This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Tra-MoE: Learning Trajectory Prediction Model from Multiple Domains for Adaptive Policy Conditioning

Jiange Yang [1,2] Haoyi Zhu[2,3] Yating Wang[2,4] Gangshan Wu[1] Tong He[2] Limin Wang[1,2*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Shanghai AI Laboratory, [3]University of Science and Technology of China, [4]Tongji University
jiangeyang.jgy@gmail.com, {gswu,lmwang}@nju.edu.cn,
{zhuhaoyi, wangyating, hetong}@pjlab.org.cn
https://github.com/MCG-NJU/Tra-MoE

Abstract

*Learning from multiple domains is a primary factor that influences the generalization of a single unified robot system. In this paper, we aim to learn the trajectory prediction model by using broad out-of-domain data to improve its performance and generalization ability. Trajectory model is designed to predict any-point trajectories in the current frame given an instruction and can provide detailed control guidance for robotic policy learning. To handle the diverse out-of-domain data distribution, we propose a sparsely-gated MoE (**Top-1** gating strategy) architecture for trajectory model, coined as **Tra-MoE**. The sparse activation design enables good balance between parameter cooperation and specialization, effectively benefiting from large-scale out-of-domain data while maintaining constant FLOPs per token. In addition, we further introduce an adaptive policy conditioning technique by learning 2D mask representations for predicted trajectories, which is explicitly aligned with image observations to guide action prediction more flexibly. We perform extensive experiments on both simulation and real-world scenarios to verify the effectiveness of Tra-MoE and adaptive policy conditioning technique. We also conduct a comprehensive empirical study to train Tra-MoE, demonstrating that our Tra-MoE consistently exhibits superior performance compared to the dense baseline model, even when the latter is scaled to match Tra-MoE's parameter count.*

1. Introduction

In computer vision and natural language processing, significant progress has been made through learning models from multiple domains data. For example, many works use board data to perform self-supervised pre-training, and then adapt to diverse downstream tasks through zero-shot [1, 10, 49] or fine-tuning [13, 15, 23, 58]. Additionally, some research efforts [43, 51, 71, 72] focus on learning a generalist model capable of unifying various tasks. In contrast, in robot learning, due to data scarcity and homogeneity, some groundbreaking methods [2, 37] resort to training only using in-domain data. Recently, some works [3, 4, 52, 61, 69] have shifted to learn from action-free human video data. A particularly scalable and promising paradigm [5, 20, 59, 63, 65] involves learning a trajectory prediction model from action-free video data, and then using small-scale action-labeled demonstrations to learn trajectory-guided policies, thereby achieving greater sample efficiency and better generalization.

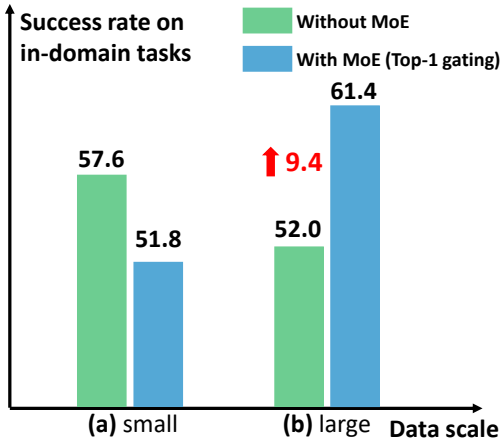

Figure 1. **(a)** Training with small-scale, in-domain data. **(b)** Joint training with in-domain data and large-scale, out-of-domain data.

Despite the potential benefits, it still remains underexplored how to effectively leverage broad out-of-domain video data for jointly training trajectory models. In particular, the impact of incorporating data encompassing diverse environments, objects, skills and embodiments has not fully investigated for in-domain tasks. For instance, ATM [63], a notable work in this field, mainly relies on in-domain data (same tasks and environments) for training both trajectory

*Corresponding author.

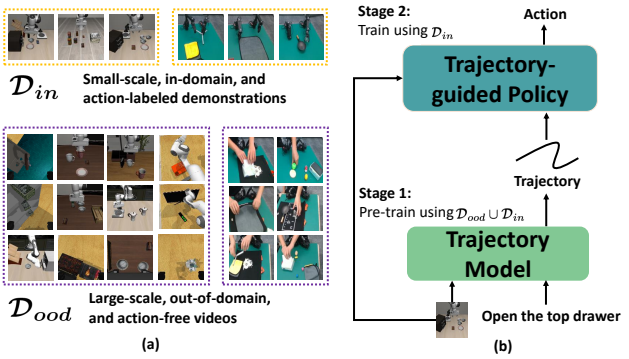

Figure 2. **Training trajectory prediction model from multiple domains**. **(a)** The visualization of Dataset $\mathcal{D}_{ood}$ and $\mathcal{D}_{in}$. The former may contain additional environments, objects, skills and embodiments. **(b)** Our pipeline: first co-training trajectory prediction model and then adapting it for downstream policy learning.

models and policies. We argue that learning from multiple domains is the primary factor to build a powerful trajectory model for policy guidance. Therefore, we propose to train trajectory model under the setup of jointly learning from large-scale out-of-domain and small-scale in-domain data, as shown in Fig. 2(a). However, directly using large-scale out-of-domain video data for hybrid pre-training involves two critical challenges: **(i)** Joint training on data from different environments, objects, skills, and embodiments will add learning difficulty and might lead to optimization conflicts. It is unclear how to effectively fuse the complementary information of multiple domains data within a single unified model. *Naively expanding the out-of-domain training data does not necessarily lead to improved performance on in-domain tasks*, as shown in Fig. 1, which results in a 5.6 (57.6 $\rightarrow$ 52.0) performance decrease on in-domain tasks. **(ii)** How to ensure both high performance and computational efficiency during trajectory model scaling up process.

To address these challenges, we design a new sparsely-gated Mixture-of-Expert (MoE) architecture [53] to scale up our trajectory model, coined as **Tra-MoE**. Specifically, we integrate several MoE blocks to replace the original transformer blocks. This design enables efficient joint training of most parameters on the broad out-of-domain data, capturing the complementary patterns for mutual cooperation and the shared knowledge for better generalization. Meanwhile, for different data and different tokens within the same data, our Tra-MoE naturally forms a specialization when activating different experts. To maintain high computational efficiency while scaling up model capacity, we implement a **top-1** gating strategy for token-choice, ensuring constant FLOPs per token. In sparse MoE, some auxiliary losses are commonly used to enhance performance, including the router z-loss [74] for improving training stability and the load-balancing loss [36] for balancing expert activations. We further conduct a comprehensive empirical study on scaling up trajectory models and find that the touter z-loss improves performance while the load-balancing loss tends to cause performance degradation. Furthermore, we observe that when expanding pre-training data from small-scale in-domain to large-scale in-domain and out-of-domain data, employing MoE (Tra-MoE) can lead to obvious performance improvement on in-domain tasks, whereas the dense counterpart (Tra-baseline) suffers from performance drop, as illustrated in Fig. 1. Notably, even when expanding the dense Tra-baseline's parameter to match that of Tra-MoE, performance improvements remain elusive. These findings collectively demonstrate the effectiveness of our sparsely-gated MoE architecture, attributing its capability of balancing parameter cooperation and specialization when expanding training with out-of-domain data.

Furthermore, given a pre-trained trajectory model, how to effectively condition the policy module using the 2D trajectories still remain an unresolved challenge. We further propose an adaptive policy conditioning technique for trajectory-guided policy. This enables better spatial alignment between 2D trajectories and images, while achieving more flexible trajectory representation and guidance for the policy. Finally, we also conduct extensive real-world experiments to further strengthen our conclusions. In summary, our main contributions can be summarized as follows:

- We propose a sparsely-gated MoE architecture to learn trajectory models, called as Tra-MoE, with large-scale, out-of-domain, and action-free video data, coupled with a comprehensive empirical study on its training techniques.
- Extensive simulation and real-world environments demonstrate that our Tra-MoE (with Top-1 gating) can more effectively benefit from large-scale out-of-domain data (such as cross-embodiments data and different physics engine simulated data) compared to the dense baseline.
- We demonstrate that Tra-MoE still significantly outperform expanded dense counterpart with equivalent parameters, highlighting the effectiveness of sparse MoE architecture in leveraging out-of-domain data through better parameter cooperation and specialization.
- We propose an adaptive policy conditioning technique for trajectory-guided policy, which ensures adaptive 2D trajectory representation and explicit spatial alignment with the observations, thereby achieving superior performance.

2. Related Work

Learning from multiple domains in Robot Learning. Scaling laws [30] have fueled substantial advancements in various fields. In robot learning, many recent studies [7, 8, 29, 51] investigate how compute, model size, and training data quantity affect the model performance on robotic manipulation tasks. Based on this, recent works have focused on collecting larger-scale real-world robotic datasets [11, 19, 32, 60, 64], and have successfully trained several robotics foundational models [6, 9, 16, 33, 38, 47, 57,

62] that demonstrate superior transfer capabilities. It is worth noting that these works mainly employ pre-training and fine-tuning paradigm rather than joint co-training paradigm, while focusing on evaluating the combination generalization and neglecting the performance impact on in-domain tasks. Additionally, these works directly combine heterogeneous robotic data with different action and observation spaces for joint training, potentially leading to sub-optimal solutions. In contrast, we leverage sparsely-gated MoE to incorporate larger-scale, out-of-domain, and action-less video data. Furthermore, our experiments demonstrate the importance of MoE for expanding out-of-domain data due to its superior capacity for parameter cooperation and specialization.

Mixture-of-Experts. The Mixture-of-Experts (MoE) framework [12] is built upon a simple yet effective concept: dividing a model into specialized components, or 'experts', each focusing on distinct tasks or data aspects. This paradigm allows for selective activation of relevant experts based on the input, thereby maintaining computational efficiency while leveraging a vast reservoir of specialized knowledge through increased model capacity. Recently, in computer vision, natural language processing, and multi-modal, there has been a surge of foundation models trained based on the MoE architecture, including Swin-Moe [24], MoE-LLaVA [39], DeepSeek-V2 [40], Mixtral-8x22B [28], and so on. MoE typically encompass dense [48] and sparse [14] variants. Dense MoE activates all experts in each iteration, while sparse MoE activates only a subset, generally resulting in lower computational overhead. However, sparse MoE can suffer from expert load imbalance and training instability issues. Techniques such as adding noise, load balancing loss [36], and router z-loss [74] are often employed to mitigate these issues and improve performance. In our work, we conduct a comprehensive empirical study of these techniques when training our sparsely-gated MoE-based trajectory model.

Representations and Techniques for Policy Conditioning. Policy conditioning equips a multi-task robot policy with the ability to modulate its behavior by incorporating task-specific guidance information. This information, which may take various forms, guides the policy's action generation, enabling a single model to effectively learn and perform diverse tasks. Classic policy conditioning representations often encompass task identifiers [50], natural language instructions [8, 22, 45, 54], goal images [44], and demonstration videos [25, 27]. These are typically integrated with visual observations within the policy through concatenation or attention mechanisms. Recent works have explored alternative policy conditioning representations to enhance generalization and sample efficiency. These include leveraging internet-trained models to obtain bounding box [55] and object mask [42, 66, 68, 73], as well as learning plan [4], affordance [3, 46], future observations [17, 18, 34, 67], 2D or 3D trajectory [5, 61, 63, 65, 70] from human video data, and also hand-drawn trajectory [21] or sketch [56]. Different from these approaches that directly utilize raw or manually crafted signals for integration with visual observations, we introduce an adaptive policy conditioning technique that explicitly maps 2D trajectories onto the visual observations in space and encodes them as learnable embeddings. These designs ensure adaptive 2D trajectory representation and explicit spatial alignment with the visual observations.

3. Methodology

In this section, we first describe our problem formulation in Sec. 3.1. Subsequently, we give an overview of our proposed framework, which is composed of a sparsely-gated MoE-based trajectory prediction model and a trajectory-guided policy model using the adaptive policy conditioning technique, detailed in Sec. 3.2. Finally, we describe our training process and loss functions in Sec. 3.3.

3.1. Problem Formulation

Following the setup of previous works [5, 63, 65], our approach encompasses a trajectory prediction model to conduct language-conditioned any-point trajectory prediction and a trajectory-guided policy model to predict executable robot actions. In particular, as depicted in Fig. 2, we consider two types of datasets for training, which are large-scale, out-of-domain, and action-free videos $\boldsymbol{\mathcal{D}_{ood}}$, as well as small-scale, in-domain, and action-labeled demonstrations $\boldsymbol{\mathcal{D}_{in}}$. For trajectory prediction model, we employ $\boldsymbol{\mathcal{D}_{ood}}$ and $\boldsymbol{\mathcal{D}_{in}}$ for joint pre-training, whereas for trajectory-guided policy model, we only use $\boldsymbol{\mathcal{D}_{in}}$ for training.

3.2. Architecture

ATM Revisited. In this paper, we use ATM [63] as the baseline for our method. The overall pipeline of ATM is illustrated on Fig. 2(b). It first learns a trajectory model from video data to perform language-conditioned trajectory prediction. The predicted trajectories is then used to learn a trajectory-guided policy using robot demonstrations. Specifically, as for trajectory model, its objective is to predict the future of any-point trajectories in a frame. Formally, given an image observation o_t at timestep t, any set of 2D query points simpled on the image frame $\mathbf{p}_t = \{p_{t,k}\}_{k=1}^{K}$, and a language instruction ℓ, the trajectory model learns a mapping $\mathbf{p}_{t:t+H} = \tau_\theta(o_t, \mathbf{p}_t, \ell)$ that predicts the coordinates of these query points in the camera frame for the next H time steps, where $\mathbf{p}_{t:t+H} \in \mathbb{R}^{H\times K\times 2}$. For the trajectory-guided policy, it leverages the current observation o_t and the predicted trajectory $\mathbf{p}_{t:t+H}$ to predict the subsequent robot actions.

Sparsely-gated MoE-based trajectory model (Tra-MoE). ATM proposes a track transformer to implement the trajectory model. In track transformer, language tokens, image tokens, and track tokens to be decoded involve global interactions within the transformer block. However, directly

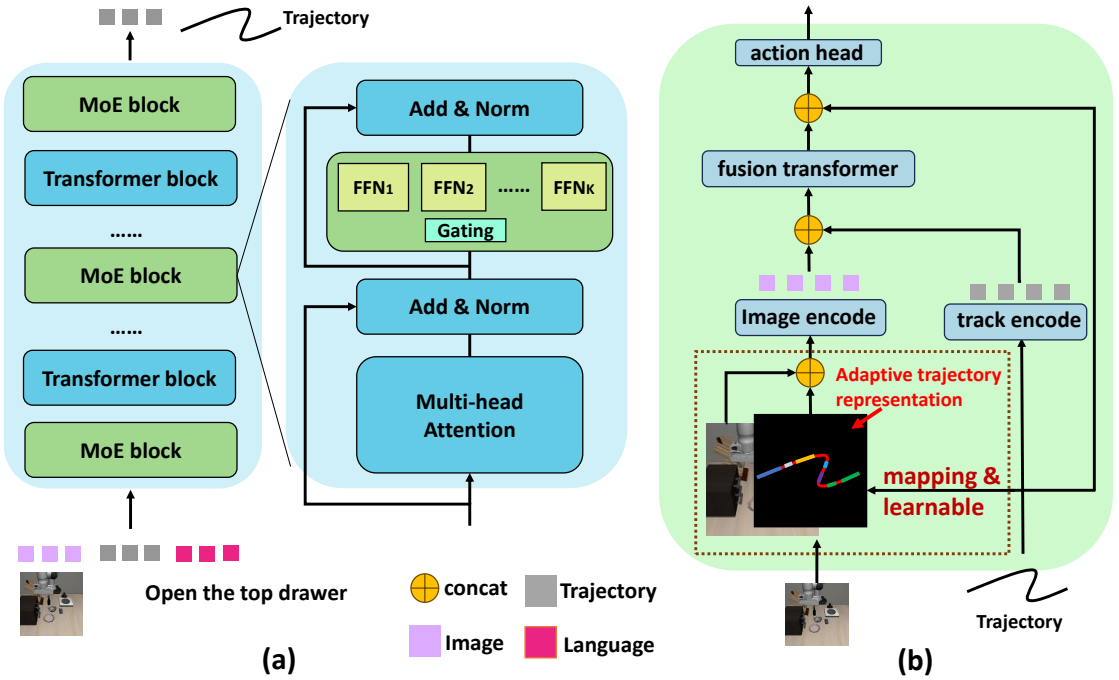

Figure 3. **(a)** The pipeline of our sparsely-gated MoE-based trajectory model (**Tra-MoE**). **(b)** The pipeline of our trajectory-guided policy using the adaptive policy conditioning technique. Mapping means concatenating the trajectory mask with image observations, while learnable refers to setting each point in the trajectory mask as a learnable embedding.

processing diverse domains and multiple modalities data within a unified transformer may yield suboptimal results. Hence, we propose our Tra-MoE framework. To develop our sparsely-gated MoE-based trajectory model, as depicted in Fig. 3(a), we replace certain layers of the transformer with MoE blocks. In each MoE block, it incorporates multiple feedforward networks (FFNs), each designated as an expert, and utilizes a gating function to activate a selected subset (Top-K) of these experts. In this paper, we set K to a fixed value of 1 to ensure that the FLOPs per token remain constant relative to the baseline. Formally, the gating network $\mathcal{G}$, parameterized by $\mathbf{\Theta}$ and typically consisting of a linear-softmax network, yields the output $\mathcal{G}(\mathbf{x_s}; \mathbf{\Theta})$, where $\mathbf{x_s}$ represents our token sequences input. We formulate the sparsely-gated token-choice mechanism as follows:

$$\mathcal{G}(\mathbf{x_s}; \mathbf{\Theta})_i = \text{softmax}(\text{Top-K}(g(\mathbf{x_s}; \mathbf{\Theta}), k))_i, \tag{1}$$

$$\text{Top-K}(g(\mathbf{x_s}; \mathbf{\Theta}), k)_i = \begin{cases} g(\mathbf{x}; \mathbf{\Theta})_i, & \text{cond}, \\ -\infty, & \text{oth}. \end{cases}, \tag{2}$$

$$\text{cond} : \text{if } g(\mathbf{x}; \mathbf{\Theta})_i \text{ is in top-}k \text{ elements of } g(\mathbf{x}; \mathbf{\Theta}). \tag{3}$$

More specifically, Top-K$(\cdot, k)$ function retains only the top-k values in a vector, setting the rest to $-\infty$. Consequently, the output of the sparsely-gated MoE layer can be formulated as:

$$\mathcal{F}^{\text{MoE}}_{\text{sparse}}(\mathbf{x_s}; \mathbf{\Theta}, \{\mathbf{W}_i\}_{i=1}^K) = \sum_{i=1}^{K} \mathcal{G}(\mathbf{x_s}; \mathbf{\Theta})_i f_i(\mathbf{x_s}; \mathbf{W}_i), \tag{4}$$

where f_i represents the i-th expert, usually a linear-GELU-linear FFN layer, is parameterized by $\mathbf{W}_i$.

Trajectory-guided policy using the adaptive policy conditioning technique. After completing the trajectory model learning, we develop our trajectory-guided policy through the adaptive policy conditioning technique based on the initial ATM, as shown in Fig. 3(b). In our policy model, independently encoded track tokens and image tokens undergo a early fusion within the fusion transformer. Subsequently, the fused features are further integrated with the raw trajectory data through a late fusion mechanism along the channel dimension, ensuring the preservation of original trajectory information for more accurate action prediction.

Fundamentally, the policy model bridges the gap between the predicted 2D any-point trajectories and the executable 3D robot actions, functioning as an inverse dynamic model and eliminating the necessity of other task specification. However, how to effectively condition the policy using the predicted 2D trajectories remains an unresolved challenge. Based on our analysis, we present two key insights: **(i)** Explicit spatial alignment and fusion of the 2D trajectories with image observations turns out to be highly beneficial.

It simplifies the mapping of 2D trajectories to image space and highlights motion-relevant regions, enabling the policy model to focus on important areas and reducing the learning difficulty. **(ii)** Trajectory positions exhibit different characteristics: starting points typically emphasize local motion, while endpoints focus more on global trend. Therefore, an adaptive trajectory representation and conditioning technique is expected to flexibly guide the policy by effectively capturing these variations.

Based on the aforementioned analysis, we propose an adaptive policy conditioning technique to incorporate trajectory information into 2D images. Specifically, we construct an additional mask modality, and populate it with 2D trajectories based on their specific spatial locations, while setting the value of each point in the trajectory mask as a learnable embedding. Before encoding the image, we concatenate this mask modality with the image observations on the channel dimension, forming a tensor of dimensions $H \times W \times 4$, as shown in Fig. 3(b). To sum up, our adaptive policy conditioning technique yield an adaptive 2D trajectory representation, which could be explicitly aligned with the image observations to guide the policy prediction more flexibly.

3.3. Training

We argue that trajectory models could be be shared by multiple domains data. Therefore, we first pre-train a generalizable trajectory model on $\boldsymbol{\mathcal{D}_{ood} \cup \mathcal{D}_{in}}$. The $\boldsymbol{\mathcal{D}_{ood}}$ typically contains additional environments, objects, skills, and embodiments. Then we train the trajectory-guided policy model on $\boldsymbol{\mathcal{D}_{in}}$ based on the pre-trained trajectory model. In trajectory-guided policy training phase, we freeze the trajectory model.

Pre-training sparsely-gated MoE-based trajectory model. Following ATM, we train the trajectory model using the ground truth tracks generated by CoTracker [31]. We use MSE to optimize the trajectory prediction loss $\mathcal{L}_{\text{tra}}$. Additionally, we also consider adding an auxiliary load-balancing loss [36] to balance the activation frequency among experts. Given a batch of queries $\mathcal{B}$, it contains B samples and each sample contains S tokens. We define $\mathcal{Q}_i$ as the proportion of tokens distributed to expert i, and $\mathcal{P}_i$ as the proportion of the gating probability assigned to expert i. The $\mathcal{L}_{\text{lo-ba}}$ can be formulated as follows:

$$\mathcal{Q}_i = \frac{1}{S \cdot B} \sum_{x_s^m \in \mathcal{B}} \mathbb{1}\{\operatorname{argmax} \mathcal{G}(\mathbf{x_s^m}; \mathbf{\Theta}) = i\}, \quad (5)$$

$$\mathcal{P}_i = \frac{1}{S \cdot B} \sum_{x_s^m \in \mathcal{B}} \mathcal{G}(\mathbf{x_s^m}; \mathbf{\Theta})_i, \quad (6)$$

$$\mathcal{L}_{\text{lo-ba}} = N \sum_{i=1}^{N} \mathcal{Q}_i \mathcal{P}_i. \quad (7)$$

Additionally, ST-MoE [74] proposed using the router z-loss to penalize large logits entering the gating network, thereby improving training stability. We similarly applies it to the training of our trajectory model, as shown below:

$$\mathcal{L}_z(x) = \frac{1}{S} \sum_{k=1}^{S} \left(\log \sum_{i=1}^{N} e^{\mathbf{g}_i^{(k)}} \right)^2, \quad (8)$$

where S is the total number of tokens, N is the total number of experts, and $\mathrm{g} \in \mathcal{R}^{S \times N}$ are the logits going into the router. In summary, the trajectory model's training loss is as follow:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{tra}} \cdot \mathcal{L}_{\text{tra}} + \lambda_{\text{lo-ba}} \cdot \mathcal{L}_{\text{lo-ba}} + \lambda_z \cdot \mathcal{L}_z. \quad (9)$$

Training trajectory-guided policy through adaptive policy conditioning technique. For trajectory-guided policy training, we employ the paradigm of behavior cloning. Specifically, we use the MSE between predicted actions and ground truths as the loss function.

4. Experiments

We perform evaluation experiments using the LIBERO benchmark [41] and a real low-cost dual-arm robot [35]. In experiments, we quantitatively find that the success rate of downstream manipulation tasks is generally positively correlated with the performance of the trajectory model. Detailed quantitative experiment results can be found in the appendix. Meanwhile, CoTracker [31] can provide relatively accurate trajectory annotations; therefore, we report the average success rate of downstream manipulation tasks as our metric. Our experiments aim to study the following questions:

Q1: Does the techniques of router z-loss, load-balancing loss, and adding noise to gating logits enhance the training of sparsely-gated Tra-MoE?

Q2: Does sparsely-gated Tra-MoE, compared to the dense Tra-baseline, benefit more from large-scale out-of-domain data (such as cross-embodiments and different physics engine data) and demonstrate superior scaling up capabilities?

Q3: When we scale up the dense Tra-baseline to match the parameter count of Tra-MoE, does the sparsely-gated Tra-MoE still maintain its advantage?

Q4: Is our proposed adaptive policy conditioning technique for trajectory-guided policy more effective than ATM?

4.1. Simulation Experiments

Experiment setup. The LIBERO [41] is divided into five categories: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, LIBERO-Long, and LIBERO-90. LIBERO-90 includes 90 tasks, while each of the other categories contains 10 tasks. Compared to other categories, LIBERO-90 presents a highly varied selection of object categories, layouts, environments, and skill goals. It is typically used for *pre-training* rather than *joint co-training*. Therefore, our main ablation experiments select 90 tasks from LIBERO-90 to construct

	Spatial	Goal	Object	Long	Avg.
0	62.0	73.0	71.0	21.5	56.9
$5{\cdot}10^{-4}$	69.5	73.5	68.0	22.5	58.4
$1{\cdot}10^{-4}$	62.5	81.0	73.5	28.5	**61.4**
$1{\cdot}10^{-5}$	60.0	65.5	68.5	27.0	55.3
$1{\cdot}10^{-6}$	60.0	79.0	53.0	31.5	55.9

(a) The value of λ_z.

	Spatial	Goal	Object	Long	Avg.
0	62.0	73.0	71.0	22.5	56.9
$1{\cdot}10^{-3}$	58.5	74.0	60.0	18.0	52.6
$1{\cdot}10^{-5}$	61.0	69.5	56.0	26.0	53.1
$1{\cdot}10^{-7}$	66.5	67.0	73.5	22.0	**57.3**

(b) The value of λ_{lo-ba}.

	Spatial	Goal	Object	Long	Avg.
w/o. noise	62.0	73.0	71.0	22.5	**56.9**
w. noise	71.5	67.0	57.5	27.0	55.8

(c) Adding a noise term to the gating logits.

	Spatial	Goal	Object	Long	Avg.
MoE	62.5	81.0	73.5	28.5	**61.4**
Width	66.5	50.5	61.5	15.0	48.4
Depth	61.5	75.5	46.5	26.5	52.5

(d) Scaling Tra-baseline from width and depth.

Num.	Spatial	Goal	Object	Long	Avg.
1	49.5	67.0	56.5	35.0	52.0
2	59.0	72.5	65.5	27.0	56.0
3	64.0	68.0	71.0	17.5	55.1
4	62.0	73.0	71.0	21.5	**56.9**

(e) The number of experts.

MoE	OOD	Spatial	Goal	Object	Long	Avg.
		67.5	68.5	68.0	26.5	57.6
✓		63.0	64.0	60.5	19.5	51.8
	✓	49.5	67.0	56.5	35.0	52.0
✓	✓	62.5	81.0	73.5	28.5	**61.4**

(f) Scaling with MoE and OOD data.

Table 1. The ablation experiments of the sparsely-gated Tra-MoE. All baselines are the track transformer of ATM.

dataset $\mathcal{D}_{ood}$, utilizing 20 action-free videos per task; and choose 40 tasks from LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long to build dataset $\mathcal{D}_{in}$, using 10 action-labeled demonstrations per task. In summary, for training Tra-MoE, our main ablation experiments use a total of $2,200$ action-free videos, with an out-of-domain to in-domain data ratio of $9:2$. Additionally, on this basis, we also further select 92 tasks from the RLbench [26] (using CoppeliaSim rather than MuJoCo as physics engine) simulation environment and generate 5 videos for each task, totaling $2,660$ videos in conjunction with the LIBERO data, to train Tra-MoE. The different physics engines and dynamics properties further increase the data distribution discrepancy in $\mathcal{D}_{ood}$. Finally, as for the trajectory-guided policy, we train a separate multi-task policy for each category. Each policy undergoes behavioral cloning training with 100 action-labeled demonstrations from 10 tasks.

Implement details. Building on ATM [63], we develop our sparsely-gated Tra-MoE and adaptive policy conditioning technique. Specifically, when using only LIBERO data, we replace the 1th, 5th, and 8th layers of the track transformer with MoE blocks. When further incorporating RLbench data, we additionally replace the 2th and 7th layers. Unless otherwise specified, we set the number of experts to 4 by default and use a **top-1** gating strategy for token choice to ensure the FLOPs per token remain constant.

Experiment results and analysis. We present empirical simulation results to address the aforementioned research questions and make the following findings:

Result 1: In Tab. 1a, we examine the effect of the router z-loss weight λ_z. The results indicate that a higher λ_z can effectively enhance performance, with an improvement of 4.5 (61.4 vs. 56.9) at $\lambda_z = 1 \cdot 10^{-4}$. In Tab. 1b, we study the impact of the load-balancing loss weight λ_{lo-ba}. The results show a significant performance decline with larger values of λ_{lo-ba}, with only a slight improvement (56.7 $\rightarrow$ 57.3) at $\lambda_{lo-ba} = 1 \cdot 10^{-7}$. In Tab. 1c, we verify the effectiveness of adding noise to the gating logits, which results in a performance decrease of by 1.1 (56.9 $\rightarrow$ 55.8).

	Spatial	Goal	Object	Long	Avg.
ATM	51.0	42.0	55.5	19.0	41.9
Tra-MoE	54.0	62.0	61.0	41.0	**54.5**

Table 2. The comparison results when training trajectory model using RLbench video data.

Finding 1: Based on the above results, we find that using router z-loss to penalize large logits entering the gating network can effectively improve training stability and lead to performance gains. Meanwhile, the load-balancing loss used to balance the activation frequency among experts and the addition of noise to enhance expert allocation exploration cannot bring effective performance improvements. An intuitive explanation is that when the data distribution is uneven, the load-balancing loss might force experts to set shared parameters on data with a large domain gap, which could lead to mutual counteraction of learning gradients. This undermines the advantage of parameter specialization in sparsely-gated MoE. Similarly, adding noise is also detrimental to learning parameter specialization among experts.

Result 2: We comprehensively investigate the impact of in-

	Spatial	Goal	Object	Long	Avg.
ATM	62.5	81.0	73.5	28.5	61.4
+ DM	69.0	58.0	85.5	33.5	61.5
+ AM	69.5	77.0	88.0	30.5	**66.3**

Table 3. The ablation experiments of the adaptive policy conditioning technique. DM (RT-Trajectory style) and AM respectively represent hand-drawn mask and adaptive mask.

troducing sparsely-gated MoE and large-scale out-of-domain data on scaling up trajectory models in Tab. 1f. Firstly, when training only with in-domain data, Tra-baseline achieves an average success rate of 57.6, while Tra-MoE only achieves an average success rate of 51.8. Secondly, when adding out-of-domain data to jointly train Tra-baseline, the average success rate drops to 52.0 (57.6 $\rightarrow$ 52.0), with particularly significant performance declines in LIBERO-Spatial and LIBERO-Object. However, when adding out-of-domain data to jointly train Tra-MoE, the average success rate increases to 61.4 (51.8 $\rightarrow$ 61.4). Additionally, in Tab. 1e, we also explore the impact of the number of experts in sparsely-gated MoE. The results show that using just two experts can lead to an increase of 4.0 (52.0 vs. 56.0) in the average success rate. With additional experts, performance generally trends upward, albeit with some fluctuations. Finally, we report the comparison results of training trajectory model by further using RLbench data in Tab. 2. The results indicate that although the introduction of RLbench data led to a performance drop in LIBERO, our Tra-MoE still outperform the baseline by 12.6 (54.5 vs. 41.9).

Finding 2: Based on the above results, we have the following findings and analysis: **(i)** When the data scale is small, naively expanding the model capacity through sparsely-gated MoE does not lead to performance improvements. A explanation is that the larger model may lead to a overfitting problem when training on small-scale data. Moreover, small-scale data often leads to some experts not being sufficiently trained, thereby affecting performance. **(ii)** Naively introducing out-of-domain data for joint training often leads to performance degradation in the target domain. However, simultaneously incorporating out-of-domain data and expanding models with sparsely-gated MoE may improve performance or alleviate performance degradation. This phenomenon suggests that although out-of-domain data generally enhances model generalization, it could hinder task-specific performance in the target domain due to insufficient in-domain data learning. In contrast, expanding the model with out-of-domain data using sparsely-gated MoE achieves a more optimal balance between parameter cooperation and specialization across diverse data and even different tokens within the same data input. This design allows these large-scale data to jointly train most of parameters, ensuring the capture of general patterns for mutual cooperation. **(iii)** Even with a few experts, Tra-MoE exhibits significant scaling up capabilities. Additionally, further increasing the number of experts generally leads to performance improvement. However, this improvement is not guaranteed due to factors like data-expert complexity matching and training stability.

Result 3: In Tab. 1d, we also investigate the performance comparison between the dense Tra-baseline and the sparse Tra-MoE with the same number of parameters. Specifically, we expand Tra-baseline to match the model capacity of Tra-MoE by increasing its model width or depth separately. These expanded versions achieve average success rates of 48.4 and 52.5 respectively, which are significantly lower than Tra-MoE's 61.4.

Finding 3: Based on these results, we are surprised to find that simply expanding the dense Tra-baseline in depth and width can not effectively improve performance. This highlights the importance of the sparsely-gated MoE architecture when training with large-scale out-of-domain data, as it ensures dynamic activation of different experts based on different data input and different tokens within the same data input, significantly reducing optimization conflict problems.

Result 4: In Tab. 3, we investigate the effectiveness of the adaptive policy conditioning. Specifically, building upon ATM, we first introduce an additional hand-drawn mask modality before image encoding. In this mask, the first half of the trajectory is set to 128, while the second half is set to 255. We call it 'ATM + hand-drawn mask' and this is similar to RT-Trajectory [21]. Next, we replace the hand-drawn mask with an adaptive mask, where each point on the trajectory is instantiated as a learnable embedding. We call it 'ATM + adaptive mask'. The results show that 'ATM + hand-drawn mask' achieves only a 0.1 improvement compared to the baseline. However, except for a significant performance decrease in LIBERO-Goal, it achieves performance improvements of 6.5, 12.0, and 5.0 in LIBERO-Spatial, LIBERO-Object, and LIBERO-Long respectively. In addition, the 'ATM + adaptive mask' achieves a performance improvement of 4.9 (66.3 vs. 61.4) compared to the baseline. It further improves performance across all suites compared to 'ATM + hand-drawn mask', especially achieving a 19.0 performance gains in LIBERO-Goal (77.0 vs. 58.0).

Finding 4: Based on these results, we find that explicitly mapping 2D trajectories to visual observations significantly improves the performance of LIBERO-Spatial, LIBERO-Object, and LIBERO-Long, but causes a notable decrease in LIBERO-Goal. By instantiating the points in the 2D trajectory as learnable embeddings, we not only further enhances the performance of LIBERO-Spatial, LIBERO-Object, and LIBERO-Long, but also significantly improves LIBERO-Goal. We analyze that the decline in LIBERO-Goal is mainly due to overlapping front and back parts of some trajectories in the 2D images, leading to mapping errors. However, our adaptive trajectory representation and conditioning technique effectively mitigates this issue. In summary, these

Human Data	MoE	Adaptive Trajectory Mask	Pour	Push	Pick and Pass	Tissue	Fold	Average
			40.0	45.0	50.0	30.0	25.0	38.0
✓			45.0	35.0	50.0	25.0	35.0	38.0
✓	✓		40.0	50.0	60.0	35.0	45.0	46.0
✓	✓	✓	60.0	70.0	65.0	35.0	50.0	**56.0**

Table 4. The ablation of ood human data, MoE architecture, and adaptive policy conditioning technique on our real-world tasks.

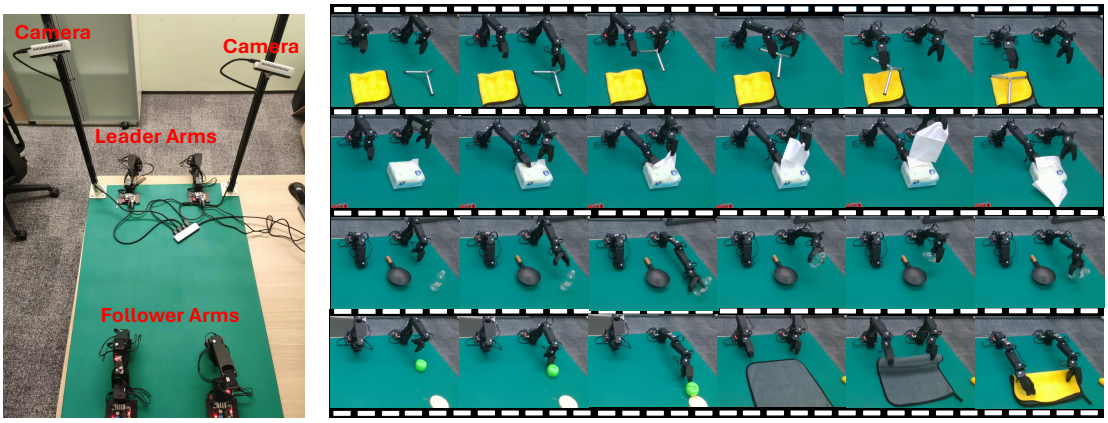

Figure 4. **Left:** The real-world experiments hardware platform setup. **Right:** The real-world tasks evaluation demonstrations.

experiments collectively demonstrate the effectiveness of explicitly aligning 2D trajectories with images in space as well as applying the adaptive conditioning technique.

4.2. Real-World Experiments

Experiment setup. As shown in Fig 4, we utilize a dual-arm robot [35] to conduct real-world experiments. We evaluate 5 tasks, which include two single-arm tasks: pouring water and pushing a vegetable to the side of the cutting board, as well as three dual-arm tasks: pulling out tissues, folding towels, and picking up and passing a holder. Specifically, for each task, we collect 50 action-labeled robot demonstrations. Concurrently, we also collect 50 videos of human (cross-embodiments) performing for each task. Compared to the robot data, the human video encompass more environment variations and distractor objects. Please see the appendix for more detailed evaluation details and real-world setup.

Implement details. Following our simulation experiments, we utilize two camera views and uniformly resize their resolution to 128×128 for trajectory model and policy training. We train our real-world Tra-MoE on borh human video and robot video data, with a separate policy trained for each task. Therefore, our real-world Tra-MoE integrates data across different embodiments (human, single-arm robot, and dual-arm robot). They have significantly different visual appearances and dynamics properties. Finally, we also present some real-world task evaluation demonstrations in Fig. 4.

Experiment results and analysis. To further strengthen our conclusions that sparsely-gated MoE can better learn trajectory model from out-of-domain data and our adaptive policy condition technique could lead to more flexible guidance for policy learning, we conduct real-world ablation experiments, as shown in Tab. 4. The results indicate that simply introducing human video data cannot lead to performance improvement. However, when expanding the model by MoE architecture, the performance increases from 38% to 46%. We speculate this may be due to the significant differences in appearance and dynamics between human and robot, with the former typically moving much faster and more flexibly. Finally, when further incorporating an adaptive trajectory mask for policy conditioning, the performance is further increased from 46% to 56%.

5. Conclusion

In this work, we have introduced **Tra-MoE**, a novel framework that utilizes a sparsely-gated MoE architecture with **Top-1** gating strategy to learn trajectory model from multiple domains. Tra-MoE achieves superior parameter cooperation and specialization, enabling more effective utilization of large-scale out-of-domain data while maintaining constant FLOPs per token. We train our Tra-MoE and conduct a comprehensive empirical study, demonstrating that Tra-MoE consistently exhibits superior performance compared to the dense baseline, even when the latter is scaled to match Tra-MoE's parameter count. Additionally, we propose an adaptive policy conditioning technique for trajectory-guided policy, ensuring adaptive 2D trajectory representation and explicit spatial alignment with observations, thereby achieving superior performance. Extensive experiments in both simulated and real-world environments validate our findings and demonstrate the effectiveness of our proposed methods.

Acknowledgements

This work is supported by the National Key R&D Program of China (No. 2022ZD0160900), Jiangsu Frontier Technology Research and Development Program (No. BF2024076), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] 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. *IJRR*, 39(1): 3–20, 2020. 1

[3] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *CVPR*, pages 13778–13790, 2023. 1, 3

[4] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *ICRA*, pages 6904–6911. IEEE, 2024. 1, 3

[5] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *arXiv preprint arXiv:2405.01527*, 2024. 1, 3

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2

[7] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Manon Devin, Alex X Lee, Maria Bauza Villalonga, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving generalist agent for robotic manipulation. *TMLR*, 2023. 2

[8] 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

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 2

[10] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[11] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 2

[12] Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*, 2024. 3

[13] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 1

[14] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *CVPR*, pages 11828–11837, 2023. 3

[15] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[16] Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *CoRL*. 2

[17] Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, Leslie Pack Kaelbling, et al. Video language planning. In *The Twelfth International Conference on Learning Representations*. 3

[18] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *NeurIPS*, 36, 2024. 3

[19] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023. 2

[20] Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Cai Zhehao, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. In *ICLR*. 1

[21] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *ICLR*. 3, 7

[22] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In *CORL*, pages 175–187. PMLR, 2023. 3

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1

[24] Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, et al. Tutel: Adaptive mixture-of-experts at scale. *Proceedings of Machine Learning and Systems*, 5:269–287, 2023. 3

[25] Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, et al. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. *arXiv preprint arXiv:2403.12943*, 2024. 3

[26] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *RAL*, 5(2):3019–3026, 2020. 6

[27] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *CoRL*, pages 991–1002. PMLR, 2022. 3

[28] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. 3

[29] 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. 2

[30] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2

[31] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 5

[32] 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. 2

[33] 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. 2

[34] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *ICLR*. 3

[35] Alexander Koch. Low-cost robot arm. `https://github.com/AlexanderKoch-Koch/low_cost_robot`, 2024. GitHub repository. 5, 8

[36] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 2, 3, 5

[37] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *JMLR*, 17(39):1–40, 2016. 1

[38] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *ICLR*. 2

[39] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 3

[40] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024. 3

[41] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *NeurIPS*, 36, 2024. 5

[42] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*. 3

[43] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2022. 1

[44] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *CoRL*, pages 1113–1132. PMLR, 2020. 3

[45] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *RAL*, 2023. 3

[46] Soroush Nasiriany, Sean Kirmani, Tianli Ding, Laura Smith, Yuke Zhu, Danny Driess, Dorsa Sadigh, and Ted Xiao. Rt-affordance: Affordances are versatile intermediate representations for robot manipulation. *arXiv preprint arXiv:2411.02704*, 2024. 3

[47] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 2

[48] Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*, 2023. 3

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICLR*, pages 8748–8763. PMLR, 2021. 1

[50] Rouhollah Rahmatizadeh, Pooya Abolghasemi, Ladislau Bölöni, and Sergey Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *ICRA*, pages 3758–3765. IEEE, 2018. 3

[51] 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. 1, 2

[52] Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *ICLR*, pages 19561–19579. PMLR, 2022. 1

[53] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 2

[54] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *CoRL*, pages 894–906. PMLR, 2022. 3

[55] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Brianna Zitkovich, Fei Xia, Chelsea Finn, et al. Open-world object manipulation using pre-trained vision-language models. In *CoRL*, 2023. 3

[56] Priya Sundaresan, Quan Vuong, Jiayuan Gu, Peng Xu, Ted Xiao, Sean Kirmani, Tianhe Yu, Michael Stark, Ajinkya Jain, Karol Hausman, et al. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches. 2024. 3

[57] 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. 2

[58] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 1

[59] Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. In *ICRA*, pages 5397–5403. IEEE, 2024. 1

[60] 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 *CoRL*, pages 1723–1736. PMLR, 2023. 2

[61] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *CoRL*. 1, 3

[62] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *CoRR*, abs/2409.20537, 2024. 3

[63] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 1, 3, 6

[64] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024. 2

[65] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024. 1, 3

[66] Jiange Yang, Wenhui Tan, Chuhao Jin, Bei Liu, Jianlong Fu, Ruihua Song, and Limin Wang. Pave the way to grasp anything: Transferring foundation models for universal pick-place robots. *arXiv preprint arXiv:2306.05716*, 1(2), 2023. 3

[67] Jiange Yang, Bei Liu, Jianlong Fu, Bocheng Pan, Gangshan Wu, and Limin Wang. Spatiotemporal predictive pre-training for robotic motor control. *arXiv preprint arXiv:2403.05304*, 2024. 3

[68] Jiange Yang, Wenhui Tan, Chuhao Jin, Keling Yao, Bei Liu, Jianlong Fu, Ruihua Song, Gangshan Wu, and Limin Wang. Transferring foundation models for generalizable robotic manipulation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 1999–2010, 2025. 3

[69] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024. 1

[70] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024. 3

[71] Jinguo Zhu, Xizhou Zhu, Wenhai Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, and Jifeng Dai. Uni-perceiver-moe: Learning sparse generalist models with conditional moes. *NeurIPS*, 35:2664–2678, 2022. 1

[72] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *CVPR*, pages 16804–16815, 2022. 1

[73] Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. In *CoRL*. 3

[74] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022. 2, 3, 5