Moto: Latent Motion Token as the Bridging Language for Robot Manipulation

Yi Chen[1,2], **Yuying Ge**[2†], **Yizhuo Li**[1,2], **Yixiao Ge**[2], **Mingyu Ding**[3], **Ying Shan**[2], **Xihui Liu**[1†]
[1]The University of Hong Kong, [2]ARC Lab, Tencent PCG,
[3]University of California, Berkeley
https://chenyi99.github.io/moto/

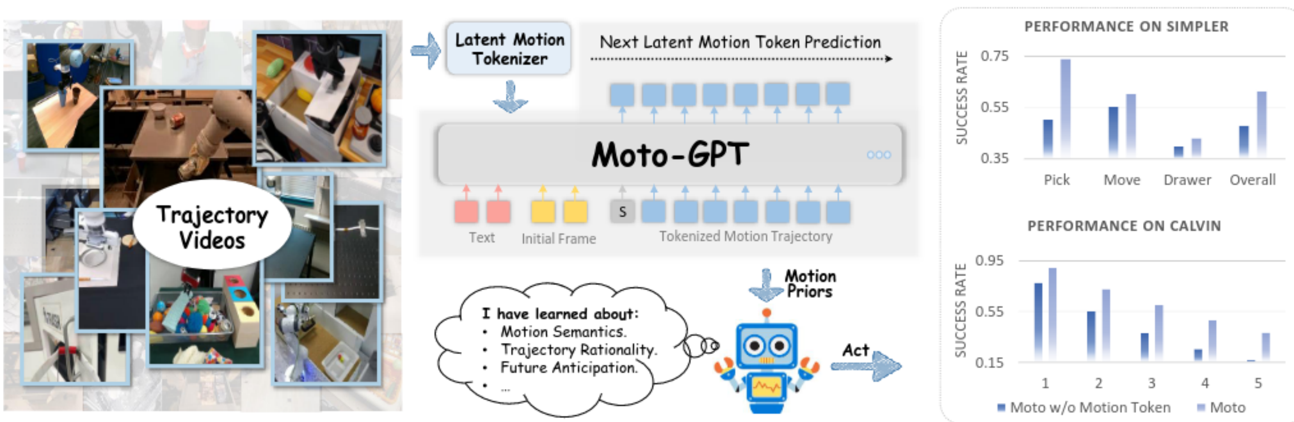

Figure 1. The overview of **Moto**, which utilizes Latent **Mo**tion **To**kens as a bridging "language" for autoregressive pretraining on video data. The Moto-GPT pre-trained through next motion token prediction learns a wealth of motion-related prior knowledge from videos, which can be seamlessly transferred to enhance downstream robot manipulation tasks with significant performance gains.

Abstract

Recent developments in Large Language Models (LLMs) pre-trained on extensive corpora have shown significant success in various natural language processing (NLP) tasks with minimal fine-tuning. This success offers new promise for robotics, which has long been constrained by the high cost of action-labeled data. We ask: given the abundant video data containing interaction-related knowledge available as a rich "corpus", ***can a similar generative pre-training approach be effectively applied to enhance robot learning?*** *The key challenge is to identify an effective representation for autoregressive pre-training that benefits robot manipulation tasks. Inspired by the way humans learn new skills through observing dynamic environments, we propose that effective robotic learning should emphasize motion-related knowledge, which is closely tied to low-level actions and is hardware-agnostic, facilitating the transfer of learned motions to actual robot actions. To this end, we introduce* ***Moto****, which converts video content into latent* ***Mo****tion* ***To****ken sequences by a Latent Motion Tokenizer, learning a bridging "language" of motion from videos in an unsupervised manner. We pre-train Moto-GPT through motion token autoregression, enabling it to capture diverse visual motion knowledge. After pre-training, Moto-GPT demonstrates the promising ability to produce semantically interpretable motion tokens, predict plausible motion trajectories, and assess trajectory rationality through output likelihood. To transfer learned motion priors to real robot actions, we implement a co-fine-tuning strategy that seamlessly bridges latent motion token prediction and real robot control. Extensive experiments show that the fine-tuned Moto-GPT exhibits superior robustness and efficiency on robot manipulation benchmarks, underscoring its effectiveness in transferring knowledge from video data to downstream visual manipulation tasks.*

1. Introduction

Recent advancements in Natural Language Processing (NLP) have stemmed from successful autoregressive pretraining on large text corpora via next-word prediction [6, 18, 44, 46, 50]. Pre-trained Large Language Models

†Corresponding Authors.

(LLMs) have shown exceptional performance across various downstream NLP tasks after fine-tuning on smaller datasets. This success opens new opportunity for robotics, which has been limited by the high costs of action-labeled data. Given the abundance of interaction-rich video data [3, 57], we ask: *Can we leverage autoregressive pre-training on video data to improve robot learning?*

The main challenge is finding an appropriate representation for autoregressive pre-training on video data that effectively captures prior knowledge for robot manipulation. Pioneering research in video pre-training for robotics primarily focused on static frames, emphasizing frame-level visual details [9, 19, 54]. However, humans learn skills by observing dynamic environments, focusing on changes in state—what we term motion. Thus, we argue that effective autoregression for robotics should *prioritize motion-related knowledge*, which aligns closely with low-level robot actions and is hardware-agnostic, facilitating the transfer of learned motions to actual robot actions through fine-tuning.

In this work, we introduce **Moto**, which utilizes Latent **Mo**tion **To**kens as a bridging "language" to model visual motions between video frames in an unsupervised manner. As illustrated in Fig. 1, we first train a discrete Latent Motion Tokenizer to produce compact latent motion tokens that capture dynamics between video frames without external supervision. We then pre-train Moto-GPT using a GPT-based architecture to predict the next latent motion token, absorbing motion priors from videos. These learned priors are subsequently transferred to enhance robot manipulation tasks through a co-fine-tuning strategy.

Specifically, as shown in Fig. 2, the Latent Motion Tokenizer encoder employs a VQ-VAE-based architecture [51] to compress two successive video frames into discrete tokens. By regularizing the decoder to reconstruct the second frame from the first frame and the tokens, the tokenizer is trained to effectively capture the changes between video frames, which often arise from motion. Once the tokenizer is trained, we obtain latent motion tokens of every two consecutive frames in a video clip and concatenate them into a sequence to represent the motion trajectory. Subsequently, Moto-GPT is pre-trained on these sequences by predicting the next token based on the initial frame and corresponding language instruction. After this pre-training phase, Moto-GPT is capable of generating plausible trajectories by predicting latent motion tokens autoregressively.

To adapt Moto-GPT for downstream robot manipulation tasks, we concatenate action query tokens with latent motion token chunk at each time step for co-fine-tuning on action-labeled robot data. The action query tokens are processed by a learnable module to predict low-level actions, while the motion tokens are fine-tuned using the original next-token prediction mechanism. This co-fine-tuning strategy effectively transfers abstract intentions in learned motion priors into precise action execution, allowing the model to utilize the inherent knowledge of the pre-trained Moto-GPT for successful manipulation.

We conduct extensive experiments to validate our claims from various perspectives: (1) **Latent Motion Token as an Interpretable Motion Language**: Experiments show that latent motion tokens encapsulate compact and expressive representations of motion, effectively reconstructing and understanding motion trajectories in videos. (2) **Pre-trained Moto-GPT as a Useful Motion Prior Learner**: Results indicate that the pre-trained Moto-GPT achieves promising outcomes in predicting plausible motion trajectories and assessing the rationality of robot trajectories based on output likelihood. (3) **Fine-tuned Moto-GPT as an Effective Robot Policy**: The fine-tuned Moto-GPT demonstrates significant performance improvements over counterparts trained without motion priors, especially with limited training data, highlighting its effectiveness in transferring learned motion knowledge to robot manipulations.

In summary, our contributions are threefold as below:

- Introduction of Latent Motion Tokens, which model visual motions between video frames in an unsupervised manner, serving as a bridging "language" for autoregressive pre-training to enhance robot learning.
- Pre-training of Moto-GPT through next latent motion token prediction on video data, enabling the model to learn useful motion priors without requiring action annotations.
- Implementation of a co-fine-tuning strategy to successfully transfer learned motion priors to actual robot manipulations, with the fine-tuned model showing competitive performance on robotic benchmarks.

We believe the vast reservoir of interaction-rich knowledge in video data presents a crucial opportunity for advancing robot learning and hope this paper inspires further exploration of effective autoregressive representations for acquiring valuable priors through pre-training, ultimately enhancing robotic capabilities.

2. Related Work

Vision-Language-Action Models. Recent studies have increasingly employed transformers as unified vision-language-action (VLA) architectures to generate robot actions from sequential observations and language instructions [5, 25, 48]. Inspired by the success of pre-training in vision-language transformers [1, 6, 36, 44], VLA model pre-training has gained traction. One approach fine-tunes policy models from powerful vision-language models pre-trained on large image-text datasets [16, 32, 62]. Another explores training generalist policy models on diverse cross-embodiment robot data with action labels [15, 28, 42, 52]. In contrast, our work aims to enhance VLA models through generative pre-training on video data, which offers richer interaction details than text and images and requires no

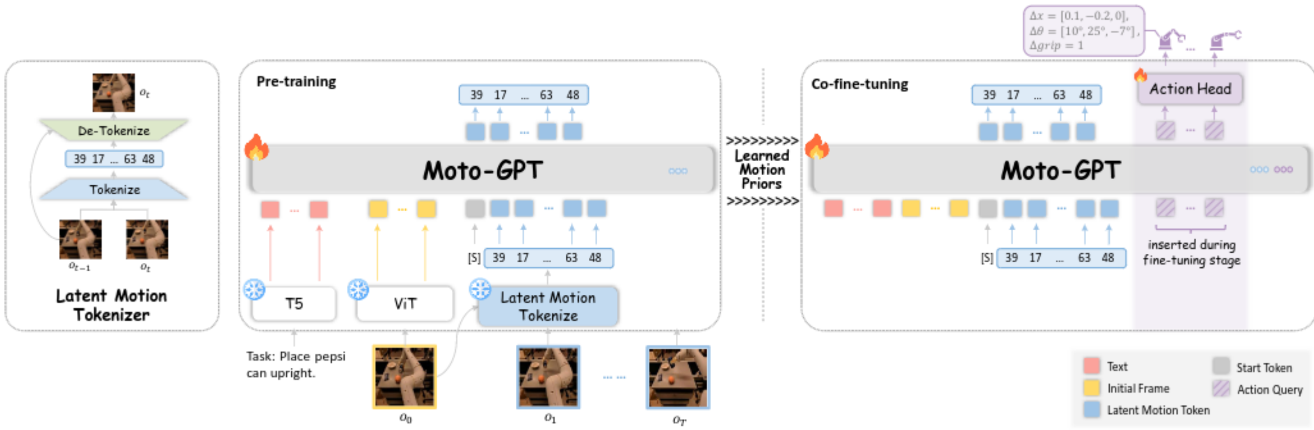

Figure 2. Overview of Moto's three training stages: (1) The Latent Motion Tokenizer encodes key visual motions between video frames into compact latent tokens in an unsupervised manner using pure video data. (2) Moto-GPT is pre-trained with autoregressive motion token prediction to learn motion priors from video-instruction pairs. (3) Moto-GPT is co-fine-tuned on action-labeled trajectories to predict robot actions based on the output of learnable action query tokens while maintaining the next-motion-token prediction objective.

hardware-specific labels of low-level robot actions. Beyond VLA models, several contributions focus on improving robot manipulation performance. Some extend input observations from single-view RGB images to include multi-perspective views and depth information [8, 35, 59]. Techniques like action chunking and policy diffusion also enhance action precision [13, 22, 27]. Additionally, some works [20, 34] decompose high-level language instructions into latent skills learned through auxiliary training objectives during imitation learning.

Robot Learning from Videos Videos provide rich knowledge about physical dynamics, making them ideal for robot learning. Early works [38, 43] utilized contrastive learning with egocentric videos to enhance visual representations for manipulation. Some studies [4, 17, 29, 30, 33] generate videos or images as intermediate plans for guiding low-level control. Recent research [9, 23, 54] has shifted towards generative video pre-training followed by fine-tuning to create end-to-end policy models. Escontrela et al. [19] pre-trains an autoregressive video prediction model to provide reward signals for reinforcement learning. These works primarily use pixel values or patch-level tokens of video frames as their pretraining target. In contrast, our approach focuses on latent motion tokens as prediction targets, emphasizing key visual motions while decoupling irrelevant details. Additionally, some studies build world models through action-conditioned video generation [21, 55, 56], facilitating reinforcement learning or serving as interactive environments. Notably, Genie [7] proposes unsupervised learning of latent actions from large-scale videos to create a versatile 2D gaming simulator. Our goal, however, is to train a generalized policy model for robot manipulation, which is more complex than developing a 2D gaming simulation environment. Concurrently, Ye et al. [58] pre-train a policy model to predict one-step future latent actions, while Chen et al. [12] use latent actions as intermediate goals for low-level policies. Our approach differs by pre-training an end-to-end policy model to autoregressively predict a trajectory of latent motion tokens for future video clips.

3. Methodology

3.1. Overview

Moto utilizes autoregressive generative pre-training on latent motion token sequences to learn motion priors from videos, followed by co-fine-tuning on action-labeled data for robot control. As illustrated in Figure 2, Moto consists of three stages: 1) unsupervised training of the Latent Motion Tokenizer, 2) pre-training of the generative model Moto-GPT, and 3) co-fine-tuning for robot action policy. In Sec 3.2, we detail the Latent Motion Tokenizer, which encodes visual dynamics into quantized latent motion tokens. We also describe the training procedures for Moto-GPT, including motion token autoregressive pre-training in Sec 3.3 and supervised co-fine-tuning in Sec 3.4. Implementation details can be found in the Supplementary Material.

3.2. Latent Motion Tokenizer

The Latent Motion Tokenizer, as shown in Figure 3, learns a latent "language" to capture essential visual motions between successive video frames[1] in an unsupervised manner. The architecture follows a standard auto-encoder design for motion tokenization and detokenization. The tokenization

[1]To ensure significant visual differences, we down-sample the original video by a certain rate.

employs an M-Former, a multi-layer transformer that extracts motion features from the last-layer patch features of the current frame o_t and the preceding frame o_{t-1} using a frozen pre-trained ViT encoder [24]. We concatenate 8 learnable query embeddings with these patch features as additional input to the M-Former, where the queries interact through self-attention layers. The output query features are then processed by a VQ codebook with a vocabulary size of 128 to produce discrete latent motion tokens.

For de-tokenization, we use a ViT Decoder for image reconstruction, which takes the linearly embedded patches of o_{t-1} and recovers the pixel values for o_t based on the latent motion tokens. An MLP projects the concatenated quantized embeddings of the latent motion tokens into a compact embedding (1 token), which is added to each input patch embedding. This conditional embedding acts as an information bottleneck between the encoder and decoder, enabling the ViT Decoder to capture nuanced changes between frames and accurately transform o_{t-1} into o_t.

The components of the Latent Motion Tokenizer are jointly optimized using the standard VQ-VAE objective [51], which includes reconstruction loss, vector quantization loss, and commitment loss. We specifically use the MSE loss between the output pixel values from the ViT Decoder and the ground-truth pixel values of o_t as the reconstruction loss. Once trained, the Latent Motion Tokenizer is frozen to produce unified sequential motion representations for videos through "bi-frame" tokenization. Additionally, with the initial observation and specified latent motion tokens, the decoder can function as a "simulator" to generate rollouts for visualizing environmental changes.

3.3. Motion Token Autoregressive Pre-training

With the Latent Motion Tokenizer, Moto-GPT is allowed to learn about diverse visual motions from videos, using latent motion tokens as a bridging language. As shown in Figure 2, Moto-GPT is pre-trained with a next-motion-token prediction objective. For a video clip $[o_0, o_1, ..., o_T]$, we derive a chunk of latent motion tokens for each pair of consecutive frames, concatenating them chronologically to form a sequence. Moto-GPT employs a GPT-style transformer for autoregression on these motion token trajectories. Additionally, we prepend the text features from the instruction and the visual features from the initial video frame as input prompts. The pre-training objective maximizes the likelihood of the ground-truth latent motion token sequence given the language instruction and the initial video frame:

$$\mathcal{L}_{motion} = -\sum_{i=1}^{M} \log P(m_i | \boldsymbol{l}, \boldsymbol{v}, \boldsymbol{m}_{<i}; \boldsymbol{\Theta}), \qquad (1)$$

where $\boldsymbol{l}$ and $\boldsymbol{v}$ are text and visual features from the frozen pre-trained T5 [47] and ViT [24] models, respectively. $\boldsymbol{m}_{<i}$ represents the latent motion tokens preceding the current token m_i, and $\boldsymbol{\Theta}$ denotes the trainable model parameters. Here, $M = K * T$, where K is the number of tokens for motion between successive frames and T is the video length.

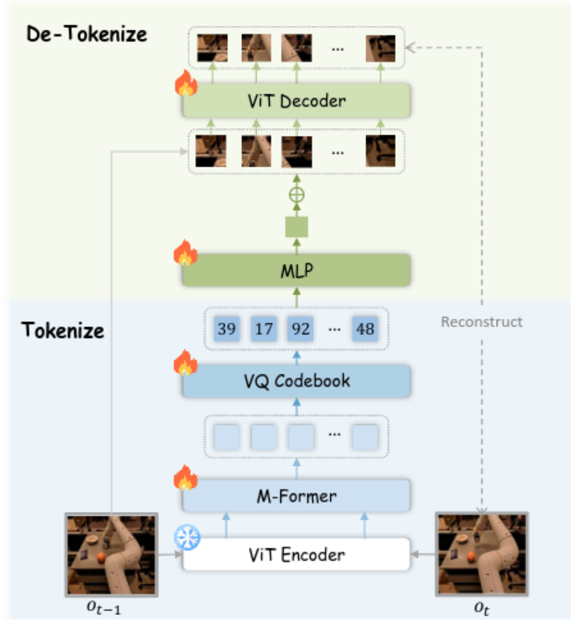

Figure 3. The Latent Motion Tokenizer produces discrete motion tokens from two consecutive video frames. It regularizes the decoder to reconstruct the second frame based on the first one and the discrete tokens, effectively capturing the motion between frames.

3.4. Co-fine-tuning for Robot Manipulation

After pre-training, Moto-GPT can anticipate future trajectories by generating latent motion tokens based on language instructions and initial observations. This process resembles the policy inference of real robots if we take the codebook of latent motion tokens as an abstract action space. However, a gap remains in achieving precise robot control.

To address this, during fine-tuning, we introduce special action query tokens into Moto-GPT's input, enabling the generation of real robot actions through a flexible action head, as illustrated in the right part of Figure 2. Specifically, N query tokens are added after the latent motion token chunk at each time step, where N corresponds to the number of robot actions occurring between two video frames. The fine-tuning stage follows the same causal mask mechanism as pre-training in general. Nevertheless, the latent motion tokens do not attend to the newly inserted action query tokens to stay consistent with the pre-training setting. Besides, we randomly mask 50% of the attention from action query tokens to latent motion tokens, allowing knowledge transfer while reducing dependency on ground-truth conditions. This also improves inference efficiency, enabling direct queries to Moto for real actions without generating latent motion tokens. This can be achieved by using padding tokens as placeholders for latent action tokens, blocking attention from action query tokens to these placeholders.

An MLP-based action head projects the output hidden state of each action query token into the real robot action space. We apply Smooth-L1 loss for continuous action components, such as positional (Δx) and rotational ($\Delta\theta$) displacements, and Binary Cross Entropy (BCE) loss for binary components, like the gripper's open/close state ($\Delta grip$)[2]. The total action loss $\mathcal{L}_{action}$ is defined as:

$$\mathcal{L}_{action} = \mathcal{L}(\Delta x) + \mathcal{L}(\Delta\theta) + \mathcal{L}(\Delta grip) \qquad (2)$$

We retain the training objective for latent motion token prediction to ensure Moto-GPT retains the motion priors learned from videos. Thus, the overall loss function for the fine-tuning stage is:

$$\mathcal{L}_{ft} = \mathcal{L}_{motion} + \mathcal{L}_{action} \qquad (3)$$

4. Experiment Setup

4.1. Benchmarks and Datasets

We use SIMPLER [31] and CALVIN [40] as the main evaluation benchmarks for robot manipulation.

SIMPLER. On the SIMPLER benchmark, we focus on three tasks concerning the Google Everyday Robot embodiment: `Pick Coke Can`, `Move Near`, and `Open/Close Drawer`, as illustrated in Figure 4. The "Pick Coke Can" task involves grasping and lifting the empty coke can in three different orientations: horizontal laying, vertical laying, and standing. The "Move Near" task places 3 (out of 8) objects in a triangle pattern on the tabletop and instructs the robot to move a designated source object near another object as the target. We utilize a subset of Open-X-Embodiment [52] to train the Latent Motion Tokenizer and pre-train Moto-GPT, which consists of 109k real-world trajectory videos [5, 10, 14, 37, 39, 41, 45, 49, 53, 60, 61] across various embodiments. For fine-tuning Moto-GPT, we use 73k action-labeled expert trajectories from the RT-1 Robot Action dataset [5].

CALVIN. On the CALVIN benchmark [40], we assess long-horizon task completion with the Franka Emika Panda robot, requiring the robot to consecutively complete 5 out of 34 manipulation tasks in each trial. There are four different environments (A, B, C, D), each containing a desk with a sliding door, a drawer, differently colored blocks, a

Figure 4. Illustration of the evaluation tasks in SIMPLER [31].

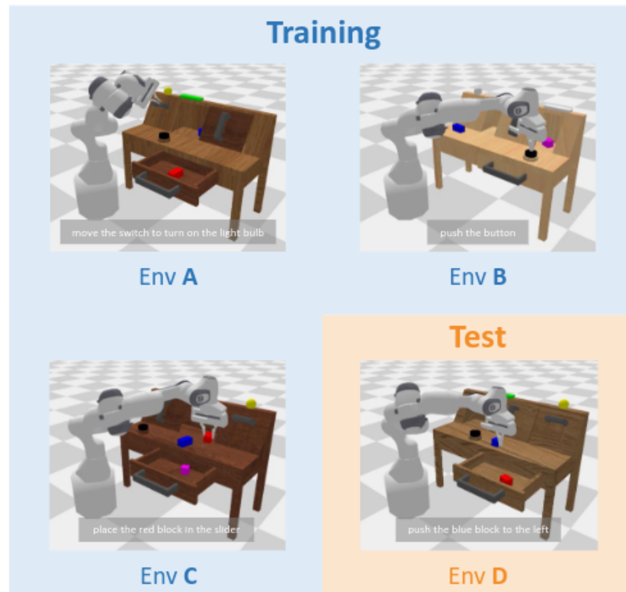

Figure 5. Illustration of the four different environments in CALVIN, adapted from the original figure in Mees et al. [40].

button that toggles an LED, and a switch controlling a lightbulb. As shown in Figure 5, the environments differ in the textures of the desk, and the positions of all static elements including the sliding door, the drawer, the LED button, and the lightbulb switch. We conduct experiments under the most challenging ABC$\longrightarrow$D setting, i.e., training on data from environments A, B, and C while zero-shot testing in D. Specifically, we use all play videos from environments A, B, and C to train the Latent Motion Tokenizer, with 35% of the data (18k trajectory videos) containing language annotations for pre-training Moto-GPT. 18k expert trajectories with language annotations and action labels from environments A, B, and C are used for fine-tuning Moto-GPT.

4.2. Compared Models

SIMPLER. On the SIMPLER benchmark, we compare Moto-GPT with four representative models pre-trained with Open-X-Embodiment datasets:

- **RT-1-X** [5] uses a transformer backbone to output tokenized actions with a FiLM EfficientNet to fuse language

[2] The action space may vary with different robot embpdiments. For example, the Google Everyday Robot uses a continuous value for gripper extension, necessitating Smooth-L1 loss for $\Delta grip$.

and 6 history images into token inputs.
- **RT-2-X** [62] adapts the pre-trained large vision-language model (VLM), PaLI-X (55B) [11], into a robot policy by casting tokenized actions into text tokens.
- **Octo-Base** [42] employ a transformer architecture to process language and image tokens, with a diffusion-based action head to produce actions.
- **OpenVLA** [28] builds on a pre-trained Prismatic-7B [26] VLM backbone for robot action prediction.

CALVIN. On the CALVIN benchmark, we select the following baseline models that leverage pre-training strategies to improve robot manipulation performance:
- **SuSIE** [4] pre-trains an image editing model to generate the goal image, which is fed into a low-level policy for action prediction.
- **RoboFlamingo** [32] is a robot policy model adapted from OpenFlamingo [2], a large VLM pre-trained on extensive vision-language corpus.
- **GR-1** [54] pre-trains a GPT-style transformer to directly predict the pixel values of a single-step future observation for each input observation.
- **MT-R3M** [54] is a variation of GR-1, which leverages the pre-trained robot visual encoder R3M [43] to encode observation images.

Ablations of Moto-GPT. We also study the following variations of Moto-GPT as optional baselines:
- **Moto w/o Motion Token** shares the same backbone with Moto-GPT but is trained from scratch on action-labeled robot data without latent motion tokens.
- **Moto-IML** undergoes the same pre-training stage as Moto-GPT. It keeps latent motion tokens in the input sequence but ignores the next-motion-token-prediction loss during the fine-tuning stage.
- **Moto-DM** is pre-trained in the same way as Moto-GPT but completely discards latent motion tokens in the input sequence during fine-tuning.

4.3. Training Details

Latent Motion Tokenizer. The implementation details for the trainable modules of the Latent Motion Tokenizer are summarized in Table 1. We use the hyperparameters listed in Table 2 to train this model on four A100-40G GPUs. To facilitate the learning of latent motion tokens, we downsample the original videos in the training dataset, ensuring that the visual motion between frames is sufficiently distinct. Specifically, for videos from the Open-X-Embodiment datasets, we sample one frame every three frames (i.e., $\Delta t = 3$) and train the Latent Motion Tokenizer for 350k steps. For videos from the CALVIN dataset, we adopt a sampling rate of one frame every five frames ($\Delta t = 5$) and train the model for 150k steps.

Table 1. Implementation details of the Latent Motion Tokenizer.

Component	Parameter	Value
M-Former	num_queries	8
	num_layers	4
	hidden_size	768
	num_heads	12
ViT Decoder	patch_size	16
	num_layers	12
	hidden_size	768
	num_heads	12
VQ Codebook	num_codes	128
	latent_dim	32

Table 2. Training hyperparameters for Latent Motion Tokenizer.

Parameter	Value
batch_size	256
optimizer	AdamW
lr_max	1e-4
lr_schedule	cosine decay
weight_decay	1e-4
warmup_steps	1000

Table 3. Implementation details of Moto-GPT.

Component	Parameter	Value
GPT backbone	num_layers	12
	hidden_size	768
	num_heads	12
Action Head	num_layers	2
	hidden_size	384

Table 4. Training hyperparameters for Moto-GPT.

Parameter	Value
batch_size	512
optimizer	AdamW
lr_max	1e-4
lr_schedule	cosine decay
weight_decay	1e-4
warmup_epochs	1

Moto-GPT. We present the implementation details of Moto-GPT in Table 3, where the Action Head is included only during the fine-tuning phase. Moto-GPT handles a maximum video length of three frames, and the video downsampling rate applied during both the pre-training and fine-tuning stages is consistent with the rate used for training the Latent Motion Tokenizer. When fine-tuning Moto-GPT across different benchmarks, the number of action query tokens inserted after the latent motion tokens at each time step varies. Specifically, for the SIMPLER benchmark, we insert three action query tokens, whereas for the CALVIN benchmark, we insert five. For pre-training, Moto-GPT is trained for 10 epochs using eight A100-40G GPUs, with the relevant hyperparameters outlined in Table 4. The hyperparameters for fine-tuning remain consistent with those used during pre-training, except for the number of epochs. We fine-tune Moto-GPT for three epochs on the RT1-Robot-Action dataset and 18 epochs on the CALVIN dataset, utilizing four A100-40G GPUs.

5. Experiments

To comprehensively evaluate the effectiveness of Moto, we study three key experimental questions:

- **Q1 (Interpretability):** Does the Latent Motion Tokenizer learn interpretable latent motion tokens that effectively represent visual motions from videos?
- **Q2 (Motion Priors):** Does Moto-GPT gain meaningful prior knowledge of motion trajectories through autoregressive pre-training on latent motion token sequences?
- **Q3 (Performance):** Can the motion priors be transferred to enhance policy performance in robot manipulation benchmarks through efficient fine-tuning?

5.1. Latent Motion Token as an Interpretable Motion Language

As illustrated in Figure 6, the next frame reconstructed by the Latent Motion Tokenizer using ground-truth latent motion tokens is authentic, effectively capturing the key dynamics between the initial frame and the ground-truth next frame. This suggests that latent motion tokens can represent fine-grained motion details, and the Latent Motion Tokenizer's decoder serves as a qualified simulator for visualizing environmental changes.

Figure 7 further explores the controllability and consistency of latent motion tokens. Each row demonstrates that different token chunks produce visual motions with varying orientations and scales relative to the initial frame. Conversely, within each column, identical token chunks yield similar effects on the resulting positions and postures across different starting observations. By concatenating latent motion token chunks for every two consecutive frames from a video, we create a sequential representation of motion trajectories, akin to natural language context. As shown in Figure 8, this representation can be applied to different initial observations, generating contextualized motion trajectories and highlighting the potential of latent motion tokens as a unified language interface for guiding imitation learning.

Table 5 presents quantitative evidence of the semantic interpretability of latent motion tokens. We trained a video classifier using ViT patch features from the initial frame, alongside concatenated latent motion tokens for the subsequent seven frames to predict semantic labels for 34 tasks from the ABC⟶D split of the CALVIN dataset. The classifier utilizing latent motion tokens achieved an accuracy of 79.7%, comparable to the performance of a classifier using ViT patch features for all eight frames, despite the former reducing input features for each subsequent frame from 196 tokens to just 8. In contrast, classifiers relying solely on the initial frame or a repeated initial frame sequence struggled, achieving accuracies below 30%. These results indicate that, despite training without text or action labels, latent motion tokens provide a highly compact and expressive representation of visual motions, serving as an interpretable language of motion linked to high-level semantics.

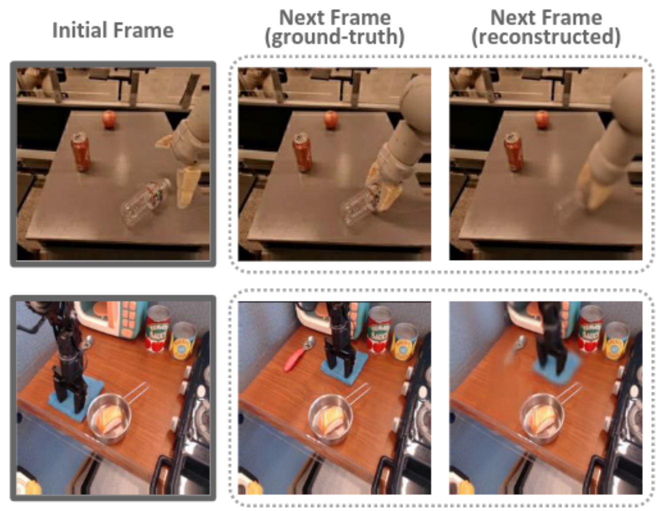

Figure 6. Qualitative examples of reconstruction results, where discrete motion tokens obtained from the Latent Motion Tokenizer based on the initial and next frame, are fed into the decoder along with the initial frame to reconstruct the target frame.

5.2. Moto-GPT as a Useful Motion Prior Learner

The pre-training stage of Moto-GPT involves autoregression on video data using latent motion tokens, enabling it to predict motion trajectories based on initial observations and various language prompts, as illustrated in Figure 9. Table 6 presents the top-k accuracy of Moto-GPT in predicting ground-truth latent motion tokens from a 128-size codebook on the validation splits of the pre-training datasets. These results demonstrate Moto-GPT's effective acquisition of prior knowledge for motion trajectory prediction, which is crucial for robot action inference based on human instructions. Thus, the learned motion priors hold the potential to benefit downstream robotic tasks.

Additionally, latent motion tokens allow Moto-GPT to interpret trajectory videos as compact token sequences and evaluate their rationality through the autoregressive likelihood defined in Eq. 3.3. Figure 10 illustrates the potential

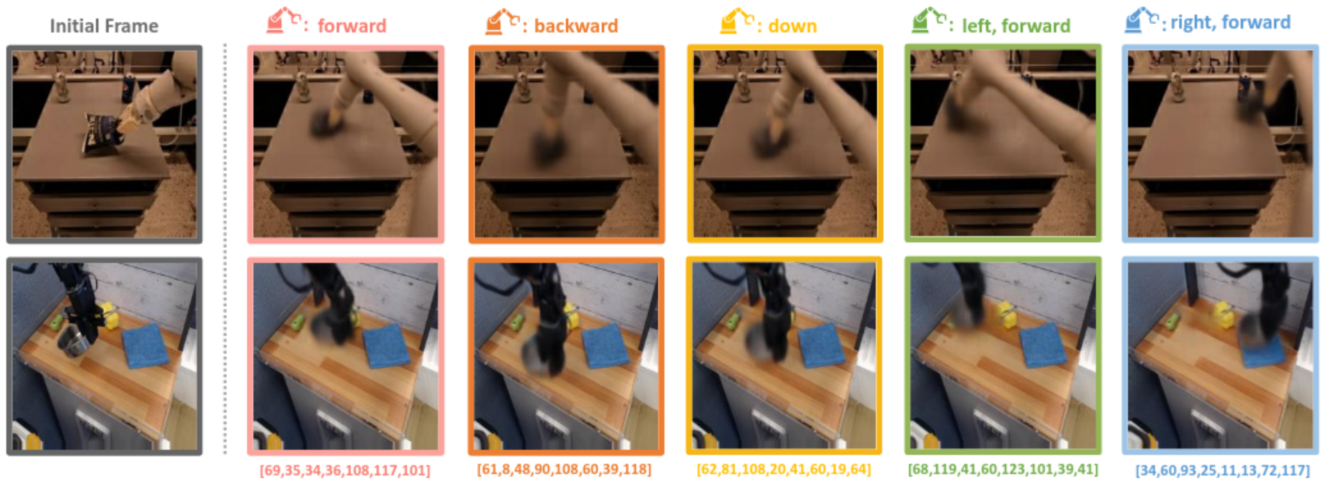

Figure 7. Visualization of latent motion token interpretability. Each row displays reconstructed frames from the same initial frame using different latent motion tokens, while each column shows frames reconstructed from the same latent motion tokens with varying initial frames. The latent motion tokens exhibit consistent (see columns) and discriminative (see rows) semantics, despite being trained in an unsupervised manner.

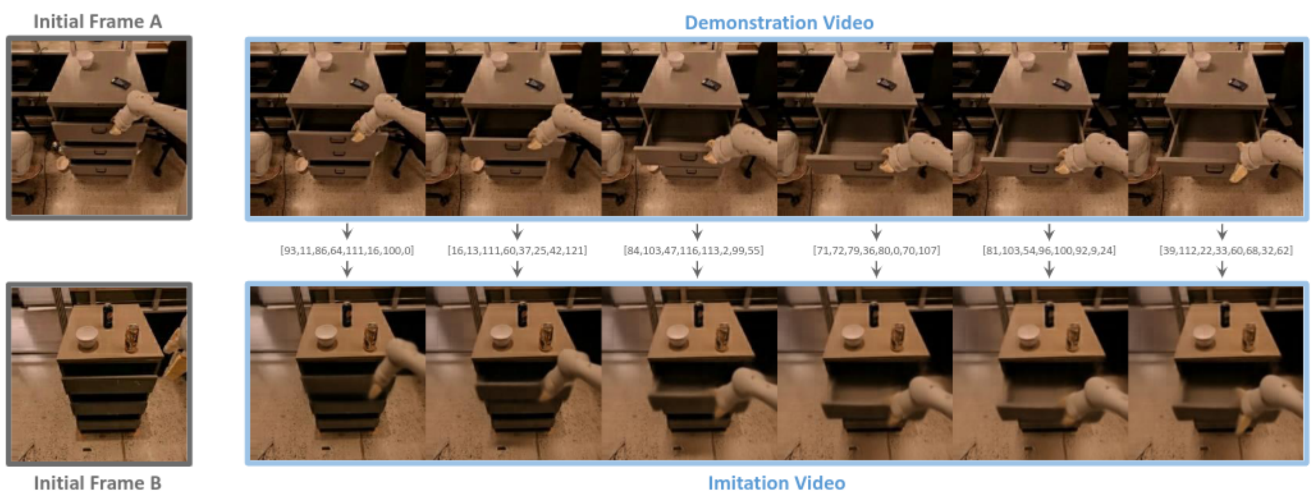

Figure 8. Video imitation generation via latent motion tokens, where a sequence of latent motion tokens from a demonstration video are extracted by the Latent Motion Tokenizer and are decoded into a new video. This generated video is based on a different initial frame while preserving the original robot movement semantics.

Table 5. Video classification accuracy with varied representations.

Video Representation	Semantic Acc.
Initial frame	0.292
Initial frame repeated by 8 times	0.283
Initial frame + 7 subsequent frames	0.828
Initial frame + 7 latent motion token chunks	0.797

of using Moto's log-likelihoods as a reward signal for trajectory videos, indicating how well a trajectory aligns with Moto-GPT's distribution and measuring the temporal consistency of behavior. To assess this, we collected 98 video triplets in CALVIN using the baseline policies and a random policy. Each triplet consists of three types of trajectory videos originating from the same environment state. The averaged log-likelihoods for each trajectory type at each sequence step, shown in Figure 10, clearly differentiate successful trajectories from failures and random attempts.

5.3. Moto-GPT as an Effective Robot Policy

Overall Performance. After fine-tuning, we evaluated Moto-GPT[3] against baseline models on the SIMPLER and

[3]For simplicity, we will refer to Moto-GPT as Moto in the following experimental tables and figures.

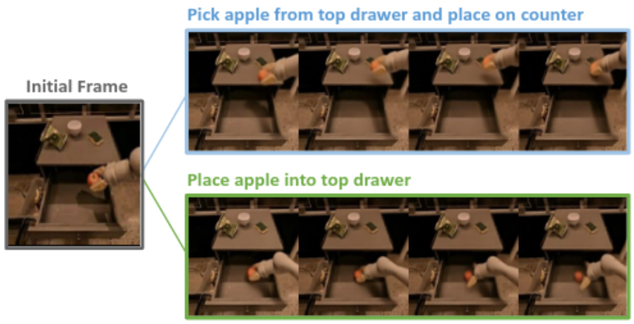

Figure 9. Visualization of video trajectories generated from a sequence of latent motion tokens, which are predicted by the pre-trained Moto-GPT given different language instructions.

Table 6. Top-K motion token prediction accuracy of Moto-GPT.

Dataset	Top-5	Top-10	Top-20
Oepn-X-Embodiment	0.521	0.698	0.853
Calvin (ABC⟶D)	0.298	0.518	0.768

CALVIN benchmarks, as shown in Tables 7 and 8. Overall, Moto-GPT outperforms the baselines on both benchmarks. Notably, on SIMPLER, Moto-GPT surpasses larger models like RT-2-X (PaLI-X 55B) and OpenVLA (Prismatic-7B), despite having only 98M parameters for the GPT-style backbone. Moto-GPT also shows strong generalization in the unseen CALVIN environment. The baseline models utilize various pre-training strategies: SuSIE employs a pre-trained image-editing model for goal image generation, RobotFlamingo is initialized from a large vision-language model, MT-R3M uses a pre-trained robot visual encoder, and GR-1 predicts future pixel values based on input observations. In contrast, Moto-GPT, pre-trained through autoregressive motion token prediction, achieves competitive performance despite relying solely on RGB images from a static camera. This is particularly impressive when compared to GR-1, which uses images from both static and gripper cameras along with proprioceptive robot state data. Our findings support the idea that focusing on motion-related dynamics rather than frame-level visual details is a more effective approach for learning from videos. Additionally, Moto-GPT significantly outperforms its variant trained from scratch on action-labeled robot data without latent motion tokens (Moto w/o Motion Token). This highlights the effectiveness of our latent-motion-token-based pre-training and co-fine-tuning strategy in enhancing policy performance for practical robot manipulation tasks.

Data Efficiency. Moto-GPT's pre-training relies solely on videos, eliminating the need for supervised robot data with action labels. This allows for pre-training on large-scale, easily accessible video datasets, followed by fine-tuning with smaller-scale action-labeled trajectories for policy adaptation. To simulate a low-resource scenario, we fine-tune Moto-GPT with varying proportions of action-labeled data and evaluate its performance on CALVIN (ABC⟶D). As shown in Figure 11, the performance gap between Moto-GPT and its variant fine-tuned from scratch without latent motion tokens (Moto w/o Motion Token) widens with limited fine-tuning data. Notably, Moto-GPT achieves a success rate of 52.5% with just 1% of the labeled data, while Moto w/o Motion Token records a 0% success rate. This highlights Moto-GPT's efficiency in adapting to produce accurate actions and its potential to enhance performance in downstream robot manipulation tasks by leveraging larger pre-training video datasets.

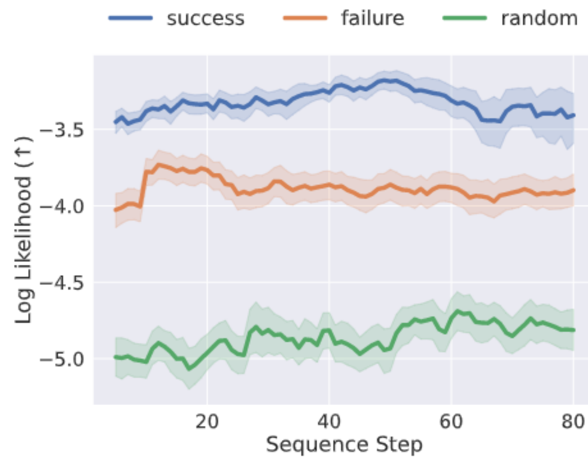

Figure 10. Moto-GPT distinguishes successful, failed, and random robot trajectories using log-likelihoods, enabling effective assessment of trajectory rationality and potential reward signals.

Ablations on Policy Fine-tuning Methods. In Figure 12, we evaluate the effectiveness of Moto's co-fine-tuning strategy. Moto-IML and Moto-DM share the same pre-training approach as Moto-GPT but differ in their fine-tuning methods. Specifically, Moto-IML omits the loss term for latent motion token prediction, while Moto-DM excludes latent motion tokens from the input sequence entirely. When compared to Moto w/o Motion Tokens, which is trained from scratch without latent motion tokens, both Moto-IML and Moto-DM show performance improvements due to the motion priors gained during pre-training. However, they still fall short of Moto-GPT's performance. This highlights the importance of retaining latent motion tokens in the sequence, allowing action query tokens to transfer knowledge through direct attention. Furthermore, co-fine-tuning for latent motion token prediction helps preserve the learned motion priors in Moto-GPT.

Table 7. SIMPLER evaluation results of models pre-trained on Open-X-Embodiment [52] datasets. The "Overall" column reports the success rate averaged across the sub-tasks of all task types.

Method	Pick Coke Can				Move Near	Open / Close Drawer			Overall
	Horizontal	Vertical	Standing	Average	Average	Open	Close	Average	Average
RT-1-X [5]	**0.820**	0.330	0.550	0.567	0.317	**0.296**	**0.891**	**0.597**	0.534
RT-2-X [62]	0.740	**0.740**	0.880	**0.787**	**0.779**	0.157	0.343	0.250	0.607
Octo-Base [42]	0.210	0.210	0.090	0.170	0.042	0.009	0.444	0.227	0.169
OpenVLA [28]	0.270	0.030	0.190	0.163	0.462	0.194	0.518	0.356	0.248
Moto	**0.820**	0.500	**0.900**	0.740	0.604	0.130	0.732	0.431	**0.614**
Moto w/o Motion Token	0.600	0.190	0.740	0.503	0.554	0.000	0.796	0.398	0.480

Table 8. Comparison of models adopting different pre-training techniques on CALVIN (ABC⟶D). Avg. Len. is a comprehensive metric indicating the average number of tasks accomplished in a row across 1,000 trial sequences. "Static RGB" and "Gripper RGB" denote the RGB images from a static camera or a gripper view, respectively. "Proprio" is short for the proprioceptive robot state.

Model	Observation Space	Tasks competed in a row (1000 chains)					
		1	2	3	4	5	Avg. Len.
SuSIE [4]	Static RGB	0.870	0.690	0.490	0.380	0.260	2.69
RoboFlamingo [32]	Static RGB + Gripper RGB	0.824	0.619	0.466	0.331	0.235	2.47
MT-R3M [54]	Static RGB + Gripper RGB + Proprio	0.529	0.234	0.105	0.043	0.018	0.93
GR-1 [54]	Static RGB + Gripper RGB + Proprio	0.854	0.712	0.596	**0.497**	**0.401**	3.06
Moto	Static RGB	**0.897**	**0.729**	**0.601**	0.484	0.386	**3.10**
Moto w/o Motion Token	Static RGB	0.779	0.555	0.380	0.256	0.167	2.14

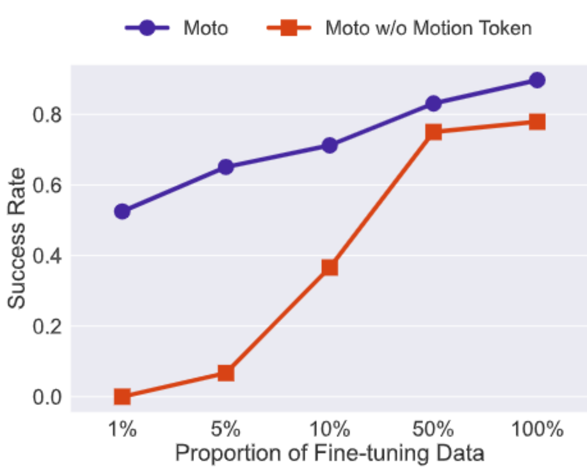

Figure 11. Task success rate of models fine-tuned with different proportions of data on CALVIN (ABC⟶D).

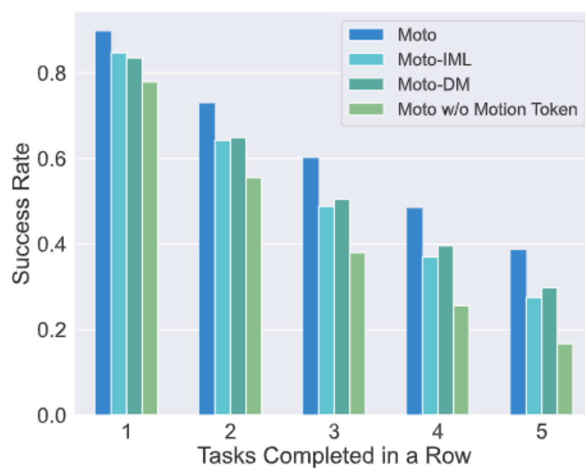

Figure 12. Ablations of Moto-GPT on CALVIN (ABC⟶D).

6. Conclusion and Discussion

This paper introduces Moto, a novel method that uses latent motion tokens as a "language" interface to bridge generative pre-training on video data with precise robot control. Moto opens several exciting avenues for future work.

Firstly, Moto demonstrates the feasibility of learning a unified language to interpret diverse visual dynamics from videos, eliminating the need for hardware-specific action labels. The latent motion trajectories tokenized from videos provide a rich resource for models to learn motion priors closely related to low-level actions. While we currently mainly use robot videos to train the Latent Motion Tokenizer, the learned latent motion tokens demonstrate the potential to produce consistent visual motions across varied contexts and embodiments. We believe a similar approach could be applied to human motion representation, enabling models to learn a wealth of world knowledge from Internet-scale human videos.

Besides, the Moto-GPT pre-trained on videos tokenized into latent motion token sequences and fine-tuned on action-labeled trajectories, effectively transfers motion priors learned from videos to actual robot action prediction. This is particularly beneficial in low-resource scenarios. Future work could involve scaling up pre-training video data and optimizing fine-tuning to improve model performance on downstream robot tasks further.

Moreover, while Moto is primarily utilized to enhance imitation learning for robot manipulation tasks, it shows potential as a reward model for measuring trajectory rationality and as a vivid environment simulator. Future research could explore Moto's use in improving the robustness of reinforcement learning agents and extending its application to a wider range of robotic tasks, such as navigation and locomotion, to develop a more versatile robot action policy.

References

[1] 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. *NeurIPS*, 2022. 2

[2] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 6

[3] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22861–22872, 2024. 2

[4] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 6, 10

[5] 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, 5, 10

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901. Curran Associates, Inc., 2020. 1, 2

[7] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. 3

[8] Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[9] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 2, 3

[10] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home. 5

[11] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023. 6

[12] 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. *arXiv preprint arXiv:2411.00785*, 2024. 3

[13] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 3

[14] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. Clvr jaco play dataset, 2023. 5

[15] 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 *8th Annual Conference on Robot Learning*, 2024. 2

[16] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, pages 8469–8488. PMLR, 2023. 2

[17] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[18] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1

[19] Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2, 3

[20] Divyansh Garg, Skanda Vaidyanath, Kuno Kim, Jiaming Song, and Stefano Ermon. Lisa: Learning interpretable skill abstractions from language. *Advances in Neural Information Processing Systems*, 35:21711–21724, 2022. 3

[21] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. 3

[22] Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. BAKU: An efficient transformer for multi-task policy learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[23] Haoran He, Chenjia Bai, Ling Pan, Weinan Zhang, Bin Zhao, and Xuelong Li. Learning an actionable discrete diffusion policy via large-scale actionless video pre-training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 4

[25] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, pages 14975–15022, 2023. 2

[26] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024. 6

[27] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *ICRA 2024 Workshop—Back to the Future: Robot Learning Going Probabilistic*, 2024. 3

[28] 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, 6, 10

[29] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[30] Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially annotated data via multi-modal goal conditioned policy. *arXiv preprint arXiv:2408.14368*, 2024. 3

[31] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. 5

[32] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 6, 10

[33] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024. 3

[34] Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16467–16476, 2024. 3

[35] Fanfan Liu, Feng Yan, Liming Zheng, Chengjian Feng, Yiyang Huang, and Lin Ma. Robouniview: Visual-language model with unified view representation for robotic manipulation. *arXiv preprint arXiv:2406.18977*, 2024. 3

[36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 2

[37] Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multi-stage cable routing through hierarchical imitation learning. *arXiv pre-print*, 2023. 5

[38] 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 *The Eleventh International Conference on Learning Representations*, 2023. 3

[39] 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 *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1048–1055. IEEE, 2019. 5

[40] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334, 2022. 5

[41] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Con-*

ference on Robotics and Automation (ICRA), London, UK, 2023. 5

[42] Oier Mees, Dibya Ghosh, Karl Pertsch, Kevin Black, Homer Rich Walke, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. 2, 6, 10

[43] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2023. 3, 6

[44] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1, 2

[45] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation, 2021. 5

[46] Alec Radford. Improving language understanding by generative pre-training. 2018. 1

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

[48] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. Featured Certification, Outstanding Certification. 2

[49] Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task agnostic offline reinforcement learning. 2022. 5

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[51] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2, 4

[52] Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, Chelsea Finn, Max Du, Moo Jin Kim, Alexander Khazatsky, Jonathan Heewon Yang, Tony Z. Zhao, Ken Goldberg, Ryan Hoque, Lawrence Yunliang Chen, Simeon Adebola, Gaurav S. Sukhatme, Gautam Salhotra, Shivin Dass, Lerrel Pinto, Zichen Jeff Cui, Siddhant Haldar, Anant Rai, Nur Muhammad Mahi Shafiullah, Yuke Zhu, Yifeng Zhu, Soroush Nasiriany, Shuran Song, Cheng Chi, Chuer Pan, Wolfram Burgard, Oier Mees, Chenguang Huang, Deepak Pathak, Shikhar Bahl, Russell Mendonca, Gaoyue Zhou, Mohan Kumar Srirama, Sudeep Dasari, Cewu Lu, Hao-Shu Fang, Hongjie Fang, Henrik I Christensen, Masayoshi Tomizuka, Wei Zhan, Mingyu Ding, Chenfeng Xu, Xinghao Zhu, Ran Tian, Youngwoon Lee, Dorsa Sadigh, Yuchen Cui, Suneel Belkhale, Priya Sundaresan, Trevor Darrell, Jitendra Malik, Ilija Radosavovic, Jeannette Bohg, Krishnan Srinivasan, Xiaolong Wang, Nicklas Hansen, Yueh-Hua Wu, Ge Yan, Hao Su, Jiayuan Gu, Xuanlin Li, Niko Suenderhauf, Krishan Rana, Ben Burgess-Limerick, Federico Ceola, Kento Kawaharazuka, Naoaki Kanazawa, Tatsuya Matsushima, Yutaka Matsuo, Yusuke Iwasawa, Hiroki Furuta, Jihoon Oh, Tatsuya Harada, Takayuki Osa, Yujin Tang, Oliver Kroemer, Mohit Sharma, Kevin Lee Zhang, Beomjoon Kim, Yoonyoung Cho, Junhyek Han, Jaehyung Kim, Joseph J Lim, Edward Johns, Norman Di Palo, Freek Stulp, Antonin Raffin, Samuel Bustamante, João Silvério, Abhishek Padalkar, Jan Peters, Bernhard Schölkopf, Dieter Büchler, Jan Schneider, Simon Guist, Jiajun Wu, Stephen Tian, Haochen Shi, Yunzhu Li, Yixuan Wang, Mingtong Zhang, Heni Ben Amor, Yifan Zhou, Keyvan Majd, Lionel Ott, Giulio Schiavi, Roberto Martín-Martín, Rutav Shah, Yonatan Bisk, Jeffrey T Bingham, Tianhe Yu, Vidhi Jain, Ted Xiao, Karol Hausman, Christine Chan, Alexander Herzog, Zhuo Xu, Sean Kirmani, Vincent Vanhoucke, Ryan Julian, Lisa Lee, Tianli Ding, Yevgen Chebotar, Jie Tan, Jacky Liang, Igor Mordatch, Kanishka Rao, Yao Lu, Keerthana Gopalakrishnan, Stefan Welker, Nikhil J Joshi, Coline Manon Devin, Alex Irpan, Sherry Moore, Ayzaan Wahid, Jialin Wu, Xi Chen, Paul Wohlhart, Alex Bewley, Wenxuan Zhou, Isabel Leal, Dmitry Kalashnikov, Pannag R Sanketi, Chuyuan Fu, Ying Xu, Sichun Xu, brian ichter, Jasmine Hsu, Peng Xu, Anthony Brohan, Pierre Sermanet, Nicolas Heess, Michael Ahn, Rafael Rafailov, Acorn Pooley, Kendra Byrne, Todor Davchev, Kenneth Oslund, Stefan Schaal, Ajinkya Jain, Keegan Go, Fei Xia, Jonathan Tompson, Travis Armstrong, and Danny Driess. Open x-embodiment: Robotic learning datasets and RT-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition @ CoRL2023*, 2023. 2, 5, 10

[53] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023. 5

[54] 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 *The Twelfth International Conference on Learning Representations*, 2024. 2, 3, 6, 10

[55] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye HAO, and Mingsheng Long. ivideoGPT: Interactive videoGPTs are scalable world models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[56] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour,

Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[57] Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*, 2024. 2

[58] 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. 3

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

[60] Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, Chelsea Finn, and Abhinav Gupta. Train offline, test online: A real robot learning benchmark. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 5

[61] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *6th Annual Conference on Robot Learning (CoRL)*, 2022. 5

[62] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023. 2, 6, 10