CVPR #23

SPARK: Simple Post-training for Adapting pRetrained Knowledge to Robot Control

Anonymous CVPR submission

Paper ID 23

Abstract

*Large scale representation learning has produced public foundation models that provide strong general purpose visual features. However, their pretraining objectives are not designed for robot representation learning, so the resulting embeddings tend to emphasize global semantics rather than the temporally sensitive representations required for robot control. Training robot models from scratch is also undesirable due to the limited scale and high collection cost of robotics data, as well as the substantial computational burden of pre-training. In this paper, we propose **SPARK** (**S**imple **P**ost-training for **A**dapting p**R**etrained **K**nowledge), a post training method that adapts foundation models to robot control. SPARK is built on two principles: dynamics-aware abstraction, which encourages the encoder to derive compact features that preserve information essential for understanding temporal changes and action-relevant scene structure, and knowledge preservation, which aligns patch-level representations with those of the original foundation model to retain useful pretrained semantics. These objectives yield compact visual state representations that remain semantically meaningful while preserving useful prior knowledge. Experiments across multiple robotics benchmarks show that SPARK consistently improves success rates and generalization over vanilla foundation models, and further demonstrate that these gains transfer to real-world robot manipulation.*

1. Introduction

The evolution of robot representation learning [5, 17–19, 21, 23] has primarily centered on modeling dynamics in spatiotemporal visual observations. This line of research has produced representations effective for imitation learning and robot control, where success depends on capturing object motion, scene transitions, and task progress from visual input. In parallel, general-purpose visual representation learning has increasingly emphasized visual-language alignment for transfer. From this perspective, conventional robot models remain limited, being largely optimized for narrow control distributions rather than language-grounded semantics.

Learning robot representations directly from paired video-language data offers a direct route to unifying embodied dynamics with language-level semantics. However, collecting robot vision-action data at scale is expensive and typically limited to laboratory settings, making it difficult to cover diverse scenes, objects, and interactions. This has led prior work to learn visual representations from large-scale video data without robot actions, in order to capture physical dynamics that may benefit downstream robot learning. While such approaches improve scalability and provide temporal supervision, they still lack the broad semantic coverage and vision-language alignment available in foundation models trained on web-scale corpora. These limitations motivate a different strategy that adapts already well-aligned foundation models [20, 22, 24, 26, 28] to the requirements of robot control.

In this paper, we propose **SPARK** (Simple Post-training for Adapting pRetrained Knowledge), a post-training method that adapts foundation models and general-purpose vision models to robot control. The central goal of SPARK is to enable pretrained representations more suitable for control without sacrificing the general visual knowledge acquired during large-scale pretraining. To this end, SPARK is built on two key principles: **dynamics-aware abstraction** and **knowledge preservation**. Dynamics-aware abstraction encourages the encoder to produce compact features that preserve information essential for understanding temporal changes by leveraging the prior knowledge acquired during large-scale pretraining. However, naively enforcing such compact abstraction can degrade useful prior knowledge inherited from large-scale pretraining. Thus, we adopt a retention objective that aligns the patch-wise representations of the adapted encoder with those of the original foundation model.

We empirically demonstrate that SPARK yields consistent improvements across multiple robotics benchmarks.

CVPR #23

Models adapted with SPARK yield higher success rates and better generalization than their vanilla models on average on robot manipulation tasks, highlighting that cost-efficient post-training can substantially enhance the generalizability of the foundation model on robot control. Moreover, experiments on real-world robot tasks show that these gains extend beyond simulation. Consequently, SPARK presents a promising recipe for mitigating the mismatch between foundation models and the visual representations required for embodied intelligence.

2. Related Work

Pretrained Vision and Foundation Models. Large-scale vision models or vision-language models, including CLIP and SigLIP families, are trained on numerous image and text datasets and provide strong zero-shot and few-shot performance [1–4, 10, 11, 20, 22, 26, 28]. These foundation models capture high-level semantics and diverse visual regularities and have become standard backbones for many vision tasks. However, their pre-training objectives are designed for recognition and language alignment rather than robot representation learning, resulting in embeddings suitable for global semantics and caption-level alignment, which may not be effective for temporally sensitive structure within consecutive observations, which is crucial for control. This objective mismatch motivates post-training recipes that align these models with the demands of robotic perception without pre-training from scratch.

Visual Representations in Robotics. In robotic perception, pretrained vision backbones are widely used to encode observations for downstream manipulation, navigation, and policy learning [5, 14]. Several studies have shown that even frozen features from generic encoders can already provide strong representations for imitation learning and control [21]. Beyond direct reuse, robotics-oriented visual pretraining has been explored to further improve transfer by exploiting egocentric viewpoints or temporal structure in robot data. Examples include R3M, which combines time contrastive objectives with language supervision on human-centric egocentric video [19], VIP, which learns value implicit representations that connect initial and goal observations [17], and MAE-style variants such as MVP and VC-1, which curate large ego-centric manipulation corpora for masked reconstruction [5, 18, 23]. However, these works mainly introduce pre-training recipes that require huge computation costs, often with paired language annotations, which are expensive to collect and still much smaller than web-scale image and text data, limiting their accessibility and coverage.

Post-training and Adaptation. Post-training has been used to reduce the cost of training when adapting pretrained encoders to new domains or tasks through post-training. Parameter-efficient adaptation methods introduce small learnable components such as prompts and adapter layers, as in CoOP and CoCoOp, CLIP Adapter and visual prompt tuning, or apply low-rank updates like LoRA [7, 13, 29, 30]. In robotics, a prior work [18] has explored post-training using a generic reconstruction objective based on masked autoencoding [11] approaches to enhance the visual understanding capacity of the original models. Besides, our approach follows the spirit of post-training adaptation but targets the representation more directly. SPARK introduces a conservative summarization loss that trains a single class token to act as a compact state, together with a retention loss that aligns token-level features with those of the original foundation model.

3. Method

3.1. Overview

The goal of SPARK (**S**imple **P**ost-training for **A**dapting p**R**etrained **K**nowledge) is to adapt pretrained foundation models to robot control without sacrificing the broad visual knowledge acquired during large-scale pretraining. To this end, SPARK is built on two complementary principles: **dynamics-aware abstraction** and **knowledge preservation**. Dynamics-aware abstraction encourages the encoder to derive compact features that preserve information essential for understanding temporal changes and action-relevant scene structure. Knowledge preservation regularizes this adaptation so that the learned representation remains aligned with the semantically meaningful features of the original foundation model. Together, these principles enable pretrained representations to become more suitable for robot control without requiring training from scratch.

To realize dynamics-aware abstraction, SPARK trains the encoder so that a compact feature extracted from a reference observation can support the prediction of a temporally offset target scene. Specifically, the encoder processes a reference scene and produces patch tokens together with a class token, where the class token serves as a compact feature. A decoder then reconstructs token representations of the target scene from this compact feature, conditioned on only a small subset of visible target patches as hints. Because accurate prediction requires information that explains how the scene changes over time, this objective encourages the compact feature to preserve temporal and action-relevant structure rather than only static global semantics.

However, adapting the encoder solely through this abstraction objective can distort useful pretrained knowledge. To mitigate this issue, SPARK introduces a retention objective that aligns the patch-wise representations of the adapted

CVPR #23

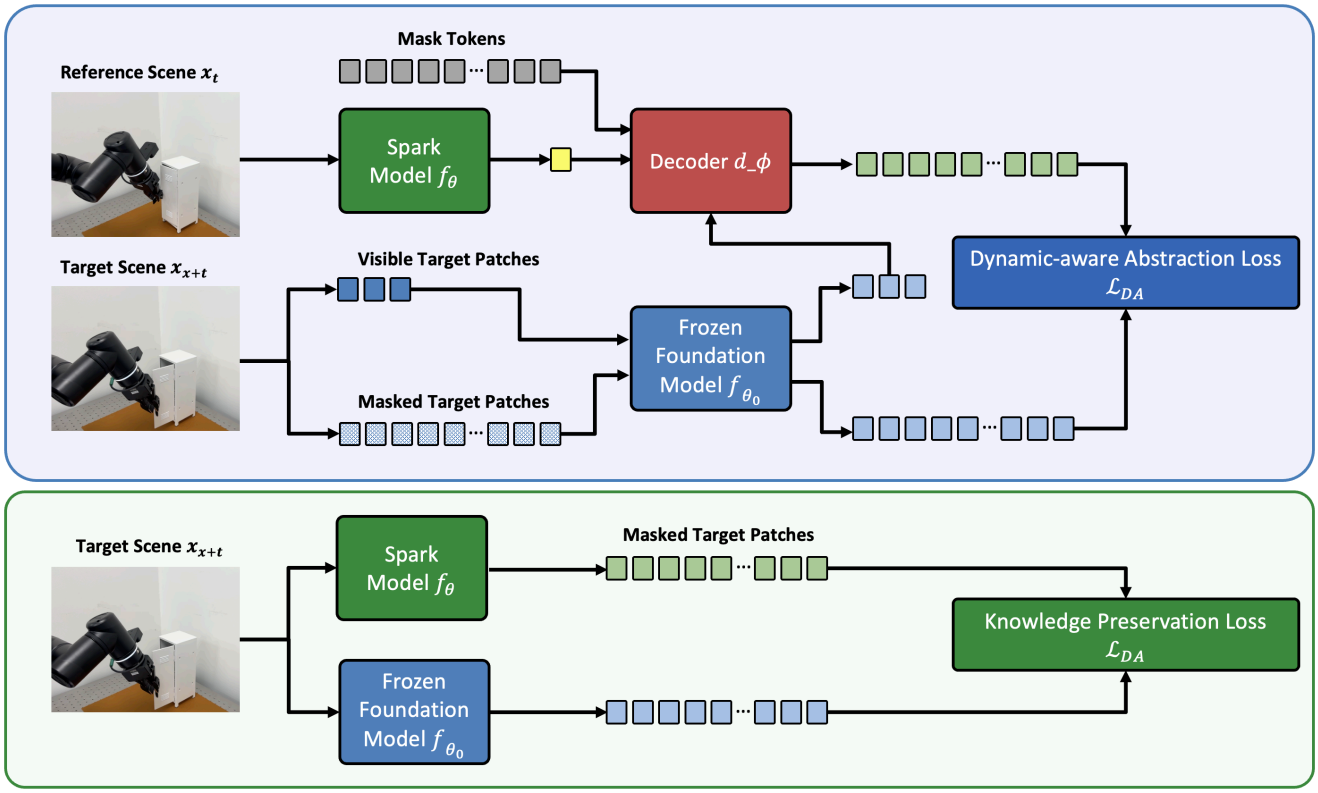

Figure 1. **Overview of SPARK (Simple Post-training for Adapting pRetrained Knowledge).** SPARK aims to encourage conservative summarization of patch-wise well-understood semantics while preserving the prior knowledge of the foundation model obtained during pre-training. Specifically, given a reference scene x_t, the SPARK encoder f_θ encodes the reference scene into patch tokens and a class token. Then, the decoder predicts the target token representations from the class token with hints from a few target patches. The conservative summarization loss $\mathcal{L}_{\text{DA}}$ minimizes the discrepancy between these predicted target token representations and the corresponding target token representations produced by a frozen baseline encoder f_{θ_0}. On the other hand, the retention loss aligns the reference token representations encoded by the SPARK encoder with the reference token representations encoded by the frozen baseline encoder f_{θ_0} at the patch level. This encourages the class token to act as a compact visual state for control while preserving the patch-level semantic knowledge of the original foundation model.

encoder with those of the frozen original foundation model on the same input. This regularization preserves semantically meaningful local features while allowing the compact feature to specialize toward dynamic understanding for control. The overall pipeline is illustrated in Fig. 1.

3.2. Formulation

Encoding procedure. Let f_θ denote the SPARK encoder and let f_{θ_0} denote the frozen foundation model used for initialization. We sample a reference scene $x_t \in \mathbb{R}^{3\times H\times W}$ and a temporally offset target scene $x_{t+k} \in \mathbb{R}^{3\times H\times W}$. We patchify both scenes into N non-overlapping patches, denoted by $\{x_t^i\}_{i=1}^N$ and $\{x_{t+k}^i\}_{i=1}^N$, respectively. The reference scene is processed by both the SPARK encoder f_θ and the frozen encoder f_{θ_0}, yielding adapted token representations $\{u_t^i\}_{i=1}^N$ and frozen token representations $\{v_t^i\}_{i=1}^N$. In addition, the SPARK encoder produces a class token $u_t^{[\text{CLS}]}$, which serves as the compact feature associated with the reference scene. For the target scene, the frozen encoder processes all target patches to produce $\{v_{t+k}^i\}_{i=1}^N$, while the SPARK encoder processes only a subset of visible target patches indexed by $M \subset \{1,\dots,N\}$, where $|M| = \lfloor rN \rfloor$ for a visible ratio $r \in (0,1)$. This yields adapted target tokens $\{u_{t+k}^i\}_{i\in M}$, which are used as hints for target-scene prediction.

Dynamics-aware abstraction loss. The abstraction objective encourages the compact feature $u_t^{[\text{CLS}]}$ to retain the information required to predict the target scene. To this end, we reconstruct target token representations from the compact feature together with the visible target hints $\{u_{t+k}^i\}_{i\in M}$. Let M^c denote the masked target-patch indices. We introduce learned mask tokens $\{m_i\}_{i\in M^c}$ and feed the concatenation of the compact feature, visible tar-

get tokens, and mask tokens into a decoder d_ϕ. The decoder predicts target token representations $\{\hat{u}^i_{t+k}\}_{i\in M^c}$ for the masked regions. We then align these predicted target tokens with the corresponding frozen target representations produced by the original foundation model:

$$\mathcal{L}_{\mathrm{DA}} = \sum_{i\in M^c} d\big(\hat{u}^i_{t+k},\ v^i_{t+k}\big)\,, \tag{1}$$

where $d(\cdot,\cdot)$ is a patch-level distance function. By requiring the compact feature to support prediction of the target scene, this loss encourages the encoder to abstract temporal and action-relevant information from the reference observation into a compact representation.

Knowledge preservation loss. While the abstraction loss specializes the representation toward temporal understanding, it may also distort the useful semantic structure inherited from the pretrained model. To prevent this drift, we introduce a retention loss that aligns the patch-wise representations of the adapted encoder with those of the frozen original encoder on the reference scene:

$$\mathcal{L}_{\mathrm{R}} = \sum_{i=1}^{N} d'\big(u^i_t,\ v^i_t\big)\,, \tag{2}$$

where $d'(\cdot,\cdot)$ is a patch-level distance function. Since f_{θ_0} is frozen, the resulting representations provide a stable semantic target during post-training. This regularization preserves general visual knowledge while allowing the adapted encoder to specialize in robot control.

Post-training objective. The overall SPARK objective combines dynamics-aware abstraction with knowledge preservation:

$$\mathcal{L}_{\mathrm{SPARK}} = \mathcal{L}_{\mathrm{DA}} + \alpha\mathcal{L}_{\mathrm{R}}, \tag{3}$$

where $\alpha > 0$ balances adaptation and preservation. Optimizing this objective yields compact visual features that capture temporal and action-relevant information while remaining grounded in the semantic knowledge of the original foundation model.

Policy learning. For downstream control, we train a policy by behavior cloning on trajectories of image observations and actions. Let f_θ be the adapted SPARK encoder and let π_ϕ be a policy network. Given a trajectory $\{(I_t, o_t, a_t)\}_{t=1}^{T}$ with image observation I_t, optional proprioceptive feature o_t, and action a_t, we extract a compact visual feature $z_t = f_\theta(I_t)^{[\mathrm{CLS}]}$ and concatenate it with proprioception to form the policy input $s_t = [z_t; o_t]$. The policy predicts $\hat{a}_t = \pi_\phi(s_t)$ and is trained with a standard behavior cloning objective:

$$\mathcal{L}_{\mathrm{BC}}(\phi) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{t=1}^{T_i}\ell\Big(\pi_\phi(s_t^{(i)}),\ a_t^{(i)}\Big)\,, \tag{4}$$

where ℓ denotes a regression loss such as mean squared error. Unless stated otherwise, the encoder f_θ is kept fixed after post-training, and only the policy network π_ϕ is optimized.

(a) Franka Kitchen (b) DeepMind Control (c) Metaworld

Figure 2. **Environments for evaluation.** We visualize three vision-based robot learning benchmarks employed for the validation of our method: (a) Franka Kitchen [8], (b) DeepMind Control Suite [25], and (c) Metaworld [14].

4. Experiments

In this section, we evaluate the impacts of the proposed method in vision-based policy learning for robotic manipulation across various environments [8, 14, 18]. Moreover, we compare our method with commonly used post-training recipes.

4.1. Evaluation suites

We evaluate on three vision-based robot learning benchmarks encompassing Franka Kitchen, DeepMind Control Suite, and Metaworld, covering a total of 15 tasks. The environments and sampled tasks for each benchmark are shown in Fig. 2.

Franka Kitchen [8] is a challenging robotic manipulation environment, which consists of a 9 DoF position-controlled Franka robot interacting with a kitchen scene. The environment contains various kitchenwares including an openable microwave, four turnable oven burners, an oven light switch, two hinged cabinets, and a sliding cabinet door. We consider the imitation learning evaluation setup with five imitation tasks: *Knob1 on*, *Light on*, *Sdoor open*, *Ldoor open*, and *Micro open*.

DeepMind Control Suite [25] is a set of continuous control tasks with simulated robots. We use five imitation learning cases: *Walker-stand*, *Walker-walk*, *Reacher-easy*, *Cheetah-run*, and *Finger-spin*. We report the normalized

Table 1. **Experimental results on vision-based robot policy learning on Franka Kitchen.** We report the performance of imitation learning agents on Franka Kitchen [8], which are trained upon representations of CLIP [22], DINOv2 [20], SigLIP [28], and Theia [24] post-trained on Kinetics-400 [15] dataset. The success rates (%) are reported for all the tasks. We report the gains of our method over the baselines.

Method		Knob1 on	Light on	Sdoor open	Ldoor open	Micro open	Average
CLIP	Baseline	23.0±2.6	29.5±4.7	69.5±4.4	13.5±3.4	22.0±2.8	31.5±2.4
	SPARK	31.5±1.9	38.5±6.0	74.5±11.4	18.0±2.0	29.0±6.2	39.8±5.6
	Gain	+ 8.5	+ 9.0	+ 5.0	+ 4.5	+ 7.0	+ 8.3
SigLIP	Baseline	17.0±1.2	38.5±1.9	75.5±3.4	8.5±1.0	16.5±1.9	31.2±0.6
	SPARK	27.5±5.0	43.0±5.0	87.5±1.9	20.0±2.8	33.0±5.3	42.2±2.4
	Gain	+ 10.5	+ 4.5	+ 12.0	+ 11.5	+ 16.5	+ 10.0
DINOv2	Baseline	26.0±4.3	38.5±2.5	71.0±6.8	9.5±2.5	37.0±6.2	36.4 ±2.7
	SPARK	21.0±4.2	51.0±8.4	81.5±2.5	17.0±2.0	33.0±3.5	40.7±2.7
	Gain	- 5.0	+ 12.5	+ 10.5	+ 7.5	- 4.0	+ 4.3
Theia	Baseline	33.5±10.5	35.5±6.6	82.5±4.7	27.5±4.7	14.0±2.0	40.1±2.3
	SPARK	35.0±2.6	56.0±6.5	75.0±5.3	34.5±2.5	21.0±2.6	44.3±2.6
	Gain	+ 1.5	+ 20.5	- 7.5	+ 7.0	+ 7.0	+ 4.2

Table 2. **Experimental results on vision-based robot policy learning on DeepMind Control Suite (DMC).** We report the performance of imitation learning agents on the DMC benchmark [25], which are trained upon representations of CLIP [22], DINOv2 [20], and SigLIP [28] post-trained on Kinetics-400 [15] dataset. The normalized scores are reported for all the tasks. We report the gains of our method over the baseline.

Method		Walker-stand	Walker-work	Reacher-easy	Cheetah-run	Finger-spin	Average
CLIP	Baseline	58.1±5.8	24.9±2.8	75.2±4.0	12.7±2.0	68.4±3.1	47.9±2.1
	SPARK	62.0±3.5	29.5±4.5	75.6±4.7	22.0±2.1	63.7±1.6	50.6±0.9
	Gain	+ 3.9	+ 4.6	+ 0.4	+ 9.3	- 4.7	+ 2.7
SigLIP	Baseline	43.9±5.9	19.9±2.8	74.6±2.5	9.0±2.5	63.6±0.4	42.1±1.5
	SPARK	45.2±6.2	17.5±2.4	75.2±3.0	8.4±2.1	66.8±1.2	42.6±1.8
	Gain	+ 1.3	-2.4	+ 0.6	-0.6	+ 3.2	+ 0.5
DINOv2	Baseline	74.2±5.6	41.5±3.4	76.8±5.9	15.9±3.7	68.3±1.6	55.3 ±0.4
	SPARK	78.8±3.5	45.2±5.0	74.3±6.5	22.3±2.9	70.8±1.0	58.3±2.4
	Gain	+ 4.6	+ 3.7	- 2.5	+ 6.4	+ 2.5	+ 3.0

scores for all tasks.

MetaWorld [27] is a suite of simulated robotic manipulation tasks with a Sawyer robot arm. We focus on a subset of five representative tasks: *Assembly*, *Bin-picking*, *Button-press-topdown*, *Drawer-open*, and *Hammer*. We measure the best success rates among the online evaluation trials.

4.2. Implementation details

Initialization and post-training setup. For post-training, we initialize the SPARK encoder from public foundation models, including CLIP [22], DINOv2 [20], SigLIP [28], and Theia [24]. Unless otherwise stated, post-training is conducted on Kinetics-400 [15] for 50 epochs. Following repeated sampling [6, 12], this corresponds to an effective training length of 200 epochs. We use AdamW [16] with a batch size of 1536 and train on dynamic scenes at a resolution of 224 × 224. Frames are sampled at 30 FPS, and the temporal offset between the reference and target scenes is uniformly sampled between 4 and 96 frames. We apply random resized cropping and horizontal flipping, using the same crop for both reference and target scenes. The target scene is masked with a ratio of 0.9. The decoder consists of eight Vision Transformer blocks. All remaining hyperparameters follow the default pretraining settings of the corresponding backbone.

Policy learning setup. For downstream control, each agent consists of a frozen visual encoder and a policy network trained with behavior cloning. The policy takes as input the compact visual feature extracted from the encoder, together with proprioceptive input when available. For

Table 3. **Experimental results on vision-based robot policy learning on Metaworld.** We report the performance of imitation learning agents on Metaworld [27], which are trained upon representations of CLIP [22], DINOv2 [20], and SigLIP [28] post-trained on Kinetics-400 [15] dataset. The success rates (%) are reported for all the tasks. We report the gains of our method over the baseline.

Method		assembly	bin-picking	button-press	drawer-open	hammer	Average
CLIP	Baseline	55.2±6.6	63.2±7.7	58.4±13.4	100.0±0.0	87.2±15.6	72.8±1.1
	SPARK	72.0±12.0	63.2±7.0	58.8±14.5	100.0±0.0	94.4±6.1	77.7±7.0
	Gain	+ 16.8	+ 0.0	+ 0.4	0.0	+ 7.2	+ 4.9
SigLIP	Baseline	69.6±9.2	52.0±12.3	59.2±15.6	100.0±0.0	84.0±9.4	73.0±7.6
	SPARK	74.6±10.3	60.1±9.5	58.5±12.7	100.0±0.0	91.5±5.8	76.9±6.8
	Gain	+ 5.0	+ 8.1	-0.7	0.0	+ 7.5	+ 3.9
DINOv2	Baseline	64.8±5.9	70.4±14.9	64.0±19.4	100.0±0.0	92.8±5.9	78.4 ±7.7
	SPARK	80.8±5.2	67.2±7.2	68.8±16.3	100.0±0.0	99.2±1.8	83.2 ±5.5
	Gain	+ 16.0	- 3.2	+ 4.8	0.0	+ 6.4	+ 4.8

Table 4. **Experimental results on vision-based robot policy learning on Franka Kitchen.** We report the performance of imitation learning agents on Franka Kitchen [8], which are trained upon representations from the ViT-S/16 model pre-trained on Kinetics-400 [15] dataset. The success rates (%) are reported for all the tasks. We underline the second-best performance. We report the gains of our method over the second-best baseline.

Methods	Architecture	Franka Kitchen	DeepMind Control	Metaworld	Average
SimCLR	ViT-S	38.7	39.7	78.4	52.3
MoCo v3	ViT-S	25.4	43.7	65.4	44.8
DINO	ViT-S	38.7	50.9	82.4	57.3
MAE	ViT-S	26.1	43.7	65.4	45.1
SiamMAE	ViT-S	30.4	56.0	81.1	55.8
CLIP	ViT-B	31.5	47.9	72.8	50.7
SigLIP	ViT-B	31.2	42.1	73.0	48.8
DINOv2	ViT-B	36.4	55.3	78.4	56.7
SPARK (CLIP)	ViT-B	39.8	50.6	77.7	56.0
SPARK (SigLIP)	ViT-B	42.2	42.6	76.9	53.9
SPARK (DINOv2)	ViT-B	**40.7**	**58.3**	**83.2**	**60.7**

Franka Kitchen, the policy network is implemented as a two-layer MLP with batch normalization. For MetaWorld and DMC, we follow the policy-learning protocols of [18]. Unless otherwise stated, the visual encoder is frozen after post-training and only the policy network is optimized.

Benchmark-specific settings. For Franka Kitchen, we evaluate five imitation learning tasks following prior protocols [19, 21]. We use either the left or right camera at a resolution of 224×224 without depth input. Training runs for 20,000 steps, with online evaluation every 1,000 steps. We report the best success rate during training, averaged over four random seeds.

For MetaWorld and DMC, we follow the setup of [18]. Proprioceptive input is provided for all benchmarks except DMC. Each agent is trained for 100 epochs and evaluated online every 5 epochs. We report normalized scores for DMC and final success rates for MetaWorld, averaged across five random seeds.

4.3. Vision-based Robot Policy Learning in Real-world Environments

Quantitative results. To evaluate whether the benefits of SPARK extend beyond simulation, we additionally conduct experiments on real-world robot manipulation tasks. We consider four imitation learning tasks: *Cloth Folding*, *Cup Stacking*, *Cabinet Opening*, and *Drawer Closing*. For each task, we collect 20 demonstration episodes for behavior cloning and evaluate policies learned from visual representations extracted by the frozen backbone. The robot operates at 20 Hz with a policy's chunk size of 20. Following the same evaluation protocol as in simulation, we compare the vanilla CLIP encoder with its SPARK-adapted counterpart with 20 trials for each task. The results are reported in Table 5. SPARK improves the success rate on three out of four tasks, increasing performance from 65.0 to 75.0 on *Cloth Folding*, from 45.0 to 55.0 on *Cup Stacking*, and from 50.0 to 55.0 on *Cabinet Opening*. On *Drawer Closing*, the SPARK-adapted model achieves comparable performance

CVPR #23

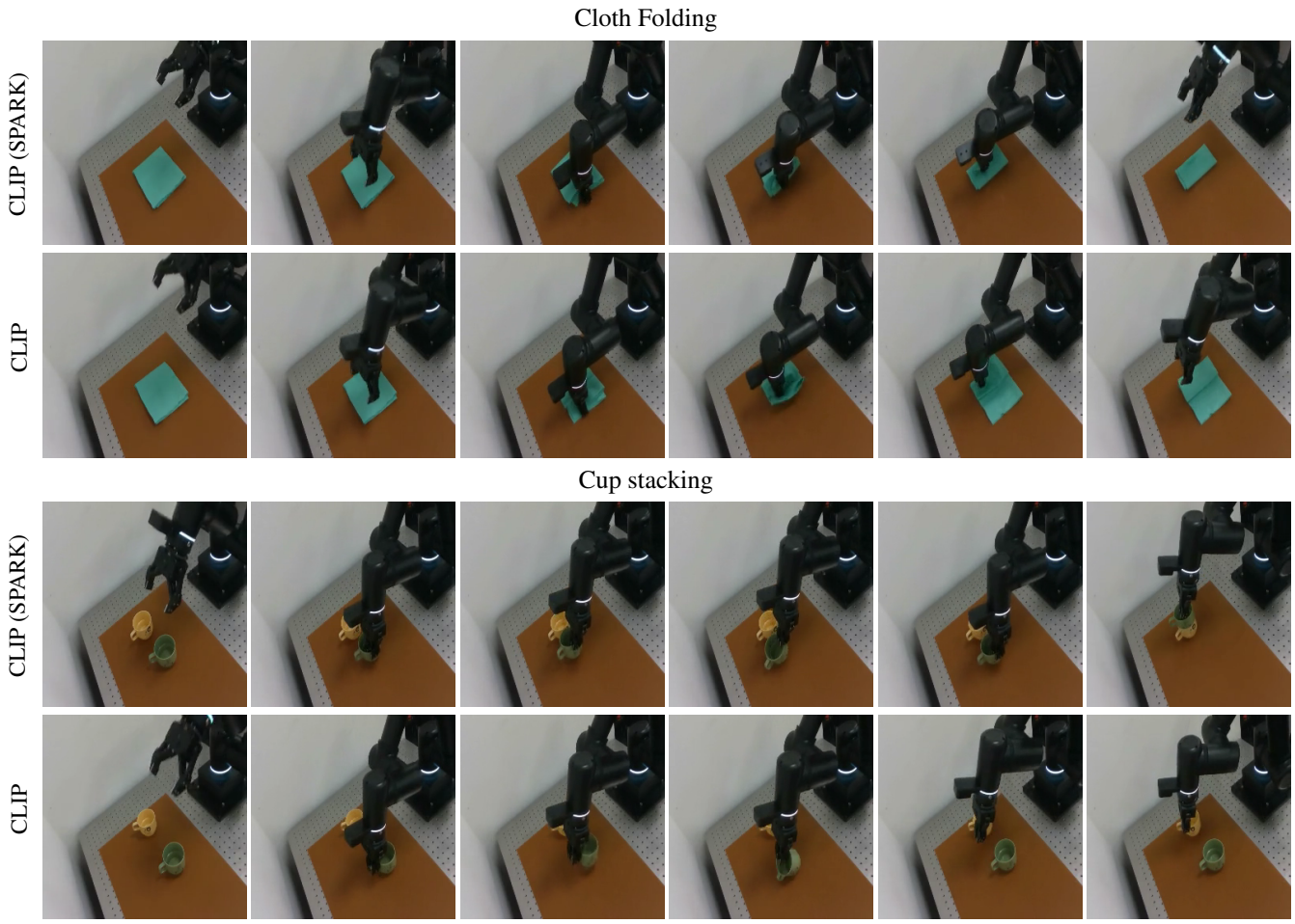

Figure 3. **Real-world demonstration trajectories.** Each row shows the temporal progression of a policy from the initial observation to intermediate interaction states and the final outcome. Across tasks, the SPARK-adapted policy exhibits stable approach, contact, and object manipulation behaviors, illustrating that the adapted visual representations capture control-relevant scene information that transfers reliably to real-world settings.

Table 5. **Performance on real-world vision-based robot policy learning.** Success rates (%) of imitation learning agents on three manipulation tasks: Cloth Folding, Cup Stacking, Cabinet Opening, and Drawer Closing. We compared CLIP [22] with our SPARK model, post-trained upon representations of CLIP on Kinetics-400 [15]. The results demonstrate the generalizability of SPARK in the real-world.

Method	Cloth Folding	Cup Stacking	Cabinet Opening	Drawer Closing
CLIP	65.0	45.0	50.0	55.0
SPARK (CLIP)	75.0	55.0	55.0	50.0

to the baseline. Overall, these results show that the gains obtained by SPARK are not limited to simulation, but transfer to real-world robot manipulation as well. This suggests that post-training foundation models with dynamics-aware abstraction and knowledge preservation improves the usefulness of pretrained visual representations in practical robotic settings.

Qualitative results. We further visualize representative real-world rollouts of the SPARK-adapted policy across multiple manipulation tasks in Fig. 3. Each row presents a temporal sequence from the initial scene, through intermediate interaction stages, to the final task outcome. The visualizations show that the policy maintains coherent manipulation progress throughout the rollout, including approach, contact, and task completion. Across different tasks and scene configurations, these examples suggest that SPARK produces visual representations that preserve the information most relevant for control, enabling robust real-world execution under variations in object pose, scene layout, and appearance.

CVPR #23

Table 6. **Ablation study varying post-training recipes** We compared our SPARK to other post-training recipes on Franka Kitchen [8] using CLIP [22] as an initial model. The success rates (%) are reported for all the tasks.

Post-training	Franka Kitchen
Baseline	31.5
Masked autoencoding	23.0
Correspondence matching	25.5
SPARK	39.8

4.4. Comparison to prior visual pretraining methods

We compare SPARK against conventional self-supervised pretraining methods for visual representations pretrained on dynamic scenes (e.g., consecutive frames), including SimCLR [2], MoCo v3 [4], DINO [1], MAE [11], and a dynamics-oriented variant, SiamMAE [9]. This evaluation verifies whether explicitly learning a compact state representation via conservative summarization is preferable to generic SSL objectives trained from scratch. As shown in Table 4, applying SPARK to a strong public encoder yields the best overall performance across suites, with *SPARK (DINOv2)* ranking first and *SPARK (CLIP)* substantially narrowing the gap to the strongest SSL baselines. Overall, adapting public encoders with SPARK is more effective for downstream robotics tasks than relying on generic SSL features, including methods designed for dynamic scenes.

5. Analysis

Comparison to various post-training recipes We conduct comparative analysis with various post-training recipes on Franka Kitchen in Table 6. We compared naive post-training using masked autoencoding (MAE) [11, 18] and correspondence matching among consecutive scenes [9]. All the post-trained models are initialized by CLIP [22]. As shown in the table, standard MAE style post-training and correspondence matching style post-training, which focus on generic reconstruction and temporal correspondence respectively, perform worse than the baseline with averages of 23.0% and 25.5%. In contrast, SPARK attains an average success rate of 39.8%, substantially outperforming both the vanilla CLIP and other post-trained models. This highlights that explicitly enforcing conservative summarization and knowledge preservation is more effective for adapting foundation models to robot control than generic reconstruction based objectives.

6. Conclusion

In this paper, we proposed SPARK, a simple post-training method for adapting pretrained foundation models to robot control. SPARK is based on two key principles: dynamics-aware abstraction and knowledge preservation. Dynamics-aware abstraction encourages the learned representation to capture temporal changes relevant to control, while knowledge preservation prevents the adapted model from losing useful semantic knowledge inherited from large-scale pretraining. In this way, SPARK makes pretrained visual representations more suitable for robot control without requiring training from scratch. Experiments on multiple robot learning benchmarks show that SPARK consistently improves over vanilla pretrained backbones. We further show that these gains extend beyond simulation to real-world robot manipulation tasks. These results suggest that cost-efficient post-training is a practical way to improve the usefulness of foundation models for embodied intelligence.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. 2, 8

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 2020. 8

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[4] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 2, 8

[5] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*. PMLR, 2023. 1, 2

[6] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3299–3309, 2021. 5

[7] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 2

[8] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Proceedings of the Conference on Robot Learning*, pages 1025–1037. PMLR, 2020. 4, 5, 6, 8

[9] Agrim Gupta, Jiajun Wu, Jia Deng, and Li Fei-Fei. Siamese masked autoencoders. In *Advances in Neural Information Processing Systems*, 2023. 8

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual repre-

sentation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2

[11] 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*, 2022. 2, 8

[12] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 5

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2

[14] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 2, 4

[15] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5, 6, 7

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

[17] 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. *arXiv preprint arXiv:2210.00030*, 2022. 1, 2

[18] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in neural information processing systems*, 2023. 2, 4, 6, 8

[19] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *6th Annual Conference on Robot Learning*, 2022. 1, 2, 6

[20] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 5, 6

[21] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *Proceedings of the 39th International Conference on Machine Learning*, pages 17359–17371. PMLR, 2022. 1, 2, 6

[22] 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 *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1, 2, 5, 6, 7, 8

[23] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Proceedings of The 6th Conference on Robot Learning*, pages 416–426. PMLR, 2023. 1, 2

[24] Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. 2024. 1, 5

[25] Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm_control: Software and tasks for continuous control. *arXiv preprint arXiv:2006.12983*, 2020. 4, 5

[26] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 1, 2

[27] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2020. 5, 6

[28] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, 2023. 1, 2, 5, 6

[29] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[30] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 2