta = (0.9, 0.999)$, $\epsilon = 1 \times 10^{-8}$ and weight decay 0.01. We use a peak learning rate of $1 \times 10^{-4}$, scheduled with a linear warmup of 10,000 steps followed by a cosine decay down to $1 \times 10^{-6}$. After the pre-training phase, the model is fine-tuned for each downstream simulation task using a small set of in-domain demonstrations with a fine-tuning learning rate of $1 \times 10^{-5}$. During inference, we generate the action chunk by solving the flow matching ODE. We employ Euler integration with a fixed step size of 10 to map the noise vector to a clean action.

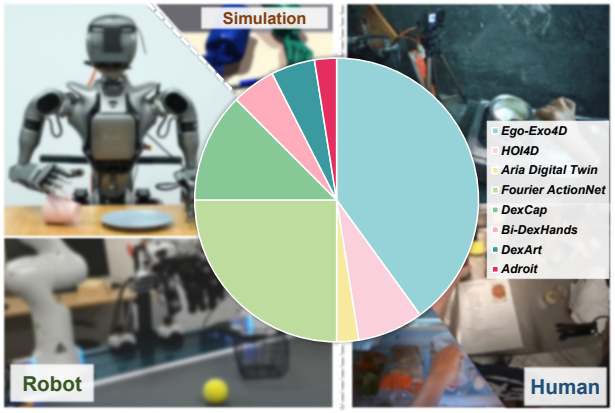


Figure 3. Composition of the offline pre-training dataset. The pie chart illustrates the relative data scale of each of the constituent datasets [2, 10, 16, 18, 37, 46, 55, 67].

### 4.2. Comparison with Baselines

**Benchmarks and Simulation Environments.** We evaluate our method by fine-tuning the pre-trained model on three challenging simulation benchmarks, totaling 11 tasks. Adroit [55] is a suite of difficult dexterous manipulation tasks using the Shadow Hand. DexArt [2] focuses on manipulating complex articulated objects. Bi-DexHands [10] features a bimanual dexterous manipulation benchmark. These benchmarks are built upon high-fidelity physics simulators, such as MuJoCo [64] and IsaacGym [40], which provide the realistic contact dynamics necessary for evaluating dexterous control.

**Baseline Methods.** We compare our method with several strong baselines, as summarized in Table 1. These include Diffusion Policy (DP) [12] and HPT [68], which are state-of-the-art policies that primarily rely on 2D images; UniAct [82], a framework based on universal action tokens; and recent 3D-based approaches such as 3D Diffusion Policy (3DDP) [77] and 3D ManiFlow Policy [74], which directly use 3D point cloud inputs similar to ours. For all baselines, we use their official, publicly available implementations to ensure a fair comparison. We present the results of 3D realizations of the 2D baselines in the Appendix.

**Key Findings.** The results across 11 tasks are shown in Table 1. Our method consistently outperforms all 2D and 3D baselines across the 11 simulation tasks. This superior performance is achieved with a remarkably efficient model. As shown in Table 1, our SAT has only **19.36M** parameters excluding the T5 tokenizer, which is an order of magnitude

| Method | Params (M) | Modality | Adroit (3) [55] | DexArt (4) [2] | Bi-DexHands (4) [10] | Average Success |
|---|---|---|---|---|---|---|
| Diffusion Policy [12] | 266.8 | 2D | $0.32_{\pm 0.03}$ | $0.49_{\pm 0.04}$ | $0.42_{\pm 0.05}$ | $0.42_{\pm 0.04}$ |
| HPT [68] | 13.99 | 2D | $0.45_{\pm 0.02}$ | $0.53_{\pm 0.05}$ | $0.44_{\pm 0.04}$ | $0.47_{\pm 0.04}$ |
| UniAct [82] | 1053 | 2D | $0.49_{\pm 0.01}$ | $0.55_{\pm 0.03}$ | $0.47_{\pm 0.07}$ | $0.50_{\pm 0.05}$ |
| 3D Diffusion Policy [77] | 255.2 | 3D | $0.68_{\pm 0.03}$ | $0.69_{\pm 0.02}$ | $0.55_{\pm 0.14}$ | $0.63_{\pm 0.06}$ |
| 3D ManiFlow Policy [74] | 218.9 | 3D | $0.70_{\pm 0.02}$ | $0.70_{\pm 0.03}$ | $0.59_{\pm 0.07}$ | $0.66_{\pm 0.04}$ |
| **SAT (Ours)** | 19.36 | 3D | $\mathbf{0.75_{\pm 0.02}}$ | $\mathbf{0.73_{\pm 0.03}}$ | $\mathbf{0.67_{\pm 0.05}}$ | $\mathbf{0.71_{\pm 0.04}}$ |

Table 1. Quantitative comparison of our method against 2D (image-based) and 3D (point cloud-based) baselines on 11 dexterous manipulation tasks from the Adroit [55], DexArt [2], and Bi-DexHands [10] simulation benchmarks.

| Token Dim. | Prams. (M) | 1-NFE FLOPs (G) | Success |
|---|---|---|---|
| 16 | 4.71 | 0.66 | $0.66_{\pm 0.02}$ |
| 32 | 8.65 | 0.77 | $\mathbf{0.71_{\pm 0.05}}$ |
| 64 | 19.36 | 0.99 | $\mathbf{0.71_{\pm 0.04}}$ |
| 128 | 52.08 | 1.65 | $\mathbf{0.71_{\pm 0.03}}$ |
| 256 | 162.74 | 3.63 | $0.70_{\pm 0.04}$ |

Table 2. Ablation on temporal compression and token dimension. We vary the token dimension ($d_{feat}$), which is the embedding size after compressing the temporal trajectory ($T$) of each joint.

| Human | Robot | Simulation | Scale | Success |
|---|---|---|---|---|
| ✓ | ✓ | ✓ | 100% | $\mathbf{0.71_{\pm 0.04}}$ |
| ✓ | ✗ | ✗ | 100% | $0.68_{\pm 0.04}$ |
| ✗ | ✓ | ✗ | 100% | $0.66_{\pm 0.05}$ |
| ✗ | ✗ | ✓ | 100% | $0.70_{\pm 0.03}$ |
| ✓ | ✓ | ✓ | 10% | $0.68_{\pm 0.03}$ |

Table 3. Pre-training data ablation. We report the average success on simulation tasks after fine-tuning models pre-trained on different combinations of Human, Robot, and Simulation datasets.

smaller than 2D baselines, and significantly more compact than other 3D-based methods. This highlights that our proposed structural-centric representation is not only effective for dexterous 3D manipulation but also parameter-efficient.

### 4.3. Ablation Studies

**Temporal Dimension Compression.** In Table 2, we ablate the **Token Dim.** $d_{feat}$. This dimension impacts both parameters and FLOPs. We observe that performance is robust to compression, only dropping slightly when the dimension is reduced to 16. This slight degradation suggests that at extremely high compression ratios, critical information begins to be lost. Nonetheless, the policy's strong resilience to compression down to 32 dimensions indicates that the temporal trajectories contain significant redundancy that can be compressed without substantial loss of performance.

**Pre-training Data Composition.** Table 3 analyzes the impact of different pre-training data combinations. While the full mixture yields the best result, we find it very interesting that pre-training on Human data alone outperforms pre-training on Robot data. This suggests our Embodied Joint Codebook successfully translates functional human motion to robotic control. Conversely, pre-training solely on simulation data outperforms both Human-only and Robot-only pre-training. This is expected, as the simulation data is directly aligned with the downstream fine-tuning tasks in terms of observation modality, action space, and environment dynamics.

**Model Component Ablation.** In Table 4, we ablate key architectural components. While removing observation features (global or local) or the causal mask leads to a moderate performance drop, we observe a more significant degradation when reverting to a conventional temporal-centric representation (the last row). This result validates our core hypothesis that the structural-centric formulation is a more effective paradigm for these high-DoF manipulation tasks. Besides, removing the joint embedding causes catastrophic failure. This is expected: our action sequence $\mathbf{A} \in \mathbb{R}^{D_a \times T}$ is inherently unordered. Without the Embodied Joint Codebook, the Transformer has no mechanism to determine which trajectory corresponds to which physical joint, making the learning task impossible.

**Few-shot Adaptation Efficiency.** To evaluate the quality of the learned prior, we compare its few-shot adaptation efficiency against UniAct [82] in Figure 4. The results clearly show that our method not only achieves a higher final success rate across all settings but also learns significantly faster, especially in low-data, few-shot regimes.

**Embodied Joint Codebook Analysis.** We further dissect the Embodied Joint Codebook in Table 5. While removing the Embodiment ID or Rotation Axis leads to a performance drop, ablating the Functional Category causes a catastrophic failure. This indicates that functional correspondence is the most critical factor for the policy to bridge the embodiment gap. To understand why this works, we analyze 10 common dexterous manipulators to inform our codebook design, including the Shadow Dexterous Hand, Allegro Hand, Inspire Robots Dexterous Hand, *etc*. In Figure 5, we plot the frequency of each joint type from our codebook across these hands. We find that Flexion/Extension joints for the MCP, CMC and PIP joints are the most frequent. This suggests these joints represent a core functional set that is essential for dexterous manipulation and should be prioritized in hardware design. We also visualize the learned codebook embedding space from these hands using t-SNE in Figure 6. The visualization illustrates clear, dominant clustering by Embodiment (a) and Rotation Axis (c). Functional Category (b) clusters are less distinct, as the model learns a highly specific embedding for each joint to resolve structural ambiguity. This suggests that generalization stems not from embedding similarity, but from the codebook's compositional structure, which allows the model to learn trans-

| Model Variant | Success |
|---|---|
| SAT | **0.71**±0.04 |
| SAT w.o. global point cloud token | 0.68±0.05 |
| SAT w.o. local point cloud token | 0.69±0.03 |
| SAT w.o. causal mask | 0.68±0.04 |
| SAT w.o. joint embedding | 0.01±0.01 |
| SAT w. temporal-centric action | 0.64±0.05 |

Table 4. Model component ablation. We analyze the impact of removing key architectural components on the average success rate.

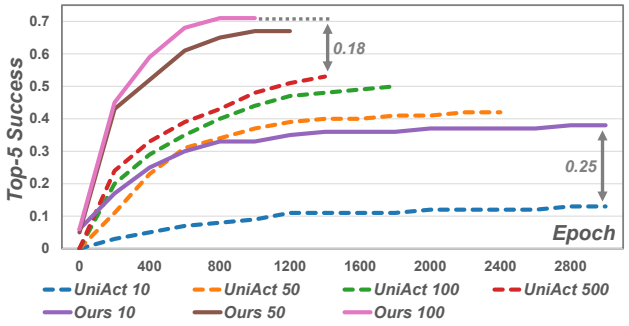


Figure 4. Few-shot adaptation efficiency. We plot the average success rate versus training epochs for our method and the UniAct [82] baseline, evaluated in few-shot settings using varying numbers of in-domain demonstrations.

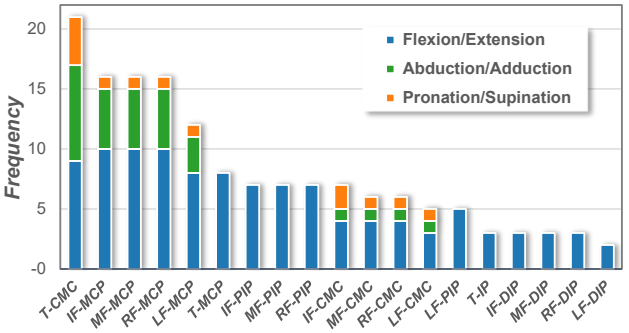


Figure 5. Frequency analysis of joint types in our Embodied Joint Codebook, derived from a survey of 10 common dexterous hands.

ferable functional patterns.

### 4.4. Real-World Experiments

**Hardware and Teleoperation Setup.** Our real-world bimanual system, shown in Figure 8 (a), is comprised of two 7-DoF xArm robotic arms, each equipped with a 12-DoF xHand dexterous hand. We use a single L515 LiDAR camera mounted above the workspace for 3D scene perception, providing point cloud observations. The set of manipulated objects, seen in Figure 8 (b), includes a pen with a cap, a Baymax toy, a cardboard box, a plastic plate, a cup with a brush, and a basketball.

**Data Collection via Teleoperation.** We collect demonstration data using a Meta Quest 3 VR headset, which captures the operator's hand and finger motions. To transfer the human motion to the robot's dissimilar kinematics, we employ a real-time retargeting strategy from AnyTele [52]. Specifically, our teleoperation system reads the 3D poses of the operator's hand links from the VR controller. It then optimizes the xHand's joint angles by minimizing an objective function defined as the difference between the wrist-to-fingertip vectors on the human hand and the corresponding vectors on the robot hand. This method allows for intuitive control and the collection of dexterous demonstration data.

| Embodiment | Function | Rotation | Success (11 tasks) |
|---|---|---|---|
| ✓ | ✓ | ✓ | **0.71**±0.04 |
| ✗ | ✓ | ✓ | 0.57±0.05 |
| ✓ | ✗ | ✓ | 0.02±0.02 |
| ✓ | ✓ | ✗ | 0.62±0.07 |

Table 5. Ablation study on the components of the Embodied Joint Codebook. We report the average success rate after fine-tuning models with different codebook components ablated.

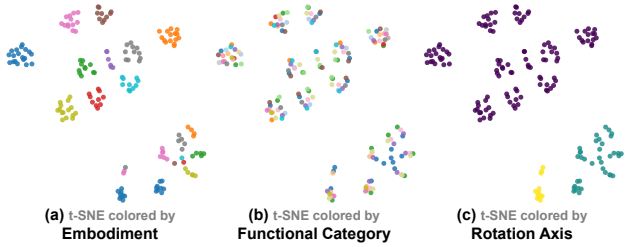


Figure 6. T-SNE visualization of the learned Embodied Joint Codebook embeddings. These embeddings, derived from 10 dexterous manipulators, are colored by **(a)** Embodiment ID, **(b)** Functional Category, and **(c)** Rotation Axis.

**Task Suite and Experimental Setup.** We evaluate performance on 6 challenging tasks, which require both single-arm precision and complex bimanual coordination.

- **Remove the pen cap:** A bimanual task where one hand must stabilize the pen body while the other grasps and removes the cap (Figure 7 (a)).
- **Hand over Baymax:** A coordinated bimanual task where the left hand picks up the toy and passes it to the right hand (Figure 7 (b)).
- **Push then grab box:** A long-horizon, multi-stage bimanual task. The left hand first pushes the box to the edge of a platform, after which the right hand grasps it from the table (Figure 7 (c)).
- **Place block in plate:** A single-arm task testing grasping and placement of a block into a target container.
- **Brush the cup:** A contact-rich bimanual task where one hand holds the cup, and the other uses a brush to perform a cleaning motion inside it.
- **Grasp basketball:** A bimanual task requiring the two hands to cooperatively form a stable, large-volume grasp on the ball.

Qualitative rollouts for the remaining tasks are included in the Appendix.

**Experimental Setup.** For each of the 6 tasks, we collect a dataset of 50 successful demonstrations. We compare three methods: HPT [68] uses its publicly available pre-trained weights. We train task-specific input stems and heads for each task. 3DDP [77] is trained from scratch for each task. SAT uses the pre-trained model. We then fine-tune this

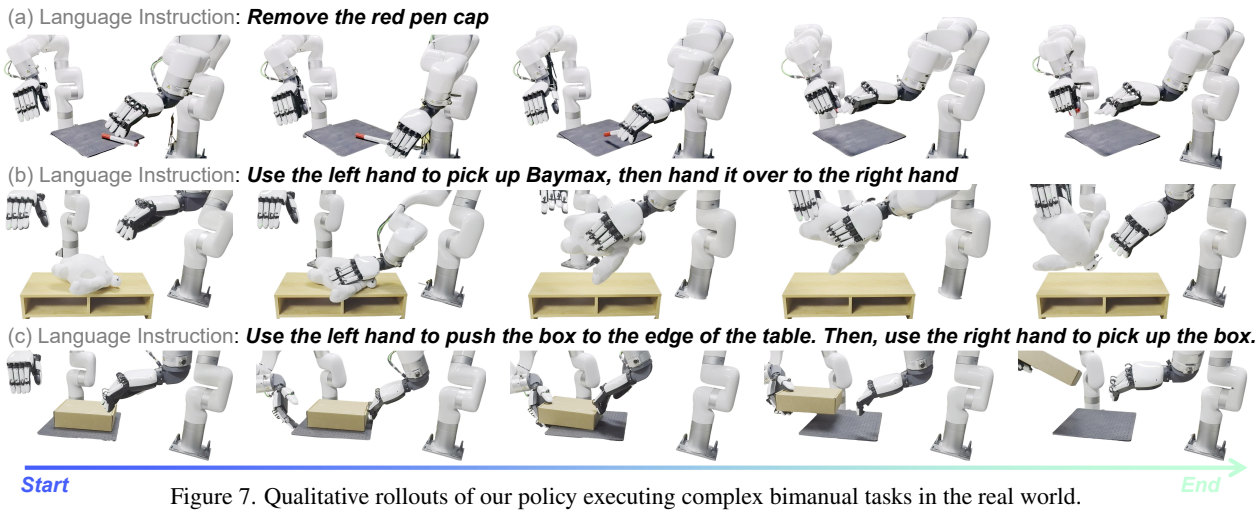


Figure 7. Qualitative rollouts of our policy executing complex bimanual tasks in the real world.

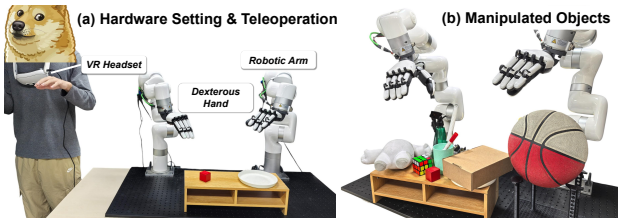


Figure 8. Real-world experimental setup. **(a)** Our bimanual hardware setup. We collect demonstration data using a VR headset for teleoperation. **(b)** The set of diverse objects used in our real-world manipulation, requiring both precision and bimanual coordination.

| Task Description | Success Rate HPT [68] | 3DDP [77] | **SAT (Ours)** |
|---|---|---|---|
| *Remove the pen cap* | 0.10 | 0.25 | **0.30** |
| *Hand over Baymax* | 0.50 | 0.75 | **0.85** |
| *Push then grab box* | 0.05 | 0.15 | **0.35** |
| *Place block in plate* | 0.60 | 0.85 | **0.90** |
| *Brush the cup* | 0.10 | 0.30 | **0.45** |
| *Grasp basketball* | 0.65 | 0.80 | **0.95** |

Table 6. Quantitative results on 6 real-world bimanual manipulation tasks. We compare our model against baselines using in-domain demonstrations.

model on the demonstrations, evaluating its ability to adapt its structural priors to the real-world domain.

**Real-World Results.** The quantitative results are summarized in Table 6. Our method achieves higher success rates than all baselines across 6 tasks. This suggests that our structural action representation pre-trained on diverse data, provides a powerful prior for learning complex bimanual coordination. Figure 7 shows qualitative rollouts of our policy executing three of the bimanual tasks.

### 4.5. Failure Case Analysis and Limitations

In addition to the quantitative ablations, we analyze failure modes which highlight the limitations of our current approach and provide avenues for future work. In complex, contact-rich bimanual tasks, one hand may severely occlude the other hand or the target object. A single, fixed camera provides no mechanism to resolve this ambiguity, starving the policy of the critical geometric information needed for precise coordination. We also observe large kinematic or contact-geometry mismatches between demonstrations and the execution platform produce action-assignment errors. These failures require explicit dynamics/force constraints or tactile closed-loop correction to resolve. We provide qualitative visualizations of these failure modes in the Appendix.

## 5. Conclusion

In this work, we introduce a fundamental shift in action representation for policy learning, challenging the conventional temporal-centric paradigm. We propose a *structural-centric* representation, which models an action chunk as a variable-length sequence of joint-wise trajectories. We demonstrate that this structural formulation provides a powerful framework for tackling cross-embodiment skill transfer in high-DoF dexterous hands. Through comprehensive evaluation across both simulation and real-world bimanual manipulation tasks, our method consistently outperforms all baselines. Our results highlight that a structural-centric prior enables temporal compression and allows the policy to bridge the morphological gap between demonstrators and robots. In future research, this structural action representation will be extended beyond imitation learning, investigating its potential as an expressive policy class for reinforcement learning, where it could offer a structured exploration space for complex high-DoF agents.

## 6. Acknowledgments

This work was supported by the National Natural Science Foundation of China under Contract 62472141, the Natural Science Foundation of Anhui Province under Contract 2508085Y040, and the Youth Innovation Promotion Association CAS. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC and the Supercomputing Center of USTC.

## References


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

[2] Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21190–21200, 2023. 2, 5, 6

[3] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 348–353. IEEE, 2000. 2

[4] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 3

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 3

[6] Rishi Bommasani. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 3

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 2, 3

[8] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 1

[9] Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. Vidbot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27661–27672, 2025. 1

[10] Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2804–2818, 2023. 2, 5, 6

[11] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024. 3

[12] 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, 5, 6

[13] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019. 1

[14] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. 2023. 3

[15] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023. 1

[16] Yao Mu Fourier ActionNet Team. Actionnet: A dataset for dexterous bimanual manipulation. 2025. 2, 5

[17] Yuyang Gao, Haofei Ma, and Pai Zheng. Glovity: Learning dexterous contact-rich manipulation via spatial wrench feedback teleoperation system. *arXiv preprint arXiv:2510.09229*, 2025. 3

[18] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19383–19400, 2024. 2, 5

[19] ByungOk Han, Jaehong Kim, and Jinhyeok Jang. A dual process vla: Efficient robotic manipulation leveraging vlm. *arXiv preprint arXiv:2410.15549*, 2024. 3

[20] Jiawei He, Danshi Li, Xinqiang Yu, Zekun Qi, Wenyao Zhang, Jiayi Chen, Zhaoxiang Zhang, Zhizheng Zhang, Li Yi, and He Wang. Dexvlg: Dexterous vision-language-grasp model at scale. *arXiv preprint arXiv:2507.02747*, 2025. 3

[21] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, et al. Dita: Scaling diffusion transformer for generalist vision-language-action policy. *arXiv preprint arXiv:2503.19757*, 2025. 2

[22] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 1

[23] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. Pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. 3

[24] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025. 3

[25] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 2

[26] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1724–1734, 2025. 3

[27] Shuo Jiang, Haonan Li, Ruochen Ren, Yanmin Zhou, Zhipeng Wang, and Bin He. Kaiwu: A multimodal manipulation dataset and framework for robot learning and human-robot interaction. *arXiv preprint arXiv:2503.05231*, 2025. 1

[28] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 1

[29] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1, 2, 3

[30] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024. 3

[31] Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *arXiv preprint arXiv:2503.07511*, 2025. 1

[32] Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *arXiv preprint arXiv:2506.07961*, 2025. 1

[33] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18061–18070, 2024. 3

[34] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2, 3

[35] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025. 2

[36] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024. 2

[37] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21022, 2022. 2, 5

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[39] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105): eads5033, 2025. 3

[40] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 5

[41] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 879–893. PMLR, 2018. 1

[42] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017. 2

[43] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 1101–1112. PMLR, 2020. 3

[44] 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. 3

[45] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. 1, 3

[46] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20133–20143, 2023. 2, 5

[47] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4195–4205, 2023. 2, 4, 5

[48] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. 3

[49] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 1, 1988. 2

[50] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 2, 4

[51] Shengyi Qian, Kaichun Mo, Valts Blukis, David F Fouhey, Dieter Fox, and Ankit Goyal. 3d-mvp: 3d multiview pretraining for manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22530–22539, 2025. 1

[52] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter

Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. *arXiv preprint arXiv:2307.04577*, 2023. 3, 7

[53] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 4

[54] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 2

[55] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 2, 3, 5, 6

[56] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 2

[57] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023. 2

[58] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (ICAIS)*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010. 2

[59] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (ICAIS)*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 2

[60] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 5

[61] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019. 2

[62] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. 3

[63] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 1, 3

[64] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5026–5033. IEEE, 2012. 5

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 2, 3

[66] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 1723–1736. PMLR, 2023. 1

[67] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024. 2, 3, 5

[68] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 37: 124420–124450, 2024. 1, 2, 3, 5, 6, 7, 8

[69] Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. *arXiv preprint arXiv:2507.01016*, 2025. 3

[70] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Generalizable and interpretable robot foundation model via self-generated reasoning. *arXiv preprint arXiv:2412.03293*, 2024. 2

[71] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025. 3

[72] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025. 1

[73] Mengda Xu, Han Zhang, Yifan Hou, Zhenjia Xu, Linxi Fan, Manuela Veloso, and Shuran Song. Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation. *arXiv preprint arXiv:2505.21864*, 2025. 3

[74] Ge Yan, Jiyue Zhu, Yuquan Deng, Shiqi Yang, Ri-Zhao Qiu, Xuxin Cheng, Marius Memmel, Ranjay Krishna, Ankit Goyal, Xiaolong Wang, and Dieter Fox. ManiFlow: A general robot manipulation policy via consistency flow training. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2025. 2, 5, 6

[75] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025. 1, 3

[76] Shiqi Yang. Ace: A cross-platform visual-exoskeleton system for low-cost dexterous teleoperation. Master's thesis, University of California, San Diego, 2025. 3

[77] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d repre-

sentations. *arXiv preprint arXiv:2403.03954*, 2024. 2, 5, 6, 7, 8

[78] Han Zhang, Songbo Hu, Zhecheng Yuan, and Huazhe Xu. Doglove: Dexterous manipulation with a low-cost open-source haptic force feedback glove. *arXiv preprint arXiv:2502.07730*, 2025. 3

[79] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1702–1713, 2025. 1

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

[81] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 1

[82] Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22508–22519, 2025. 2, 3, 5, 6, 7

[83] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 1

[84] Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Zhang Chen, Tianrui Guan, Fanlian Zeng, Ka Num Lui, Yuyao Ye, Yitao Liang, et al. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. *arXiv preprint arXiv:2502.20900*, 2025. 3

[85] Jiaming Zhou, Teli Ma, Kun-Yu Lin, Zifan Wang, Ronghe Qiu, and Junwei Liang. Mitigating the human-robot domain discrepancy in visual pre-training for robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22551–22561, 2025. 1

[86] Weijie Zhou, Manli Tao, Chaoyang Zhao, Haiyun Guo, Honghui Dong, Ming Tang, and Jinqiao Wang. Physvlm: Enabling visual language models to understand robotic physical reachability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6940–6949, 2025. 3

[87] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 2165–2183, 2023. 1, 2, 3