PREDICTIVE INVERSE DYNAMICS MODELS ARE SCALABLE LEARNERS FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

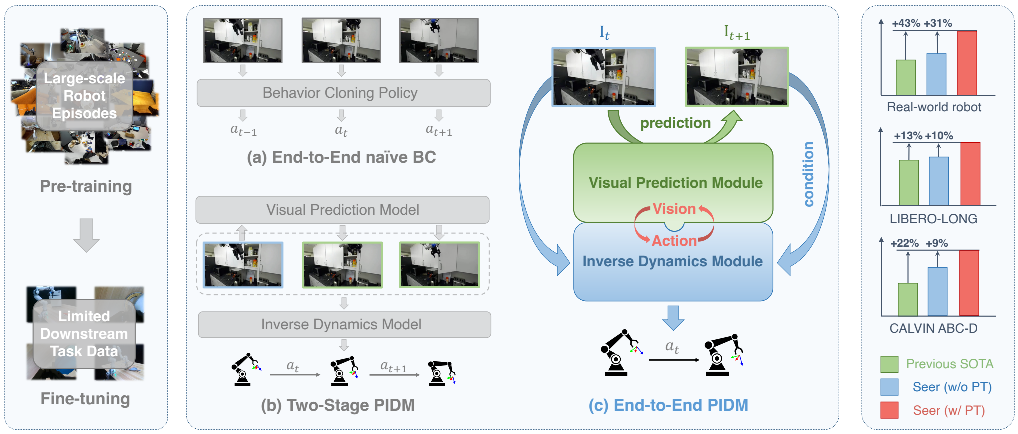


Figure 1: In contrast to previous methods that (a) conduct end-to-end naive behavior cloning from large-scale robotic data or (b) use decoupled visual prediction and inverse dynamics models to set goals and guide actions, we present end-to-end Predictive Inverse Dynamics Models (PIDM) that closes the loop between vision and action. Seer, the model we built, surpasses previous states of the art and demonstrates consistent improvements over the ablated version without pre-training.

ABSTRACT

Current efforts to learn scalable policies in robotic manipulation primarily fall into two categories: one focuses on “action,” which involves behavior cloning from extensive collections of robotic data, while the other emphasizes “vision,” enhancing model generalization by pre-training representations or generative models, also referred to as world models, using large-scale visual datasets. This paper presents an end-to-end paradigm that predicts actions using inverse dynamics models conditioned on the robot’s forecasted visual states, named Predictive Inverse Dynamics Models (PIDM). By closing the loop between vision and action, the end-to-end PIDM can be a better scalable action learner. In practice, we use Transformers to process both visual states and actions, naming the model Seer. It is initially pre-trained on large-scale robotic datasets, such as DROID, and can be adapted to real-world scenarios with a little fine-tuning data. Thanks to large-scale, end-to-end training and the continuous synergy between vision and action at each execution step, Seer significantly outperforms state-of-the-art methods across both simulation and real-world experiments. It achieves improvements of 13% on the LIBERO-LONG benchmark, 22% on CALVIN ABC-D, and 43% in real-world tasks. Notably, it demonstrates superior generalization for novel objects, lighting conditions, and environments under high-intensity disturbances. Code and models will be publicly available.

1 INTRODUCTION

Learning scalable and generalizable policies has become a main focus in robotic manipulation. Recent efforts primarily fall into two categories: one focuses on “action,” like RT-1 (Brohan et al., 2022) and OpenVLA (Kim et al., 2024), which perform naive behavior cloning from large-scale robotic data such as Open X-Embodiment and DROID (Padalkar et al., 2024; Khazatsky et al., 2024). The other emphasizes "vision" and may learn representations through discriminative or generative ways and integrate with the control policy in a two-stage manner. For example, R3M (Nair et al., 2022) and MVP (Xiao et al., 2022) learn discriminative representations from large-scale video datasets such as Ego4D (Grauman et al., 2022), while UniPI (Du et al., 2024) and Susie (Black et al., 2023) develop generative models as "world models" to facilitate manipulation policies. Apparently, the scaling laws in robot learning are still evolving, with researchers exploring strategies through diverse data and methods.

We revisit these approaches and propose that a scalable manipulation policy should integrate vision and action in a closed loop. This integration is natural and necessary, as humans typically coordinate their hands and eyes to manipulate objects. Therefore, closing the loop during training and inference are both necessary for a better scalable action learner.

This paper achieves this by introducing a simple yet effective end-to-end Predictive Inverse Dynamics Models (PIDM) that can unify the advantages of previous methods. As shown in Figure 1, it predicts actions using Inverse Dynamics Models (IDM) conditioned on the "Predictive" visual states of the robot. During training, both the visual prediction module and the inverse dynamics module are optimized synergistically in an end-to-end manner. During inference, PIDM ensures continuous synergy between vision and action at each execution step. In contrast to previous methods that use IDM, our approach is the first to optimize vision and action in an end-to-end manner. Throughout this paper, unless otherwise specified, PIDM is assumed to be end-to-end.

In practice, we use Transformers to process both visual states and actions and name the model Seer. Seer benefits from PIDM by simultaneously leveraging visual, temporal, and action information from large-scale datasets, and can be more scalable due to the Transformer architecture. We introduce a foresight token to predict future RGB images and an action token to estimate intermediate actions between current and predicted future observations. Both tokens are fused with input RGB images, robot state, and language tokens through a multi-modal encoder. Importantly, we design a unidirectional attention mask that allows the action token to deeply integrate past and future predictive information, facilitating end-to-end training.

We conduct extensive experiments on both simulation and real-world benchmarks. On two widely adopted simulation benchmarks, LIBERO-LONG (Liu et al., 2024) (10 tasks) and CALVIN ABC-D (Mees et al., 2022) (34 tasks), our method demonstrates a 10.4% improvement in success rate and a 0.71 increase in average task completion length compared to state-of-the-art baselines. Our results further indicate superiority in long-horizon task completion, unseen scene generalization, and data efficiency. Additionally, We evaluate our method on four challenging real-world tasks with over 900 trials. Leveraging the public large robot dataset DROID (Khazatsky et al., 2024), our method consistently shows robustness, even under disturbances and with limited fine-tuning data.

2 Related Work

Action-Centric Pre-training for Manipulation. Recent advancements in action-centric pre-training have significantly enhanced manipulation policies. Approaches like SMART (Sun et al., 2023) and DualMind (Wei et al., 2023) emphasize understanding the dynamics within environments. Some studies (Agrawal et al., 2016; Brandfonbrener et al., 2024) integrate current and goal information to extract effective features or serve as an auxiliary objective. Subsequently, a standard behavior cloning approach is applied during downstream task implementations. Additionally, RT-X (Padalkar et al., 2024) and Octo (Ghosh et al., 2024) focus on pre-training robot policies using diverse datasets to facilitate extensive generalization capabilities. Recently, vision-language models (VLMs) have demonstrated considerable common-sense knowledge about the world and strong capabilities in understanding both language and images. OpenVLA (Kim et al., 2024) further pre-trains VLMs using robotic data, leveraging their prior knowledge to achieve robust performance on downstream language-conditioned manipulation tasks. While these methods primarily supervise actions, they do not fully exploit the rich visual and temporal information inherent in robot demonstrations. In contrast, we pre-train policies by integrating conditional visual foresight and inverse dynamics prediction, allowing for comprehensive utilization of robotic data.

Vision-Centric Pre-training for Manipulation. Extensive research has focused on visual pre-training for visuomotor control (Karamcheti et al., 2023; Zeng et al., 2024). One major direction

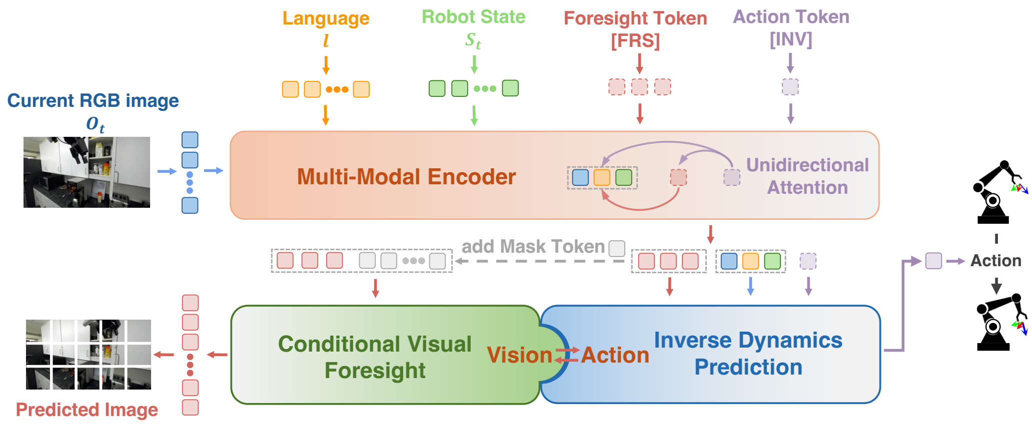


Figure 2: **Pipeline of Seer.** Seer consists of three parts: Multi-Modal Encoder, Conditional Visual Foresight and Inverse Dynamics Prediction. In Multi-Modal Encoder, Seer incorporates the foresight token [FRS] and the action token [INV]. Both tokens attend to the RGB images, language tokens, and robot state tokens, with [INV] also attending to [FRS]. In Conditional Visual Foresight, the encoded [FRS], along with new mask tokens, aims to reconstruct future RGB images. In Inverse Dynamics Prediction, the encoded [INV] and other tokens speculate intermediary actions.

involves representation learning using techniques such as masked image modeling (Xiao et al., 2022; Radosavovic et al., 2023; Seo et al., 2023), contrastive learning (Nair et al., 2022; Ma et al., 2022; 2023), and generative video pre-training (Wu et al., 2024). Another line of work focuses on visual expectations guiding actions, termed the Predictive Inverse Dynamics Model (PIDM) (Bharadhwaj et al., 2024; Wang et al., 2024; Soni et al., 2024; Chen et al., 2024). Firstly, a video generation model predicts future visual sub-goals and is pre-trained on general visual datasets. Then, an inverse dynamics (goal-conditioned) low-level policy is trained on downstream robot data to predict the intermediary actions. Compared to these two-stage PIDM, we propose an end-to-end PIDM paradigm that leverages large-scale robot data for pre-training, showing better performance.

Pre-training Datasets for Manipulation. High-quality, large-scale, and diverse pre-training data is crucial for acquiring manipulation skills. Image datasets (Deng et al., 2009), video datasets (Damen et al., 2018; Goyal et al., 2017); (Grauman et al., 2022), and robot datasets are commonly utilized for this purpose. Image datasets provide rich semantic information, while video datasets contain temporal information. Both enhance visual representations for manipulation, however, their lack of action labels and robot states limits their utility in decision-making. Some studies focus on collecting robot behavior data (Mandlekar et al., 2019; Walke et al., 2023; Dasari et al., 2019; Bahl et al., 2023; Jang et al., 2022), but the data diversity remains relatively constrained. Recent efforts aim to further scale and diversify robot datasets. For instance, the Open X-Embodiment dataset (Padalkar et al., 2024) aggregates data from 22 different robots across 21 institutions, covering 527 skills and 160,266 tasks. DROID (Khazatsky et al., 2024) includes 76,000 trajectories collected across 564 scenes and 86 tasks. In this work, we leverage DROID to pre-train policies for real-world validation, demonstrating that rich behavioral data significantly enhances success rates in downstream tasks.

3 Method

In this section, we describe Seer in detail. We begin with a brief problem formulation (Section 3.1). Next, we discuss keys in our end-to-end PIDM—conditional visual foresight and inverse dynamics prediction (Section 3.2), enabling Seer to forecast the future and adjust actions accordingly. We then elaborate on the carefully designed model architecture (Section 3.3), through which we formulate Seer in an end-to-end manner. Finally, we provide implementation details (Section 3.4).

3.1 Problem Formulation

Given a large-scale dataset of diverse manipulation demonstrations $D_1 = \{(l, o_t, s_t, a_t)_{t=0}^{T_i}\}_{i=0}^{N_1}$, and a smaller downstream dataset $D_2 = \{(l, o_t, s_t, a_t)_{t=0}^{T_j}\}_{j=0}^{N_2}$ (where $N_1 >> N_2$), our goal is to enhance downstream task performance through effective pre-training on D_1, followed by fine-tuning on D_2. Each trajectory $\{(l, o_t, s_t, a_t)_{t=0}^{T}\}$ provides the time step t, language instruction l, RGB images o_t from the eye-on-hand and eye-on-base views, robot states s_t and robot actions a_t, which include arm actions a_{arm} (6D pose) and gripper actions a_{arm} (open or close). It is important to note that current large pre-training robot data may contain incomplete language annotations l and task-agnostic actions a_t, such as random exploration in the environment (Mees et al., 2022). However, Seer could handle this scenario effectively due to the following specific design choices.

3.2 End-To-End PIDM

Vision: Conditional Visual Foresight. A key insight is that informative future states guide actions. Therefore, we propose conditional visual foresight f_{fore} to effectively anticipate future visual representations. Seer takes as input a goal g in the form of language instructions or robot states, along with historical observations h_t, and predicts the RGB images at the time step $t+n$, denoted by $\hat{o}_{t+n}$

$$\hat{o}_{t+n} = f_{\text{fore}}(g, h_t). \tag{1}$$

The historical observations h_t consist of the RGB images $o_{t-m+1:t}$ and robot states $s_{t-m+1:t}$ over the last m time steps. Due to the rich information contained in RGB images, their abundance, and easy accessibility, we select them as future representatives. Following (He et al., 2022), the loss function $\mathcal{L}_{\text{fore}}$ computes the mean squared error (MSE) at the pixel level

$$\mathcal{L}_{\text{fore}} = \| f_{\text{fore}}(g, h_t) - o_{t+n} \|_2^2. \tag{2}$$

Action: Inverse Dynamics Prediction. Given two temporally ordered observations o_t and o_{t+1}, inverse dynamics prediction estimates the intermediate action $\hat{a}_t$. Here, we extend inverse dynamics f_{inv} to predict the action sequence $\hat{a}_{t:t+n-1}$ given goal g, historical observations h_t and o_{t+n}. Specifically, we replace ground truth o_{t+n} with the predicted representation $\hat{o}^l_{t+n}$ in the latent space

$$\hat{a}_{t:t+n-1} = f_{\text{inv}}(g, h_t, \hat{o}^l_{t+n}). \tag{3}$$

The loss function $\mathcal{L}_{\text{inv}}$ comprises the arm action loss $\mathcal{L}_{\text{arm}}$ and the gripper action loss $\mathcal{L}_{\text{gripper}}$

$$\mathcal{L}_{\text{inv}} = \mathcal{L}_{\text{arm}} + \lambda \mathcal{L}_{\text{gripper}}, \tag{4}$$

where $\mathcal{L}_{\text{arm}}$ is a Smooth-L1 loss, $\mathcal{L}_{\text{gripper}}$ is a Binary Cross Entropy (BCE) loss and λ is set to 0.01.

Close the Loop between Vision and Action. Seer integrates conditional visual foresight with inverse dynamics prediction effectively through training, enabling full utilization of both vision and action information in robot data. In detail, f_{fore} incorporates a clear goal g and historical observations h_t to predict future RGB images $\hat{o}_{t+n}$. A latent representation $\hat{o}^l_{t+n}$ (leading to $\hat{o}_{t+n}$) and h_t facilitate action prediction via f_{inv}. Due to the model design of Seer, all these processes are executed in an end-to-end manner. The overall training loss $\mathcal{L}$ comprises $\mathcal{L}_{\text{fore}}$ and $\mathcal{L}_{\text{inv}}$

$$\mathcal{L} = \alpha \mathcal{L}_{\text{fore}} + \mathcal{L}_{\text{inv}}, \tag{5}$$

where α is a hyperparameter set to 0.5. Compared to single-step action prediction, predicting multiple steps provides temporal action consistency and robustness to idle actions (Chi et al., 2023). During inference, we can either discard actions beyond the first step or apply temporal ensemble techniques to compute a weighted average of the multi-step actions.

3.3 Model architecture

Input Tokenizers. As illustrated in Figure 2, the model processes three types of inputs: language, images, and robot states. We use different encoders to tokenize each modality accordingly. For language inputs, we first tokenize the text and then use a CLIP text encoder (Radford et al., 2021) to obtain text embeddings, which are subsequently projected into a latent space using a linear layer. For image inputs, the images are first patchified and passed through a pre-trained Vision Transformer (ViT) (He et al., 2022) to generate visual embeddings. Since the ViT produces hundreds of embeddings per image, imposing a significant computational burden on the transformer backbone, and much of the visual information is irrelevant to the manipulation task, we employ a perceiver resampler (Alayrac et al., 2022) to extract task-relevant visual features and reduce the number of image tokens. For the robot state, we encode it into state tokens using a multi-layer perceptron (MLP).

Multi-Modal Encoder. The multi-modal encoder in our model is based on a GPT-2 style transformer architecture. Before feeding the sequential language-image-state pairs into the transformer, we append readout tokens [INV] and [FRS] to each timestep. These readout tokens attend to embeddings from different modalities, serving as image and action latents used for conditional visual foresight and inverse dynamics prediction. To incorporate temporal information, we also add a learnable position embedding to the tokens for each timestep.

The [FRS] token aims to facilitate conditional visual foresight, corresponding to aforementioned $\hat{o}^l_{t+n}$. It attends to language, historical image and state tokens. Conversely, the [INV] token performs inverse dynamics prediction conditioned on the predicted visual foresight, attending to the input tokens and, crucially, the foresight token [FRS]. This special unidirectional attention mask in a transformer encoder, highlighted in Figure 2, brings two benefits. First, this will help the [INV] token deeply integrate both past and future predictive information within a multi-layer network. Second, this enables an end-to-end training paradigm through the fusion in the latent space.

Readout Decoders. Encoded by the multi-modal encoder, the action and image latents generated by the [INV] and [FRS] readout tokens are fed into the readout decoders to predict images and actions. The action decoder utilizes an MLP to transform the action latent into the action vector a_t. For image decoding, we employ a vision transformer (ViT) as the image decoder, following (He et al., 2022). The image decoder takes the image latent along with masked tokens as input. Processed by ViT, the output corresponding to each masked token represents a specific patch of the image.

3.4 Implementation Details

Training. During training, the pre-trained visual and text encoders remain frozen. The training objectives remain consistent–conditional visual foresight and inverse dynamics prediction, enabling a smooth transition from pre-training to fine-tuning. Notably, two key differences in model configurations exist between these phases. First, missing language instructions are common in robotic pre-training datasets. In such cases, during pre-training, the robot state token at the future time step $t + n + 1$ acts as a goal. The [FRS] would attend to it instead of the language token, ensuring [FRS] to acquire unambiguous information. Second, pre-training data may include random or meaningless behaviors, such as environmental exploration. Consequently the [INV] and [FRS] tokens do not attend to previous image and robot state tokens to prevent overfitting to any specific behaviors.

Inference. During inference, the complete language instruction l, robot states s, and image observations o are provided as inputs. The [FRS] token attends to the historical image, state, and language instruction tokens to perform conditional visual foresight, predicting the future images. In turn, the [INV] token attends to the input tokens and one more foresight [FRS] token to perform inverse dynamics prediction, outputting the action. Further details are available in the Appendix.

4 Simulation Experiments

We conduct experiments on two simulation benchmarks LIBERO-LONG (Liu et al., 2024), CALVIN ABC-D (Mees et al., 2022). Our aim is to answer: 1) How does our method perform on challenging simulation benchmarks? 2) Does our pipeline maintain consistent effectiveness as the amount of downstream fine-tuning data varies? 3) Are the training objectives in Seer effective?

4.1 Benchmarks, Baselines and Metrics

Benchmarks. LIBERO-LONG (Liu et al., 2024) encompasses diverse object interactions and versatile motor skills. We pre-train our model on the LIBERO-90 dataset, which includes demonstrations for 90 short-horizon tasks with **full annotations**, and then fine-tune and evaluate it on LIBERO-LONG, which features long-horizon tasks. CALVIN ABC-D (Mees et al., 2022) is a benchmark focusing on language-conditioned visual robot manipulation. It contains 34 tasks across four distinct environments (Env A, B, C, and D), each varying in object and scene visual appearance. For pre-training, we utilize the official robot play data with **no language instructions**, while the remaining data with full annotations is used for fine-tuning.

Baselines. For LIBERO-LONG, we implement a vanilla multi-task policy MTACT without pre-training, general image-based pre-trained policy MVP (Xiao et al., 2022), video-based pre-trained

Table 1: **LIBERO-LONG results.** For each task, we present the average performance of top-3 checkpoints averaged over 20 rollouts. Avg. Success is the average success rate across ten tasks. We demonstrate superior performance over baselines, achieving higher Avg. Success and better results on a large majority of tasks. The best results are **bolded**.

Method	Avg. Success ↑	Put soup and box in basket	Put box and butter in basket	Turn on stove and put pot	Put bowl in drawer and close it
MTACT	41.0	30.0	50.0	75.0	85.0
MVP	68.2	83.3	90.0	80.0	88.3
MPI	77.3	66.6	86.6	96.6	95.0
OpenVLA	54.0	35.0	**95.0**	65.0	45.0
Ours (w/o pretrain)	78.7	80.0	90.0	91.7	81.7
Ours	**87.7**	**91.7**	90.0	**98.3**	**100**
Put mugs on left and right plates	**Pick book and place it in back**	**Put mug on plate and put pudding to right**	**Put soup and sauce in basket**	**Put both pots on stove**	**Put mug in microwave and close it**
20.0	75.0	0.00	0.00	10.0	65.0
46.7	63.3	45.0	78.3	60.0	46.7
83.3	83.3	56.6	86.6	40.0	**78.3**
40.0	80.0	60.0	45.0	20.0	55.0
85.0	65.0	**86.7**	**88.3**	51.7	66.7
91.7	**93.3**	85.0	**88.3**	**61.7**	71.7

policy MPI (Zeng et al., 2024) and robot-data-based pre-trained policy OpenVLA (Kim et al., 2024). For CALVIN ABC-D, we select baselines that have demonstrated top competitive performance in prior reports. Roboflamingo (Li et al., 2023) is a method stepped from a vision-language model (Alayrac et al., 2022). Susie (Black et al., 2023) is a classical two-stage PIDM. GR-1 (Wu et al., 2024) relies on generative video pre-training, while 3D Diffusor Actor captures 3D representations to enhance manipulation.

Metrics. In LIBERO-LONG, each method is evaluated across 20 rollouts with varying initial states for each task. We report both per-task and average success rates. In CALVIN ABC-D, the robot executes 1,000 task sequences, with each sequence comprising five consecutive tasks; a new task is initiated only after the preceding task has been successfully completed. We report the average success rates and the average length of completed sequences.

4.2 Simulation Main Results

Table 2: **CALVIN ABC-D results.** We present the average success rates of top-3 checkpoints computed over 1000 rollouts for each task and the average number of completed tasks to solve 5 instructions consecutively (Avg. Len.). Ours shows consistent and significant superiority over baselines. The best results are **bolded**.

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len. ↑
Roboflamingo	82.4	61.9	46.6	33.1	23.5	2.47
Susie	87.0	69.0	49.0	38.0	26.0	2.69
GR-1	85.4	71.2	59.6	49.7	40.1	3.06
3D Diffusor Actor	92.2	78.7	63.9	51.2	41.2	3.27
Ours (w/o pretrain)	93.0	82.4	72.3	62.6	53.3	3.64
Ours	**94.4**	**87.2**	**79.9**	**72.2**	**64.3**	**3.98**

We conduct experiments on the LIBERO-LONG benchmark. The results, presented in Table 1, indicate that our policy achieved an average success rate of 78.7% without pre-training. After pre-training, the success rate increases by an additional 9%, significantly outperforming the baselines. Compared to MTACT, our policy is more effective and benefits further from pre-training. The visual pre-training methods MVP and MPI achieve performance levels only comparable to our policy without pre-training, suggesting that visual pre-training alone is insufficient for manipulation tasks. Pre-training the entire policy using robotic data is necessary to enhance both perception and decision-making capabilities. In comparison to OpenVLA, our model (316M) uses only 4% of its parameters (7B) yet achieves a 62% relative improvement in performance. We hypothesize that the moderate model size reduces the risk of overfitting. Overall, these results underscore the advantages Seer and demonstrate the effectiveness of our pre-training objectives.

We evaluate various methods on CALVIN ABC-D. Notably, the pre-training data lacks language annotations and includes meaningless actions and random explorations within these environments. Evaluation is conducted in Environment D, which differs visually from Environments A, B, and C where the data was collected. As shown in Table 2, our method significantly outperforms the baselines. Our method surpasses the two-stage PIDM Susie (Wu et al., 2024) by a large margin, probably due to our delicate model design and end-to-end training paradigm. It also outperforms the video generative pre-training method GR-1 (Wu et al., 2024), demonstrating the advantage of pre-training the entire policy. Furthermore, our results indicate that our method remains applicable even in the absence of language annotations and with noisy pre-training datasets. It can also effectively handle a certain degree of visual appearance variation.

4.3 Data Efficiency

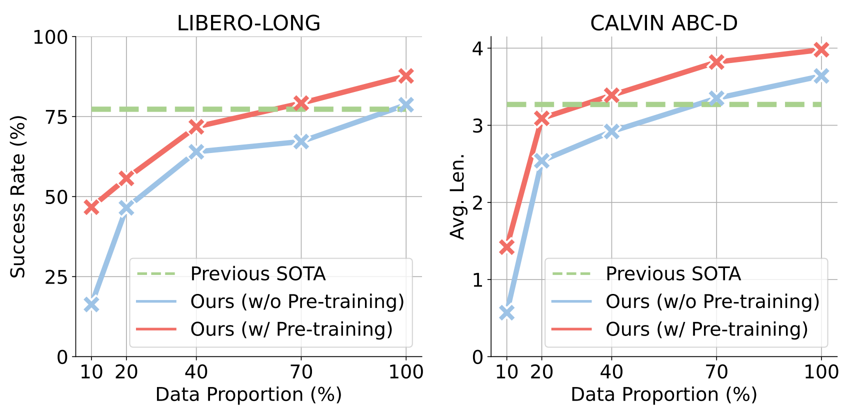


Figure 3: **Data efficiency.** We fine-tune Seer using different propotions of the downstream data. Our method consistently benefits and surpasses previous SOTA baselines with less data.

Collecting robot data is both time-consuming and labor-intensive, making data efficiency crucial for robot learning. We evaluate our method on two benchmarks: LIBERO-LONG and CALVIN ABC-D, using 10%, 20%, 40%, 70%, and 100% of the available data to fine-tune pre-trained policies or to train policies from scratch. The results, shown in Figure 3, demonstrate that our method consistently enhances policy performance across varying data sizes. Notably, under data-scarce conditions with only 10% of the training data, the pre-trained policy achieves a 187% relative improvement in success rate on LIBERO-LONG and a 150% relative improvement in average task length on CALVIN ABC-D compared to training from scratch. Additionally, our method only requires 70% data on LIBERO-LONG and 40% data on CALVIN ABC-D respectively to surpass state-of-the-art baselines. These results highlight the potential of Seer in scenarios with limited fine-tuning data.

4.4 Ablation Studies

We investigate the contributions of conditional visual foresight objective $\mathcal{L}_{\text{fore}}$ and inverse dynamics prediction objective $\mathcal{L}_{\text{inv}}$ during pre-training and fine-tuning on CALVIN ABC-D. The objectives during the fine-tuning phase are most closely related to performance in downstream tasks. Thus, we prioritize ablating the fine-tuning objectives before examining the pre-training objectives.

Fine-tuning objectives. We study the importance of $\mathcal{L}_{\text{fore}}$ and $\mathcal{L}_{\text{inv}}$ during fine-tuning. As shown in Table 3a, compared to the vanilla baseline, which directly behavior clones (w/o $\mathcal{L}_{\text{fore}}$, w/o $\mathcal{L}_{\text{inv}}$), separately predicting additional future images (w/ $\mathcal{L}_{\text{fore}}$, w/o $\mathcal{L}_{\text{inv}}$) yields improvements. This indicates the benefits of involving future image predictions (Wu et al., 2024). More importantly, integrating

Table 3: **Ablation studies on fine-tuning and pre-training objectives.** Integrating the conditional visual foresight objective $\mathcal{L}_{\text{fore}}$ and inverse dynamics prediction objective $\mathcal{L}_{\text{inv}}$ yields the best performance among pre-training and fine-tuning.

(a) **Fine-tuning objectives.**

Fine-tuning $\mathcal{L}_{\text{fore}}$	Fine-tuning $\mathcal{L}_{\text{inv}}$	1	2	3	4	5	Avg. Len.
×	×	89.9	77.6	64.6	54.4	44.8	3.31
✓	×	91.2	78.6	67.1	56.6	47.8	3.41
✓	✓	93.0	82.4	72.3	62.6	53.3	3.64

(b) **Pre-training objectives.**

Pre-train $\mathcal{L}_{\text{fore}}$	Pre-train $\mathcal{L}_{\text{inv}}$	1	2	3	4	5	Avg. Len.
×	×	93.0	82.4	72.3	62.6	53.3	3.64
✓	×	92.3	83.0	74.2	65.9	57.5	3.73
✓	✓	94.4	87.2	79.9	72.2	64.3	3.98

$\mathcal{L}_{\text{fore}}$ and $\mathcal{L}_{\text{inv}}$ results in an even greater boost in performance. This demonstrates that utilizing visual expectations to guide action predictions is a more effective strategy for leveraging the rich visual and temporal information inherent in robot data than the ablated version (w/ $\mathcal{L}_{\text{fore}}$, w/o $\mathcal{L}_{\text{inv}}$).

Pre-training objectives. Once the fine-tuning objectives ($\mathcal{L}_{\text{fore}} + \mathcal{L}_{\text{inv}}$) are established, we start to ablate pre-training objectives. The results in Table 3b indicate that pre-training only the vision prediction module (w/ $\mathcal{L}_{\text{fore}}$, w/o $\mathcal{L}_{\text{inv}}$) yields certain benefits, likely due to the vision priors learned from the extensive data. Moreover, pre-training the whole policy (w/ $\mathcal{L}_{\text{fore}}$, w/ $\mathcal{L}_{\text{inv}}$) through the integration of visual foresight and inverse dynamics results in even greater improvements. This underscores the importance of the synergy between action and vision priors distilled from large robot datasets in enhancing performance on downstream tasks.

5 Real-World Experiments

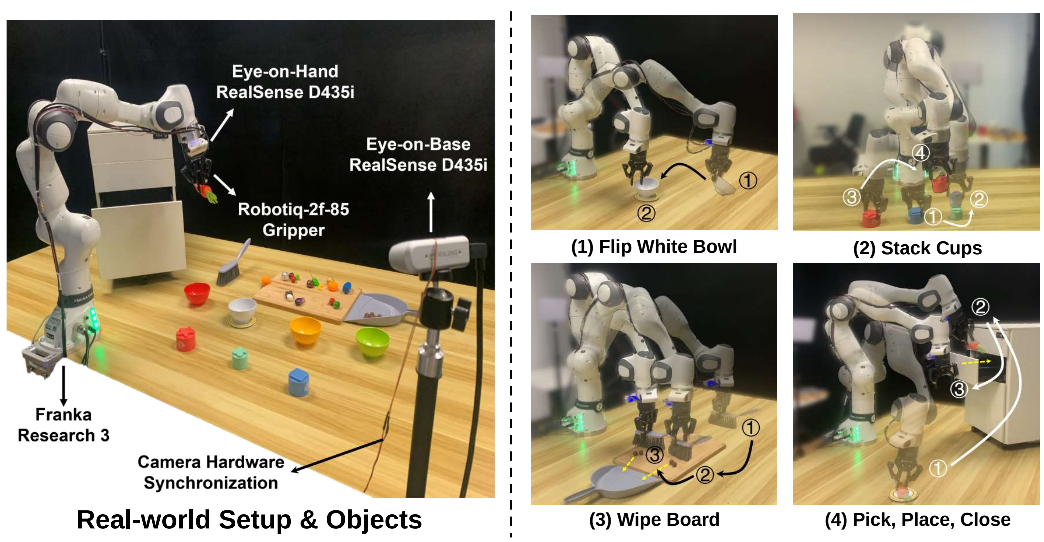


Figure 4: **Real-world Benchmark. Left:** We use a Franka Research 3 robot with a Robotiq-2f-85 gripper and two RealSense D435i cameras. We include different everyday objects on manipulation tasks. **Right:** We design four real-world tasks. (1) Flip White Bowl: The robot needs to ①pick an overturned bowl and ②place it on the coaster. (2) Stack Cups: The robot needs to ①pick the middle cup, ②cover the small one, ③pick the big one, and ④cover the middle one. (3) Wipe Board: The robot needs to ①grasp the brush, and ②③sweep all the chocolate balls into the dustpan. (4) Pick, Place, Close: The robot needs to ①pick the carrot, ②put it in the drawer, and ③close the drawer.

We evaluate Seer on four challenging real-world tasks, leveraging a large-scale robot dataset, e.g., DROID (Khazatsky et al., 2024) for pre-training. We target answering: 1) Could Seer still work in real-world tasks? 2) How will Seer perform with minimal fine-tuning data (20 demos per task)? 3) Whether pre-training will bring consistent benefits under different intensive disturbances?

5.1 REAL-WORLD BENCHMARK

Real-world Setup. We evaluate on a Franka Research 3 robot equipped with a Robotiq-2f-85 gripper across four tasks, as illustrated in Figure 4. We utilize two RealSense D435i cameras, configured as Eye-on-Hand and Eye-on-Base. Additionally, we ensure hardware synchronization.

Datasets. For pre-training, we select dataset DROID, which contains demonstrations of Franka robots executing various tasks in diverse scenes. During the fine-tuning phase, we set the control frequency to 15 Hz, record RGB images, robot states, actions, and save 100 demonstrations per task.

Baselines and Metrics. We benchmark against MVP (image-based pre-trained), MPI (video-based pre-trained), and OpenVLA (robot-data-based pre-trained). Each method is evaluated over 15 trials per task, with variations in the initial states of the objects. Each method is allowed three executions per trial, with the mean performance reported. Given the long-horizon and challenging nature of the tasks, we define two metrics: **Success Rate (SR)** and **Score** (as referenced in (Kim et al., 2024)). The **Score** accumulates during the completion of specific intermediary stages, while **SR** is recorded as 100% only upon successful completion of the entire task. Details are available in the Appendix.

Tasks. In **Flip White Bowl**, the robot picks up a randomly overturned bowl and places it on a coaster, testing basic 6DoF pick & place capability. In **Stack Cups**, the robot stacks three randomly placed cups of different colors and sizes. It firstly covers the small cup with the middle one, and then covers the middle cup with the big one. Since the cup surface is smooth and covering requires a close fit, this task challenges fine-grained action predictions. In **Wipe Board**, the robot collects 3 to 7 chocolate balls arranged in 1 to 3 clusters. It uses a brush to sweep the balls into a dustpan, testing its ability to handle multi-modal settings and perform repetitive motions. In **Pick, Place, Close**, the robot picks a randomly set carrot, transports it into an opened drawer and closes the drawer. This evaluates skills of executing consecutive actions in a large space with articulated objects.

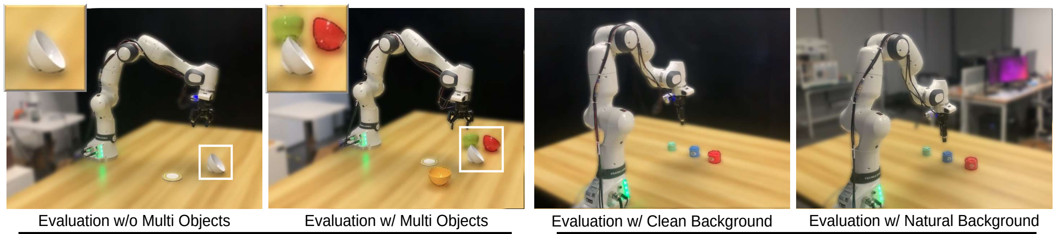
5.2 REAL-WORLD MAIN RESULTS

Table 4: **Real-world main results.** We evaluate all the methods with 15 (cases) $\times$ 3 (repeated trials) rollouts per task. Our method achieves better performances among all tasks than baselines.

Method	Demos per Task	Flip White Bowl	Stack Cups	Wipe Board	Pick, Place, Close	Avg.
		SR (%) / Score	SR (%) / Score	SR (%) / Score	SR (%) / Score	SR (%) $\uparrow$ / Score $\uparrow$
MVP	100	80.0 / 24.0	26.7 / 26.0	53.3 / 38.0	60.0 / 31.0	55.0 / 29.8
MPI	100	66.7 / 21.0	26.7 / 29.0	33.3 / 35.0	66.7 / 32.0	48.4 / 29.3
OpenVLA	100	53.3 / 19.0	0.00 / 8.00	0.00 / 4.00	13.3 / 13.0	16.7 / 11.0
Ours (w/o pre-train)	20	26.7 / 10.0	13.3 / 11.0	26.7 / 16.0	33.3 / 28.0	25.0 / 16.3
Ours	20	46.7 / 17.0	0.00 / 7.00	53.3 / 29.0	33.3 / 34.0	33.3 / 21.8
Ours (w/o pre-train)	100	60.0 / 19.0	46.7 / 35.0	60.0 / 37.0	73.3 / 40.0	60.0 / 32.8
Ours	100	**86.7 / 26.0**	**60.0 / 42.0**	**73.3 / 41.0**	**86.7 / 42.0**	**78.4 / 39.5**

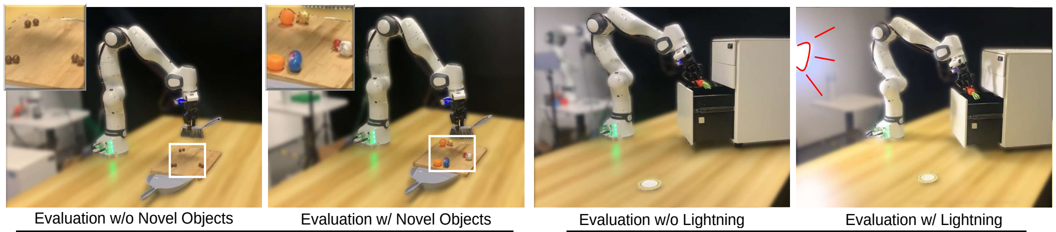
Effectiveness. As can be seen in Table 4, our pre-trained policy could outperform all the baselines over all tasks. Specifically, our method improves the average success rate and the accumulated score from 60.0% to 78.4% and from 32.8 to 39.5 compared to the version trained from scratch. In comparison with MVP and MPI, which only pre-train vision encoders, our results reinforce the importance of pre-training the entire policy, aligning with findings from simulation experiments. Regarding the performance of OpenVLA in the real world, it has a significantly larger tunable model size (3B here) during full fine-tuning and relies solely on an eye-on-base camera. This could lead to severe overfitting and coarse action predictions, particularly in tasks where objects are small (as in Stack Cups) or located far from the camera (as in Wipe Board). In contrast, our method demonstrates better handling of these tasks due to its moderate model size and comprehensive data utilization.

Efficiency. We randomly select a subset with 20 demos for fine-tuning in each task. As shown in Table 4, when the fine-tuning dataset is reduced from 100 to 20 demos per task, our pre-trained policy consistently shows improvements across most tasks. The exception is in the Stack Cups task, where we suspect that both the limited fine-tuning data and the inherent difficulty of the task contribute to low success rates across all methods, resulting in slight distinctions among them.



Type	Method	SR (%) / Score
Multi Objects	Ours (w/o pre-train)	33.3 / 11.0
	Ours	**60.0 / 18.0**

Type	Method	SR (%) / Score
Background	Ours (w/o pre-train)	6.67 / 13.0
	Ours	**33.3 / 29.0**



Type	Method	SR (%) / Score
Novel Objects	Ours (w/o pre-train)	46.7 / 37.0
	Ours	**60.0 / 39.0**

Type	Method	SR (%) / Score
Lightning	Ours (w/o pre-train)	46.7 / 30.0
	Ours	**66.7 / 37.0**

Figure 5: **Generalization evaluation.** We design a generalization test per task with different disturbances. **Top Left:** In Flip Bowl, we put several bowls with the same shape, size, material and different colors around the original white one. **Top Right:** In Stack Cups, we remove the original black backdrop and keep the natural background. **Bottom Left:** In Wipe Board, we replace the chocolate balls with diverse novel small objects. **Bottom Right:** In Pick, Place, Close, we introduce an additional light source. Among all tests, our pre-trained method brings consistent benefits.

5.3 Robustness

We propose several generalization types to assess the effectiveness of our pre-trained policy across multiple settings. As shown in Figure 5, in **Flip White Bowl**, we introduce bowls of different colors alongside the original white bowl. These bowls share identical shape, size, and material, which could potentially mislead the algorithm. In **Wipe Board**, we replace the original chocolate balls with novel objects that vary in mass, shape, and coefficients of friction, thereby increasing the task's difficulty. In both scenarios, our pre-trained policy demonstrates significant improvements in success rate (SR) and Score. We attribute these enhancements to the extensive variety of interactable and distractible objects present in the pre-training dataset, which strengthens the model's semantic understanding. Additionally, in **Pick, Place, Close**, we incorporate a strong light source that alters the visual appearance of objects in RGB images. In **Stack Cups**, we remove the clean black backdrop and replace it with a natural background, introducing complex disturbances such as variable lighting, unseen distractions, and effects on camera exposure. Even under these challenging conditions, our pre-trained policy continues to deliver satisfactory results. We believe that the extensive pre-training on large-scale robot datasets with diverse scenes contributes to this robustness.

6 Conclusion

In this work, we introduce Seer, an end-to-end predictive inverse dynamics model that synergizes conditional visual foresight with inverse dynamics prediction for manipulation. Seer shows the state-of-the-art performance on two simulation benchmarks, and demonstrates significant improvements and strong robustness in real-world experiments after being pre-trained on the large robot dataset DROID. The limitations mainly lie in two aspects. Firstly, we only evaluate 6 downstream tasks. A broader spectrum of high-precision and contact-rich tasks remain to be explored. Secondly, evaluating across different robots is also necessary to test Seer's cross-embodiments capability.

REFERENCES

Pulkit Agrawal, Ashvin Nair, P. Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *NeurIPS*, 2016.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *CVPR*, 2023.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. In *Arxiv*, 2024.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. In *NeurIPS*, 2023.

David Brandfonbrener, Ofir Nachum, and Joan Bruna. Inverse dynamics pretraining learns good representations for multitask imitation. In *NeurIPS*, 2024.

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. In *RSS*, 2022.

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. In *ArXiv*, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

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. In *Arxiv*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

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. In *NeurIPS*, 2024.

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. In *Arxiv*, 2024.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

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*, 2022.

Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *RSS*, 2023.

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. In *RSS*, 2024.

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. In *CoRL*, 2024.

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*, 2023.

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

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *ICLR*, 2022.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *ICML*, 2023.

Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *IROS*, 2019.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. In *RA-L*, 2022.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *CoRL*, 2022.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anika Singh, Anthony Brohan, Antonin Raffin, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Brian Ichter, Cewu Lu, Charles Xu, Chelsea Finn, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Chuer Pan, Chuyuan Fu, Coline Devin, Danny Driess, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Federico Ceola, Fei Xia, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Giulio Schiavi, Hao Su, Haoshu Fang, Haochen Shi, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jaehyung Kim, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeff Bingham, Jiajun Wu, Jialin Wu, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jitendra Malik, Jonathan Tompson, Jonathan Yang, Joseph J. Lim, João Silvério, Junhyek Han, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Zhang, Keyvan Majd, Krishan Rana, Krishna Parasuram Srinivasan, Lawrence Yunliang Chen, Lerrel Pinto, Liam Tan, Lionel Ott, Lisa Lee, Masayoshi Tomizuka, Maximilian Du, Michael Ahn, Mingtong Zhang, Mingyu Ding, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Manfred Otto Heess, Nikhil J. Joshi, Niko Suenderhauf, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Pannag R. Sanketi, Paul Wohlhart, Peng Xu, Pierre Sermanet, Priya Sundaresan, Quan Ho Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan C. Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Sherry Moore, Shikhar Bahl, Shivin Dass, Shuran Song, Sichun Xu, Siddhant Haldar, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Sudeep Dasari, Suneel Belkhale, Takayuki Osa, Tatsuya Harada, Tatsuya Matsushima, Ted

Xiao, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Zhao, Travis Armstrong, Trevor Darrell, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xuanlin Li, Yao Lu, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yueh hua Wu, Yujin Tang, Yuke Zhu, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zhuo Xu, and Zichen Jeff Cui. Open x-embodiment: Robotic learning datasets and rt-x models. In *ICRA*, 2024.

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 *ICML*, 2021.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *CoRL*, 2023.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *CoRL*, 2023.

Achint Soni, Sreyas Venkataraman, Abhranil Chandra, Sebastian Fischmeister, Percy Liang, Bo Dai, and Sherry Yang. Videoagent: Self-improving video generation. In *ArXiv*, 2024.

Yanchao Sun, Shuang Ma, Ratnesh Madaan, Rogerio Bonatti, Furong Huang, and Ashish Kapoor. Smart: Self-supervised multi-task pretraining with control transformers. In *ICLR*, 2023.

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*, 2023.

Boyang Wang, Nikhil Sridhar, Chao Feng, Mark Van der Merwe, Adam Fishman, Nima Fazeli, and Jeong Joon Park. This&that: Language-gesture controlled video generation for robot planning. In *ArXiv*, 2024.

Yao Wei, Yanchao Sun, Ruijie Zheng, Sai Vemprala, Rogerio Bonatti, Shuhang Chen, Ratnesh Madaan, Zhongjie Ba, Ashish Kapoor, and Shuang Ma. Is imitation all you need? generalized decision-making with dual-phase training. In *CVPR*, 2023.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *ICLR*, 2024.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. In *Arxiv*, 2022.

Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. In *RSS*, 2024.