VIPRA: VIDEO PREDICTION FOR ROBOT ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Can we turn a video prediction model into a robot policy? Videos, including those of humans or teleoperated robots, capture rich physical interactions. However, most of them lack labeled actions, which limits their use in robot learning. We present **Video Prediction for Robot Actions** (ViPRA), a simple pretraining-finetuning framework that learns continuous robot control from these actionless videos. Instead of directly predicting actions, we train a video-language model to predict *both future visual observations and motion-centric latent actions*, which serve as intermediate representations of scene dynamics. We train these latent actions using perceptual losses and optical flow consistency to ensure they reflect physically grounded behavior. For downstream control, we introduce a chunked *flow matching decoder* that maps latent actions to robot-specific continuous action sequences, using only 100 to 200 teleoperated demonstrations. This approach avoids expensive action annotation, supports generalization across embodiments, and enables smooth, high-frequency continuous control upto 22 Hz via chunked action decoding. Unlike prior latent action works that treat pretraining as autoregressive policy learning, ViPRA explicitly models both what changes and how. Our method outperforms strong baselines, with a 16% gain on the SIMPLER benchmark and a 13% improvement across real world manipulation tasks. We will release models and code at `https://vipra-robot.github.io`.

1 INTRODUCTION

Robots learn by doing, but collecting robot demonstrations, particularly at scale, is expensive, time-consuming, and limited by embodiment. In contrast, videos are abundant. From YouTube clips (Abu-El-Haija et al., 2016) of people performing tasks (Grauman et al., 2022b; 2024; Goyal et al., 2017; Damen et al., 2018) to logs of teleoperated robots (O'Neill et al., 2023), they capture rich physical interactions, diverse objects, and long-horizon behaviors that are difficult to script or reproduce. The challenge is that most of these videos may not include action labels.

At the same time, recent advances in video prediction models (Liu et al., 2024; Blattmann et al., 2023; Singer et al., 2022; Zhou et al., 2022; NVIDIA et al., 2025) open up a new opportunity: learning directly from large corpora of *actionless videos*. Beyond preserving high-level task semantics, these generative models exhibit a strong grasp of object dynamics and fine-grained physical interactions. This naturally leads to a central question: **Can a video prediction model be transformed into a control policy for physical robots?** In this work, we explore this question through a simple and scalable pretraining-finetuning framework that adapts a powerful video-language model (Liu et al., 2024) into a robot policy capable of learning from passive videos.

During pretraining, we co-train on two intuitive objectives: (i) predicting *what* happens next, in the form of *future visual observations*, and (ii) predicting *how* the scene evolves, using a compact intermediate representation known as *latent actions*[1]. By training with both objectives, the model learns to capture both semantic intent and physical dynamics. In contrast, prior latent action pretraining methods (Ye et al., 2024b; Bu et al., 2025; Chen et al., 2024; Bjorck et al., 2025) treat pretraining purely as policy learning in latent space, without leveraging video prediction or modeling state transitions, and often use temporally coarse task-centric latent actions. Our framework instead predicts state transitions through video prediction and outputs a sequence of fine-grained *motion-centric*

[1] Latent actions can be viewed as action-like tokens that summarize the transition between states without requiring access to ground-truth control commands

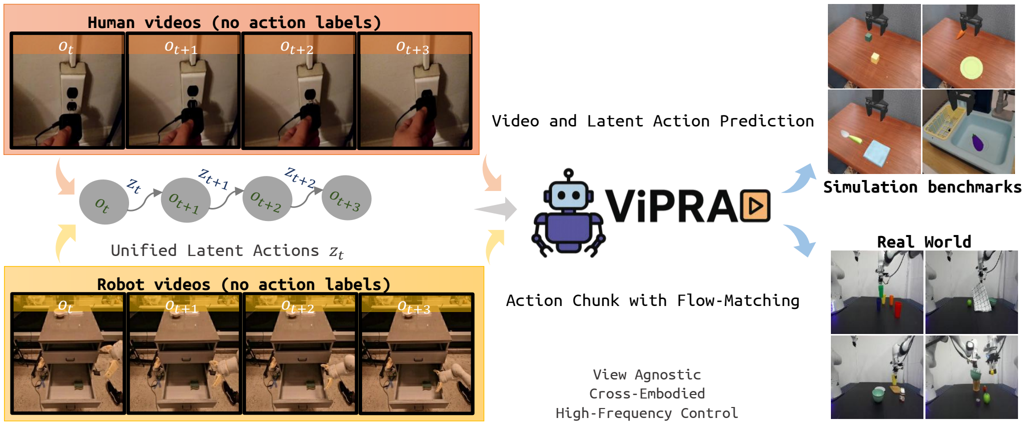

Human videos (no action labels)
o_t
o_{t+1}
o_{t+2}
o_{t+3}
z_t
z_{t+1}
z_{t+2}
Unified Latent Actions z_t
Robot videos (no action labels)
Video and Latent Action Prediction
ViPRA
Action Chunk with Flow-Matching
View Agnostic
Cross-Embodied
High-Frequency Control
Simulation benchmarks
Real World

Figure 1: We present ViPRA, which learns generalist robot policies from large-scale actionless videos by extracting motion-centric latent actions, which is used to pretrain a video-language model, and finally, finetuning a flow matching decoder with minimal labeled data for smooth, high-frequency control.

latent actions (3 to 6 Hz) over short horizons, capturing high-frequency dynamics critical for control. We further incorporate optical flow consistency as an additional supervision signal, promoting physically plausible and motion-aware latent representations.

Importantly, our pretraining leverages both unlabeled human and robot videos, which enables generalization across embodiments (see Figure 1, left). This broad exposure to passive visual data sets the foundation for effective finetuning with only a small number of teleoperated robot demonstrations. For finetuning on these demonstrations, we employ a *flow matching decoder* (Lipman et al., 2022) that maps latent actions to smooth, continuous robot action chunks (see Figure 1, right). Unlike prior vision-language-action models (VLAs) (Brohan et al., 2023; Kim et al., 2024; Li et al., 2024a; Black et al., 2024; Qu et al., 2025), which required thousands of hours of labeled action trajectories[2], our decoder aligns latent transitions with embodiment-specific motor behaviors. This design amortizes inference latency via *action chunking*, enabling smooth, high-frequency control by producing multiple low-level actions in a single forward pass. Our policy can support control rates approaching 22 Hz, to our knowledge matched only by one other 7B-parameter model (Kim et al., 2025).

In summary, our contributions are as follows.

(i) A scalable method to extract fine-grained motion-centric latent actions from unlabeled human and robot videos using perceptual and optical flow consistency losses.
(ii) A novel pretraining framework for robot control that jointly predicts future visual states and motion-centric latent actions within a unified video-language model.
(iii) A data-efficient pretraining–finetuning framework that integrates flow matching and action chunking to enable smooth, high-frequency continuous control, operating at up to 22 Hz.
(iv) Demonstrate empirical gains of 16% on the SIMPLER benchmark (Li et al., 2024c) and 13% on real world tasks over the strongest prior continuous control baselines.

2 Related Work

Vision-Language-Action Models Vision-Language-Action models (VLAs) (Brohan et al., 2023; Kim et al., 2024; Li et al., 2024a; Black et al., 2024; Qu et al., 2025) extend vision-language models (VLMs) (Touvron et al., 2023; Chen et al., 2023; Driess et al., 2023; Karamcheti et al., 2024; Beyer et al., 2024) by imitation learning on action-labeled robot demonstrations (O'Neill et al., 2023). Recent works explore auxiliary objectives including visual trace prediction (Niu et al., 2024), chain-of-thought reasoning (Wei et al., 2022), and conversational instruction tuning (Li et al., 2024b). However, all existing VLAs require extensive labeled action data, creating a fundamental scalability bottleneck due to the prohibitive cost of data collection. Furthermore, these models focus primarily on grounding language in visual semantics while lacking explicit mechanisms for modeling physical dynamics or temporal structure in action generation. In contrast, ViPRA eliminates the labeled data requirement by leveraging unlabeled videos during pretraining and incorporates temporal dynamics

[2] 10000 hours of pretraining OpenX data (O'Neill et al., 2023) and 5-100 hours of fine-tuning demonstrations

through joint prediction of future visual states and multi-step latent actions, which provides robust priors for high-frequency control.

Robot Learning from Videos Videos offer a scalable source of information about object dynamics, task structure, and human behavior. Visual planning methods (Du et al., 2023a; Wu et al., 2023a; Ko et al., 2023; Du et al., 2024; Baker et al., 2022; Liang et al., 2024; Luo et al., 2025a) use generative video models to plan in video or video-language space and rely on an inverse dynamics model to convert predicted frames into actions. While effective for long-horizon reasoning, these methods often incur high inference costs, limiting their suitability for high-frequency, dexterous control. Different from the above, policy supervision approaches (Luo et al., 2025a;b) use video models as supervision or reward sources to train policies.

Recent work explores joint training for video generation and action prediction (Li et al., 2025; Guo et al., 2024), with (Li et al., 2025) also introducing decoupled action decoding to mitigate inference overhead, but evaluations are mostly on smaller-scale datasets, simulation, and do not demonstrate scaling to internet-scale passive videos.

Other efforts leverage human videos to pretrain visual representations for downstream visuomotor control (Nair et al., 2022; Dasari et al., 2023; Xiao et al., 2022; Karamcheti et al., 2023), or extract intermediate cues such as affordances (Bahl et al., 2023; Kannan et al., 2023; Bharadhwaj et al., 2023; Liu et al., 2022; Goyal et al., 2022), interactions (Zeng et al., 2024), or visual traces (Wen et al., 2023; Bharadhwaj et al., 2024; Mandikal & Grauman, 2022; Bahl et al., 2022) from unlabeled videos to guide policy learning. These approaches depend on structured priors or explicit cue extraction, which can constrain scalability. In contrast, we learn motion-centric latent actions that capture temporal dynamics and pair them with video-language grounding, enabling scalable learning directly from large action-free video corpora.

Latent Action Spaces Latent action representations improve data efficiency by enabling learning from action-free videos via self-supervised learning (Dwibedi et al., 2018; Liang et al., 2025; Seo et al., 2022; Schmidt & Jiang, 2024; Cui et al., 2024). Recent methods impose discrete information bottlenecks with vector quantization encoders (van den Oord et al., 2017) and predict these tokens during policy learning (Ye et al., 2024a; Lee et al., 2024; Yang et al., 2024a; Bu et al., 2025; Chen et al., 2024; Bjorck et al., 2025), achieving strong real world results through imitation. Some train inverse dynamics models on limited labeled demonstrations before applying them to unlabeled video (Baker et al., 2022; Xu et al., 2023), while others treat latent actions as abstract embodiments and jointly train policies with inverse dynamics model predictions across embodiments (Jang et al., 2025). Another line of work uses these abstractions to build world simulators (Gao et al., 2025; Bruce et al., 2024) or plan in latent spaces (Ha & Schmidhuber, 2018; Weber et al., 2017; Hafner et al., 2019; 2018; Lee et al., 2019; Wu et al., 2022; Sekar et al., 2020) While these methods capture physical dynamics effectively, they struggle to generalize to novel settings due to limited semantic grounding. Video-language models can provide such multimodal grounding (Du et al., 2023a; Ko et al., 2023; Liang et al., 2024; Guo et al., 2024; Du et al., 2024), but existing approaches are typically computationally heavy and slow at inference. In contrast to existing methods, ViPRA learns fine-grained, motion-centric latent actions that capture temporal dynamics while leveraging a video-language model (Liu et al., 2024) for semantic grounding. We train a unified latent space from large-scale, action-free human and robot videos, enabling cross-embodiment transfer. By predicting action chunks during both latent pretraining and real-action finetuning, we amortize inference latency and achieve smooth, high-frequency control.

3 Background

We defer discussion on VQ-VAE for discrete latent actions, optical flow estimation, behavior cloning, and flow matching for continuous control to Appendix A.

4 Video Prediction for Robot Actions

A generalist robotic agent must combine precise low-level control with environment-agnostic high-level intelligence. Video generation models are well-suited to this goal, as future-state prediction captures both physical interaction detail and task-related semantic context. Achieving this requires

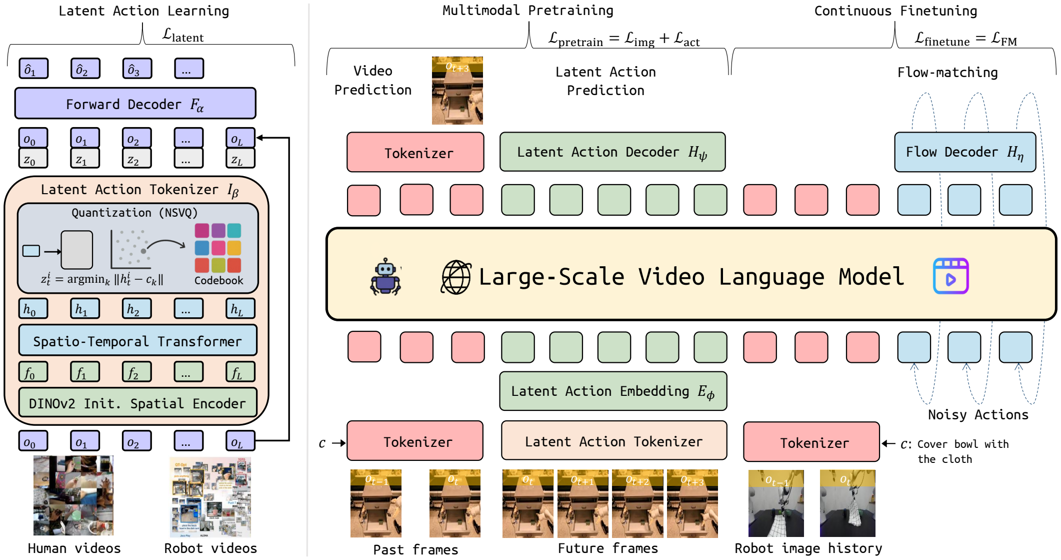


Figure 2: **ViPRA framework** comprises of: (1) **Latent Action Learning** (left): A neural quantization bottleneck extracts discrete latent actions z_t from image sequences $o_{0:L}$ in both human and robot videos, trained via reconstruction loss $\mathcal{L}_{\text{latent}}$ to capture motion-centric dynamics. (2) **Multimodal Pretraining** (center): A video-language model jointly predicts future observations o_{t+H} and latent action sequences $z_{t:t+H-1}$ from past frames (o_{t-1}, o_t) and task description c, using loss $\mathcal{L}_{\text{pretrain}}$. (3) **Continuous Finetuning** (right): A flow matching decoder maps latent actions to continuous robot actions $a_{t:t+H-1}$ using noisy action conditioning and loss $\mathcal{L}_{\text{FM}}$, enabling smooth, high-frequency control.

effectively utilizing large-scale data, architectures that expose motion-centric signals, and stable training pipelines. To this end, we present ViPRA: *(i)* learning motion-centric discrete latent actions from large-scale human and robot videos without action supervision, guided by perceptual and optical flow consistency; *(ii)* pretraining a multimodal video-language model to jointly predict future visual observations and latent action sequences, grounding temporal dynamics in semantics; and *(iii)* finetuning a flow-matching decoder that maps latent actions to smooth, continuous control using only a few hundred demonstrations. This hierarchy leverages the rich physical priors of video models while ensuring the precision needed for real world robot control.

4.1 Latent Action Learning from Actionless Videos

We first train a latent action model to represent behavior in both human and robot videos without requiring action supervision. As illustrated in Fig. 2 (left), this phase extracts motion-centric latent actions z_t that captures how the environment changes (inverse dynamics) and reciprocally also models the visual observations in response to these implicit actions (forward prediction).

Given a length-$(L+1)$ observation sequence $o_{0:L} = [o_0, o_1, \ldots, o_L]$ sampled from human or robot videos, our objective is to learn discrete latent action tokens $z_t = [z_t^1, z_t^2, \ldots, z_t^{N_{\text{latent}}}]$ that encode the motion dynamics at each timestep t. Here, N_{latent} denotes the number of latent action components, and each component z_t^i is quantized from a shared codebook $\mathcal{C}$ of size $|\mathcal{C}| = 8$.

We train an inverse dynamics encoder $I_\beta(z_t \mid o_{0:L})$ that predicts latent action token z_t by conditioning on the full observation sequence $o_{0:L}$. This non-causal design allows the latent action z_t to incorporate both past and future context, making it sensitive to local motion intent–for instance, distinguishing a pickup from a putdown based on surrounding frames. By providing the encoder with the entire clip, we reduce reconstruction ambiguity and force z_t to encode the minimal but sufficient information to explain the local transition.

We jointly train a forward decoder $F_\alpha(\hat{o}_{t+1} \mid o_{0:t}, z_{0:t})$ that predicts the future frame $\hat{o}_{t+1}$ given the history of observations $o_{0:t}$ and latent actions $z_{0:t}$. This reconstruction task ensures that the learned latent actions z_t contain sufficient information to explain scene dynamics. The model is optimized using three complementary loss components: pixel-level L_1 reconstruction loss $\mathcal{L}_{\text{rec}}$ for accurate frame prediction, perceptual loss $\mathcal{L}_{\text{LPIPS}}$ (Zhang et al., 2018) for semantic consistency, and optical flow consistency loss $\mathcal{L}_{\text{flow}}$ to encourage physically plausible motion patterns:

$$\mathcal{L}_{\text{flow}} = \frac{1}{L}\sum_{t=2}^{L+1} \|\text{OF}(\hat{o}_t, \hat{o}_{t-1}) - \text{OF}(o_t, o_{t-1})\|_1 + \frac{1}{L}\sum_{t=1}^{L} \|\text{OF}(\hat{o}_t, \hat{o}_{t+1}) - \text{OF}(o_t, o_{t+1})\|_1 \quad (1)$$

where OF(a, b) denotes optical flow between frames a and b computed via RAFT (Teed & Deng, 2020). This loss encourages predicted frames to exhibit motion patterns consistent with ground truth, supporting temporally coherent dynamics. The total latent action learning loss combines all components:

$$\mathcal{L}_{\text{latent}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \mathbb{I}(\text{step} > \alpha_{\text{flow}})\lambda_{\text{flow}}\mathcal{L}_{\text{flow}}, \quad (2)$$

where the flow loss is activated only after α_{flow} warm-up steps to avoid instability from poor early reconstructions. Thus, latent actions serve as a representation of scene dynamics effectively bridging between visual observations and any embodiment-specific control commands. We provide further architecture details in Appendix B.

4.2 Leveraging Multimodal Video Models for Action Pretraining

After obtaining discrete latent actions, we design a pretraining scheme leveraging a powerful multimodal video prediction model. Such models are trained on large-scale datasets to jointly reconstruct video tokens and predict aligned language captions, thereby encoding rich semantic cues and dynamic priors about how the world changes in their latent space. By aligning our discrete latent actions z_t with the outputs of the generative model, we can effectively pretrain a *high-level controller* that can learn from video clips.

To this end, we *jointly* predict future visual tokens and latent actions, unifying dynamic scene understanding and abstract control representation in a temporally coherent latent space. Given the most recent observations (o_{t-1}, o_t) and a task description c, the model predicts a future frame o_{t+H} that is H steps ahead, along with a latent action sequence $z_{t:t+H-1} = [z_t, z_{t+1}, \ldots, z_{t+H-1}]$ representing the transitions leading to o_{t+H}. This multi-step horizon encourages meaningful and distinct scene changes, providing robust conditioning for downstream action inference.

As shown in Fig. 2 (center), we build on the instruction-tuned LWM-Chat-1M (Liu et al., 2024) as our base policy G_θ and extend it with two modules for latent action modeling: (i) a **Latent Action Embedding** head E_ϕ that maps each discrete latent token $z_t^i \in \mathcal{C}$ to a d_z-dimensional vector $\tilde{z}_t^i = E_\phi(z_t^i)$ in the model's token space, and (ii) a **Latent Action Token Decoder** H_ψ, a multi-layer perception (MLP) that autoregressively predicts the next latent token $\hat{z}_t^{i+1} = H_\psi\left(G_\theta\left(c, o_{t-1}, o_t, o_{t+H}, \tilde{z}_{<t}, \tilde{z}_t^{\leq i}\right)\right)$ from the transformer hidden state till position i. This allows the model to generate latent action sequences in the same autoregressive manner as language or video tokens, leveraging the multimodal token space learned during pretraining.

During training, we apply *teacher forcing* to both visual and action predictions. We use LWM's VQ-VAE encoder E_{VQ} to encode ground-truth future frame o_{t+H} into N_{tokens} discrete tokens $x_{t+H} = E_{\text{VQ}}(o_{t+H})$, which serve as supervision targets for the visual prediction $\hat{x}_{t+H} = G_\theta(c, o_{t-1}, o_t)$. The pretraining objective $\mathcal{L}_{\text{pretrain}}$ combines the visual and action components as:

$$\mathcal{L}_{\text{pretrain}} = \underbrace{\sum_{i=1}^{N_{\text{tokens}}} \text{CE}(\hat{x}_{t+H}^i, x_{t+H}^i)}_{\mathcal{L}_{\text{img}}} + \underbrace{\sum_{k=t}^{t+H-1} \sum_{i=1}^{N_{\text{latent}}} \text{CE}(\hat{z}_k^i, z_k^i)}_{\mathcal{L}_{\text{act}}}, \quad (3)$$

where CE(a, b) denotes the standard cross-entropy loss between logits for a and label b.

4.3 Continuous Adaptation

While the latent action pretrained video model provides robust semantic grounding, it lacks the physical precision needed for smooth, low-level robot control. To address this gap, we augment the pretrained model to output continuous actions, utilizing a flow matching decoder trained on real robot trajectories. This adaptation enables temporally smooth, physically consistent motor commands conditioned on visual and linguistic contexts.

As shown in Fig. 2 (right), we augment the video model G_θ with two action-specific components: (i) an **Action Encoder** E_γ, and (ii) a **Flow Decoder** H_η. The encoder E_γ embeds continuous noisy actions $x_s \in \mathbb{R}^{H\times D}$ into the token space, while the decoder H_η predicts a flow field over the action chunk. Following the flow matching framework from Eq. 7, we sample a target action sequence $a_{t:t+H-1} \in \mathbb{R}^{H\times D}$, draw a noise sample $x_0 \sim \mathcal{N}(0, I)$, and interpolate:

$$x_s = s \cdot x_0 + (1-s) \cdot a_{t:t+H-1}, \quad s \sim \text{Beta}(a, b).$$

We use $a = 1.5$ and $b = 1$ for sampling from Beta distribution. This noisy input is encoded via $f_s = E_\gamma(x_s, s)$, and passed into the transformer along with VQ-encoded image tokens of (o_{t-1}, o_t) and language prompt c. The model predicts a flow field $\hat{g} = H_\eta\left(G_\theta\left(c, o_{t-1}, o_t, f_s\right)\right)$, which is supervised using the flow matching objective from Eq. 9:

$$\mathcal{L}_{\text{FM}} = \left\|a_{t:t+H-1} - x_0 - (1-s)\cdot \hat{g}\right\|_2^2.$$

At inference time, given the visual history (o_{t-1}, o_t) and task instruction c, we iteratively solve for the continuous action chunk $a_{t:t+H-1}$ using forward Euler integration (Eq. 10) of the predicted flow field from $s = 0$ to $s = 1$, over 10 uniform steps with $\Delta s = 0.1$. This continuous control refinement layer injects dynamics consistency and smoothness unavailable to the discrete latent tokens alone.

5 Experiments

To evaluate ViPRA, we conduct extensive experiments in both simulation and the real world to address the following research questions: (i) Can a generalist policy be trained to leverage both the physical dynamics and semantic understanding of video models? (ii) Does ViPRA efficiently exploit large-scale, actionless video data? (iii) Can multimodal pretraining yield strong high-level priors for downstream policy? (iv) How well does ViPRA adapt to high-frequency continuous control settings? (v) Does ViPRA outperform methods that do not exploit video foundation models?

5.1 Environments & Training

Training Dataset For learning latent actions and pretraining the video-language model, we use 198k human videos from Something-Something v2 (Goyal et al., 2017) and a subset of actionless robot videos from the OpenX (O'Neill et al., 2023) dataset: 87k Fractal (Brohan et al., 2022) videos, 25.4k BridgeV2 (Ebert et al., 2021), and 85.6k Kuka (Kalashnikov et al., 2018) videos. We describe training details and hyperparameters for latent action learning in Appendix B, pretraining in Appendix C, and flow matching finetuning in Appendix D.

Simulation Benchmarks Following prior latent action works (Ye et al., 2024b; Bu et al., 2025), we benchmark ViPRA in SIMPLER (Li et al., 2024c), an open-source suite for evaluating generalist manipulation policies. We evaluate on four Bridge task with a 7-DoF WidowX arm, a benchmark designed to test generalization across diverse manipulation goals. Since SIMPLER lacks finetuning data, we collect 100 diverse multi-task trajectories using a pretrained VLA model (Ye et al., 2024b). We provide details about our SIMPLER tasks and LIBERO Long benchmarks in Appendix E.

Real World Manipulation While SIMPLER already provides a strong correlation between simulated and real world policy performance, we further strengthen our findings with rigorous evaluations on physical robots. We evaluate ViPRA on a bimanual setup with two 7-DoF Franka Panda robots. For single-arm experiments, we finetune on three multi-instruction tasks: (1) `pick up cloth and cover` $\langle$`object`$\rangle$, (2) `pick up` $\langle$`object`$_1\rangle$ `and place on` $\langle$`object`$_2\rangle$, and (3) `pick up` $\langle$`color`$_1\rangle$ `cup and stack on` $\langle$`color`$_2\rangle$ `cup`. We use GELLO (Wu et al., 2023b) teleoperation to collect 180 trajectories, per task spanning 5 cup colors and 10 object types. For both simulation and real world settings, we report full success and partial success; partial success is defined as grasping the correct object, and full success requires completing the task (e.g., placing, stacking, covering). We evaluate with both seen and unseen objects, textures, colors, and shapes to test generalization. For real world evaluation, policies run using only a front-facing camera. We predict action chunks of length $H{=}14$ and replan after executing the first 7 steps. For this evaluation, we cap our policies at an effective closed-loop control rate of 3.5 Hz, though they can also operate at higher frequencies upto 22 Hz.

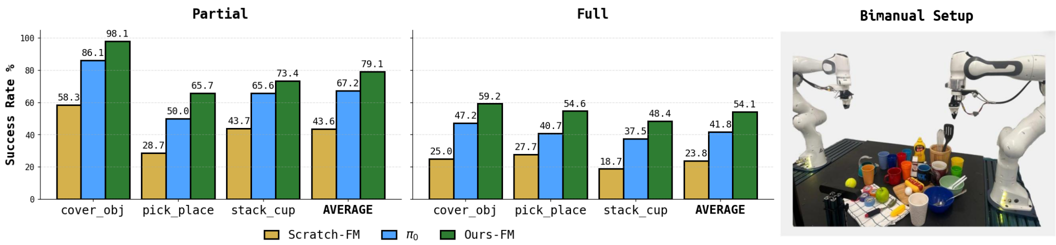


Figure 3: **Real World Evaluations** (Left) We report full and partial success rates on three manipulation tasks. ViPRA-FM significantly outperforms baselines. (Right) We show our physical robot setup and task objects.

	Discrete Actions					Continuous Actions				
Task	**Scratch-AR**	**VPT**	**OpenVLA**	**LAPA**	**ViPRA-AR**	**Scratch-FM**	**UniPI**	**π_0**	**UniVLA**	**ViPRA-FM**
					Success Rates					
StackG2Y	54.2	45.8	25.0	33.3	**66.7**	16.7	2.7	0.0	-	**54.2**
Carrot2Plate	58.3	37.5	20.8	41.7	**62.5**	33.3	2.7	20.8	-	**50.0**
Spoon2Cloth	37.5	**70.8**	50.0	66.7	66.7	50.0	0.0	4.17	-	**66.7**
Eggplant2Bask	58.3	50.0	58.3	70.8	**83.3**	66.7	0.0	**83.3**	-	79.2
AVG	52.1	51.0	38.6	53.1	**69.8**	41.7	1.7	27.1	42.7	**62.5**
					Grasp Rates					
StackG2Y	62.5	62.5	**70.8**	66.7	66.7	45.8	20.8	12.5	-	**62.5**
Carrot2Plate	54.2	54.2	37.5	**62.5**	**62.5**	45.8	33.2	25.0	-	**54.2**
Spoon2Cloth	75.0	79.2	75.0	**87.5**	75.0	62.5	22.2	16.7	-	**79.2**
Eggplant2Bask	66.7	70.8	91.7	79.2	**100**	87.5	16.0	**91.7**	-	**91.7**
AVG	65.6	66.7	68.8	73.9	**76.1**	60.4	23.1	36.5	50.0	**71.9**

Table 1: We report success rates and grasp rates on four bridge tasks in SIMPLER benchmark suite.

5.2 Baselines

We evaluate ViPRA against strong baselines across discrete and continuous action formulations.

Scratch. As a reference, Scratch finetunes the video-language backbone (LWM) (Liu et al., 2024) directly on downstream tasks with image history and action chunking, without any pretraining. It establishes baseline performance when no latent action or video-based pretraining is used.

VLA baselines. We include OpenVLA (Kim et al., 2024) and π_0 Black et al. (2024). OpenVLA discretizes actions and adds a one-step autoregressive (AR) action predictor on top of a Prismatic-7B (Karamcheti et al., 2024), while π_0 augments a PaliGemma-3B (Beyer et al., 2024) with a chunked flow matching (FM) decoder. Both use action-labeled robot demos from OpenX (O'Neill et al., 2023) containing 970k trajectories, while π_0 also uses proprietary robot data.

Latent action baselines. We include LAPA (Ye et al., 2024b) and UniVLA (Bu et al., 2025), both of which learn one-step temporally coarse latent actions without video prediction during pretraining. UniVLA improves upon LAPA by learning language-conditioned task-centric actions in DINOv2 space. UniVLA uses a Prismatic-7B backbone with a L1 action decoder, whereas LAPA uses an LWM backbone with one-step AR prediction. Both rely on OpenX demos, with UniVLA additionally leveraging Ego4D (Grauman et al., 2022a) and GNM (Yang et al., 2024b).

Video learning baselines. We include UniPI (Du et al., 2024) and VPT (Baker et al., 2022), both of which leverage videos for pretraining. UniPI trains a video diffusion model and trains an IDM to recovers actions, while VPT trains an IDM on labeled data to extract pseudo-actions that are then used to pretrain an LWM backbone. Reported results are from (Ye et al., 2024b), which evaluated them on SIMPLER in a comparable setting.

We include both ViPRA-AR, aligned with discrete autoregressive baselines, and ViPRA-FM, aligned with continuous flow-matching methods.

5.3 Simulation Results

As shown in Table 1, ViPRA achieves the best average success rate in both discrete and continuous settings. In the **discrete setting**, ViPRA-AR surpasses LAPA and OpenVLA by a large margin (69.8% vs. 53.1% and 38.6%), excelling in precision-heavy tasks such as `StackG2Y`. In the **continuous setting**, ViPRA-FM outperforms Scratch-FM by 20.8%, π_0 by 35.4%, and UniVLA by 19.8%, showing the benefits of motion-centric latents and multimodal video pretraining over training from scratch. Interestingly, due to the low noise and low ambiguity of the simulation setting, we find that

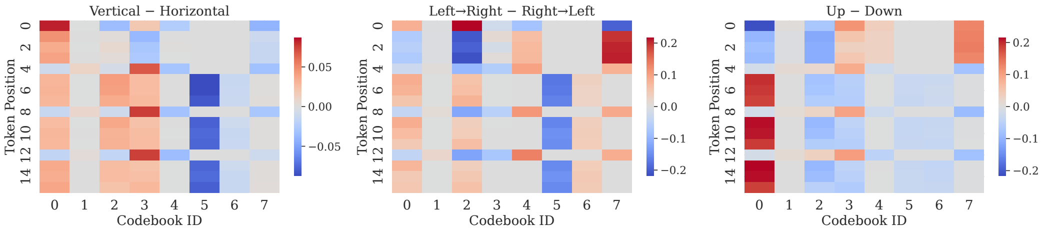


Figure 4: Positional codebook usage differences across action categories. Each heatmap shows the difference in per-position token usage between two groups: (left) vertical vs. horizontal, (middle) left → right vs. right → left, and (right) up vs. down. ViPRA learns positionally sensitive codes, with certain entries (e.g., 0, 2, 5) showing systematic variation, indicating that both token index and positions encode action dynamics.

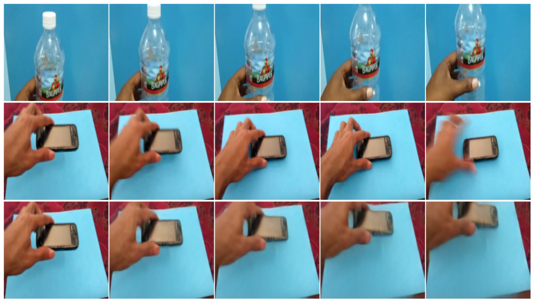


Figure 5: **Latent action transfer rollouts**: (Top) Video $o_{0:4}^{\text{UP}}$ with hand moving the bottle *up*. (Middle) Video $o_{0:4}^{\text{DOWN}}$ with hand putting the phone *down*. (Bottom) First, *up* latents are extracted using the inverse model $z_{0:3}^{\text{UP}} = I_\beta(o_{0:4}^{\text{UP}})$, then use the forward model and first *down* frame to rollout an *up* video $o_{0:4}^{\text{DOWN}\rightarrow\text{UP}} = F_\alpha(o_0^{\text{DOWN}}, z_{0:3}^{\text{UP}})$ where hand is picking the phone *up*.

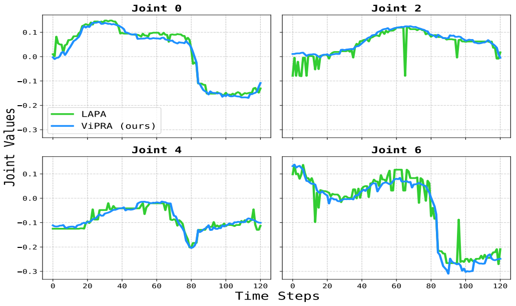


Figure 6: **Action smoothness**: ViPRA-FM (blue) produces smooth, continuous trajectories, while LAPA (green) exhibits local discontinuities and random spikes, often around contact events, despite tracking the overall trend. During real world deployment, such discontinuities triggered the emergency brake mechanism of the robot due to abrupt motor torque jumps.

ViPRA-AR outperforms the more expressive ViPRA-FM, which is slower to converge. However, ViPRA-FM achieves competitive performance, outperforming all other continuous/discrete baselines. ViPRA also surpasses other video learning approaches *i.e.*, UniPI and VPT. UniPI frequently generates action sequences that diverge from the given instruction in longer-horizon settings, while VPT provides only limited gains, indicating that IDM-derived pseudo-labels are sensitive to environment shifts. In contrast, ViPRA's joint use of latent action prediction and future state modeling yields stronger cross-environment transfer and more reliable task execution.

5.4 Real World Results

Figure 3 shows results on three real-world manipulation tasks. ViPRA-FM attains the best performance with a 54.1% average success rate, outperforming π_0 (40.1%) and Scratch-FM (23.8%). It also demonstrates robust retry behavior, repeatedly attempting grasps after failures, which leads to very high partial success rates–especially in `Cover-Obj`, where the cloth is reliably grasped even if not always placed correctly. While π_0 benefits from task-specific fine-tuning, ViPRA-FM achieves higher success with far less labeled data by leveraging dynamics priors from unlabeled videos. We exclude discrete action models from real-world evaluation, as their bin-based predictions exhibited unstable spikes under physical noise, often triggering emergency stops on the Franka arm. We provide additional analysis on generalization and robustness in Appendix G and challenging bimanual task evaluations in Appendix H.

5.5 Ablations & Analysis

Isolating effect of future state and latent prediction. Table 2 disentangles the contributions of future state prediction and latent action chunking. The LAPA baseline (latent-only) reaches 53.1%, while adding state prediction in ViPRA–*AC* boosts performance to 59.2%, showing that anticipating future observations improves control even with 1-step latents. Removing state prediction from our setup in ViPRA–*SP2* causes a drop from 69.8% to 59.4% (AR) and from 62.5% to 53.2% (FM), underscoring its importance for policy transfer. A state-only variant ViPRA–*LA* achieves 60.7%, comparable to ViPRA–*AC* but still below the full model, indicating that state prediction alone is not sufficient. Finally, adding state prediction at finetuning ViPRA+*SP3* degrades performance (53.1% AR, 31.3% FM), since the autoregressive structure couples action prediction with video prediction, causing compounding errors that drift irrecoverably on out-of-distribution states. ViPRA mitigates this by jointly predicting future visual tokens and latent action chunks during pretraining only.

Effect of action chunking. We apply chunking (Zhao et al., 2023) in both latent and real action spaces: during pretraining the model predicts latent action sequences, and during finetuning it outputs continuous chunks via flow matching. Removing chunking in ViPRA *–AC* reduces performance to 59.2% (AR) and 44.8% (FM), as single-step actions fail to capture smooth temporal dynamics. By combining chunking with future state prediction, the full model achieves the best results of 69.8% (AR) and 62.5% (FM), showing that the two objectives complement each other Finally, action chunking is not only critical in pretraining but also enables robust, high-frequency control at test time. With KV caching, ViPRA's flow matching decoder runs at 1.95 Hz per chunk, supporting effective control rates up to 22 Hz on hardware (chunk size 14). We provide additional discussion on the connection between action chunking and high-frequency execution in Appendix G.5.

Exp.	**Pretrain**	**Finetune**	**Succ.**
LAPA	1-step L	1-step A	53.1
ViPRA-AR	FS + H-step L	H-step A	**69.8**
–AC	FS + 1-step L	1-step A	59.2
–SP2	H-step L	H-step A	59.4
–LA	FS only	H-step A	60.7
+SP3	H-step L	FS + H-step A	53.1
ViPRA-FM	FS + H-step L	H-step A	**62.5**
–AC	FS + 1-step L	1-step A	44.8
–SP2	H-step L	H-step A	53.2
+SP3	H-step L	FS + H-step A	31.3

Table 2: Ablation of future state pred., latent pred., and action chunking. Shorthand: FS = future state pred., L = latent action, A = action pred., H=14 chunk size.

ViPRA enables smooth continuous control. To assess action smoothness, we compare ViPRA-FM (blue) with LAPA (Ye et al., 2024b) (green), a discrete policy, during closed-loop rollout. Both models are loaded into our inference pipeline and replayed over trajectories from the finetuning dataset, simulating deployment under real visual observations. As shown in Figure 6, both methods follow the intended trend, but LAPA exhibits sharp local spikes at contact events or occlusions, where small perceptual shifts trigger abrupt bin flips. In contrast, ViPRA-FM's flow matching head yields smooth, demonstration-aligned commands. Since such discontinuities are unsafe on hardware, we restrict real world comparisons to continuous baselines. We provide more analysis on discrete and continuous policies in Appendix F, showing how quantization, loss design, and control space affect action smoothness and deployment.

Latent action analysis. In Figure 5, a cross-video rollout test illustrates the correlation between latent actions and real action dynamics: injecting latents encoding upward motion from one video (top) into the opening frame of a downward moving video (middle) causes the reconstructions to move upward (bottom), demonstrating transferable, dynamics-aware semantics. Finally, Figure 4 analyzes codebook usage across categories by computing token-position × code-index histograms. The results indicate that both the choice of codebook entry and its position within the latent action sequence encode structured information about motion direction and dynamics.

6 Conclusion and Future Work

Video-language models provide a strong starting point for generalist robotic agents, as they capture both semantic intent and temporal dynamics crucial for real-world actions. Building on this, we introduced ViPRA, which learns motion-centric latent actions from large-scale actionless videos, pretrains a video-language model to jointly predict future states and latent actions, and refines these priors into smooth, high-frequency motor commands with a flow-matching decoder trained on only a few hundred demonstrations. Extensive evaluations in simulation and on real robots show that ViPRA outperforms methods relying solely on semantic pretraining, offering a scalable blueprint for general-purpose agents.

A key perspective is that the latent action decoder functions as a world model: given latent actions, it predicts future observations and can be conditioned on policy-sampled latents to generate multiple visual plans. This can enable alignment via reinforcement learning and test-time scaling through planning trees, where VLMs or heuristic functions act as reward models. Looking forward, extending ViPRA to dynamic, unstructured environments and integrating richer sensing modalities (e.g., wrist cameras, proprioception, tactile feedback, depth) remain exciting directions.

7 Reproducibility

All models and datasets used in our work are taken from open-sourced components. We take the publicly released LWM-Chat-1M (Liu et al., 2024) as our base video-language model and build on top of that. For latent action learning and pretraining video-language model, we use publicly available datasets: Something-Something (Goyal et al., 2017), Fractal (Brohan et al., 2022), BridgeV2 (Ebert et al., 2021), and Kuka (Kalashnikov et al., 2018). For SIMPLER (Li et al., 2024c) benchmarks, we collected finetuning trajectories by deploying a pretrained VLA model (Ye et al., 2024b). We will release this dataset for the community to reproduce our benchmarks. Moreover, we describe training details, hyperparameters, and model architecture for latent action learning in Appendix B, pretraining in Appendix C, and flow matching finetuning in Appendix D. We will release code, models, latent action labeled pretraining data and benchmark scripts at `https://vipra-robot.github.io`.

REFERENCES

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark, 2016. URL https://arxiv.org/abs/1609.08675.

Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022.

Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13778–13790, 2023.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Homanga Bharadhwaj, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation, 2024.

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

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

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

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.

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

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments, 2024. URL https://arxiv.org/abs/2402.15391.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions, 2025. URL https://arxiv.org/abs/2505.06111.

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.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint arXiv:2412.04445*, 2024.

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

Zichen Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *Advances in Neural Information Processing Systems*, 37:33933–33961, 2024.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 720–736, 2018.

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*, pp. 1183–1198. PMLR, 2023.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning, 2023a. URL `https://arxiv.org/abs/2310.10625`.

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.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. *arXiv preprint arXiv:2302.06692*, 2023b.

Debidatta Dwibedi, Jonathan Tompson, Corey Lynch, and Pierre Sermanet. Learning actionable representations from visual observations. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1577–1584. IEEE, 2018.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. In *International Conference on Machine Learning (ICML)*, 2025.

Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. Human hands as probes for interactive object understanding. In *CVPR*, 2022.

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

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

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022b.

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

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Haoran He, Yang Zhang, Liang Lin, Zhongwen Xu, and Ling Pan. Pre-trained video generative models as world simulators, 2025. URL `https://arxiv.org/abs/2502.07825`.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models, 2025. URL `https://arxiv.org/abs/2505.12705`.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pp. 651–673. PMLR, 2018.

Aditya Kannan, Kenneth Shaw, Shikhar Bahl, Pragna Mannam, and Deepak Pathak. Deft: Dexterous fine-tuning for real-world hand policies. *arXiv preprint arXiv:2310.19797*, 2023.

Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

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.

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.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

Wilhelm Kutta. *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen*. Teubner, 1901.

Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019.

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

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, et al. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024b.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024c.

Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

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.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL `https://arxiv.org/abs/2306.03310`.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with blockwise ringattention. *arXiv preprint arXiv:2402.08268*, 2024.

Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022.

Calvin Luo, Zilai Zeng, Yilun Du, and Chen Sun. Solving new tasks by adapting internet video knowledge. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Calvin Luo, Zilai Zeng, Mingxi Jia, Yilun Du, and Chen Sun. Self-adapting improvement loops for robotic learning, 2025b. URL `https://arxiv.org/abs/2506.06658`.

Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning*, pp. 651–661. PMLR, 2022.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai, 2025. URL https://arxiv.org/abs/2501.03575.

Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

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.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. URL https://arxiv.org/abs/2501.15830.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL https://arxiv.org/abs/2103.00020.

Carl Runge. Über die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.

Dominik Schmidt and Minqi Jiang. Learning to act without actions, 2024. URL https://arxiv.org/abs/2312.10812.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. *ICML*, 2020.

Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pp. 19561–19579. PMLR, 2022.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419. Springer, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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, 2025.

Mohammad Hassan Vali and Tom Bäckström. Nsvq: Noise substitution in vector quantization for machine learning. *IEEE Access*, 10:13598–13610, 2022. doi: 10.1109/ACCESS.2022.3147670.

Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, pp. 6309–6318, 2017.

Théophane Weber, Sébastien Racanière, David P Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adria Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*, 2017.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

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

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. *arXiv preprint arXiv:2312.13139*, 2023a.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning. *arXiv preprint arXiv:2206.14176*, 2022.

Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators, 2023b.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

Liang Xu, Ziyang Song, Dongliang Wang, Jing Su, Zhicheng Fang, Chenjing Ding, Weihao Gan, Yichao Yan, Xin Jin, Xiaokang Yang, et al. Actformer: A gan-based transformer towards general action-conditioned 3d human motion generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2228–2238, 2023.

Chenyu Yang, Davide Liconti, and Robert K Katzschmann. Vq-ace: Efficient policy search for dexterous robotic manipulation via action chunking embedding. *arXiv preprint arXiv:2411.03556*, 2024a.

Jonathan Yang, Catherine Glossop, Arjun Bhorkar, Dhruv Shah, Quan Vuong, Chelsea Finn, Dorsa Sadigh, and Sergey Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation, 2024b. URL `https://arxiv.org/abs/2402.19432`.

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*, 2024a.

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*, 2024b.

Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

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

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

As part of the supplementary material, we include additional details about the following.

A **Background**: Covers key paradigms relevant for ViPRA: VQ-VAE for learning discrete latent actions, optical flow estimation, behavior cloning for direct action prediction and flow matching for smooth continuous control.

B **Latent Action Learning**: Architecture design, loss formulations, and training protocols for discrete action codebook learning, including hyperparameter configurations and optimization strategies.

C **Multimodal Video-Action Integration**: Implementation details for extending LWM with latent actions, including embedding architecture, decoder design, and joint training methodology.

D **Continuous Control via Flow Matching**: Complete specification of noise scheduling, action encoder/decoder architectures, and end-to-end training procedure.

E **Simulation Benchmarks**: Detailed description of SIMPLER tasks and additional LIBERO Long benchmark.

F **Action Output Analysis**: Comparative visualization and discussion of predicted action trajectories across discrete and continuous policies, highlighting the impact of quantization, loss formulations, and control space choices on smoothness and deployment behavior.

G **Real World Experiments**: Detailed description of hardware setup, task design, policy generalization, retrying behavior, and the impact of action chunking on control frequency and real-time performance in physical deployments.

H **Bimanual Manipulation Tasks**: Evaluation of ViPRA-FM on two real world dual-arm tasks requiring spatial coordination and tool use, including task setup, challenges, quantitative results, and rollout visualizations from real robot executions.

Code, checkpoints, and latent action labeled data and rollout videos will be released at: `https://vipra-robot.github.io`.

A BACKGROUND

We review key paradigms relevant for ViPRA: VQ-VAE for imposing information bottleneck to learn discrete latent actions, optical flow estimation, behavior cloning for direct action prediction and flow matching for smooth continuous control.

Vector-Quantised VAEs (VQ-VAE). An encoder e_ζ maps an observation o_t to a cootinuou lat estimation,ent vector $\tilde{h}_t = [\tilde{h}_t^1, \ldots, \tilde{h}_t^N]$, which is quantized to a sequence of nearest codewords $z_t = [z_t^1, \ldots, z_t^N] \in \mathcal{C}$ using a shared discrete codebook. A decoder d_ζ reconstructs the observation $\hat{o}_t = d_\psi(z_t + \text{Err})$, where Err is a gradient estimator used to allow backpropagation through the non-differentiable quantization operation.

In traditional VQ-VAE (van den Oord et al., 2017), this estimator takes the form

$$\text{Err}_{\text{STE}} = \text{sg}(\tilde{h}_t - z_t), \tag{4}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator. The decoder input $z_t + \text{Err}_{\text{STE}}$ preserves the forward pass while enabling gradients to bypass the non-differentiable argmin. However, this approach typically requires auxiliary losses, such as the codebook and commitment losses, to stabilize training and encourage codebook usage.We adopt the NSVQ formulation (Vali & Bäckström, 2022), which replaces the deterministic STE with a stochastic noise-injected surrogate

$$\text{Err}_{\text{NSVQ}} = \|z_t - \tilde{h}_t\| \cdot \tilde{\epsilon} \tag{5}$$

where $\tilde{\epsilon} = \epsilon / \|\epsilon\|$ and $\epsilon \sim \mathcal{N}(0, I)$. The decoder thus receives $\hat{o}_t = d_\psi(z_t + \text{Err}_{\text{NSVQ}})$. Crucially, NSVQ enables gradients to flow to both the encoder and codebook using only the reconstruction loss, without requiring additional codebook loss terms. The noise-injected gradient estimator, combined with the *unused codebook replacement* technique applied during early training, significantly improves training stability and mitigates codebook collapse, a common issue in VQ-VAE training.

RAFT based Optical Flow Given two images, o_a and o_b, RAFT (Teed & Deng, 2020) obtains the dense displacement field $\mathbf{f}_{a \to b} \in \mathbb{R}^{H \times W \times 2}$ that maps each pixel in frame o_a to its location in o_b. It first extracts feature maps $\phi(o_a),\ \phi(o_b) \in \mathbb{R}^{H' \times W' \times d}$ (where ϕ is the feature extractor) and builds the all-pair correlation $\mathcal{R}$, with $\mathcal{R}_{ij,kl} = \langle \phi_{ij}(o_a), \phi_{kl}(o_b) \rangle$. The flow prediction is then refined iteratively: $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \Delta \mathbf{f}^{(k)}(\mathcal{R})$. When applied to temporally close frames in a video, this flow field f can give a good estimate of motion consistency.

Behavior Cloning (BC) is a supervised learning paradigm in robotics that learns policies directly from expert demonstrations. Given a dataset $\mathcal{D} = \{(o_t, a_t)\}_{t=1}^T$ of observation-action pairs from expert trajectories, BC trains a parameterized policy $\pi_{\text{BC}}(a_t | o_t; \theta)$ to minimize a distance metric between predicted and ground-truth actions:

$$\min_\theta \ \mathbb{E}_{(o_t, a_t) \sim \mathcal{D}} \left[d(\pi_{\text{BC}}(a_t | o_t; \theta), a_t) \right], \tag{6}$$

where $d(\cdot, \cdot)$ is typically the L1 or L2 distance for continuous actions or cross-entropy for discrete actions. This framework has been extended with high-capacity architectures: diffusion models (Chi et al., 2023; Zhao et al., 2023) parameterize $\pi_{\text{BC}}(a_t | o_t; \theta)$ as a denoising process that learns $p(a_t | o_t)$ through iterative refinement, while VLAs leverage pretrained language models (Touvron et al., 2023; Qwen et al., 2024) and visual encoders (Radford et al., 2021; Oquab et al., 2023; Tschannen et al., 2025) as the backbone architecture for θ, enabling multimodal grounding of actions in visual and linguistic contexts.

Flow Matching (Lipman et al., 2022) provides an alternative to diffusion models for learning continuous normalizing flows. While diffusion models learn the full denoising process, flow matching directly learns the vector field that transports samples from a source distribution to a target distribution. This approach offers computational advantages for robotics applications where real-time inference is critical.

Flow matching trains a neural network g_θ to predict the velocity field along a straight-line interpolation path. Given a source sample x_0 (typically Gaussian noise) and target sample x_1 (e.g., robot actions), the interpolation creates a path:

$$u_s = s \cdot x_0 + (1 - s) \cdot x_1, \quad \text{where} \quad s \in [0, 1] \text{ parameterizes the interpolation} \tag{7}$$

The model learns to predict the velocity field that guides samples along this path:

$$\frac{\partial}{\partial s} u_s = g_\theta(u_s, s|y), \quad \text{where} \quad y \text{ represents conditioning inputs} \tag{8}$$

In robotics applications, y typically includes visual observations and language commands that specify the desired behavior.

The training objective teaches the model to predict the correct velocity by minimizing the difference between predicted and true flow direction:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{(y,x_1)\sim\mathcal{D},\, s\sim\mathcal{U}[0,1]} \left\| \underbrace{x_1 - x_0}_{\text{true direction}} - (1-s) \cdot \underbrace{g_\theta(u_s, s \mid y)}_{\text{predicted velocity}} \right\|_2^2 . \tag{9}$$

At inference time, samples are generated by integrating the predicted velocity field from noise ($s = 0$) to the target ($s = 1$):

$$u_{s+\Delta s} = u_s + \Delta s \cdot g_\theta(u_s, s|y), \quad \text{where} \quad \Delta s \text{ is the integration step size} \tag{10}$$

This produces the final sample $x_1 \approx u_1$. Forward Euler integration is commonly used due to its efficiency (Black et al., 2024), though more sophisticated solvers like Heun's method or Runge-Kutta can improve stability for high-dimensional control tasks (Kutta, 1901; Runge, 1895). Flow matching has demonstrated superior smoothness and precision compared to direct action prediction, particularly for temporally extended manipulation tasks (Black et al., 2024; Bjorck et al., 2025).

B Latent Action Learning

We detail our latent action learning framework in Algorithm 1, which extracts discrete action tokens from video sequences using a combination of reconstruction, perceptual, and optical flow losses. A detailed diagram of this procedure is shown in Figure 7, and the complete training configuration–including model architecture and optimization hyperparameters–is provided in Table 3.

The inverse dynamics encoder maps each frame o_t into a sequence of spatial features using a DINOv2 (Oquab et al., 2023)-initialized backbone. These features are enriched with clip-level context through factorized spatio-temporal attention layers, where the temporal branch employs bidirectional attention to aggregate information across the full sequence. The contextualized features are then discretized via Noise-Substitution Vector Quantization (NSVQ) (Vali & Bäckström, 2022), producing N_{latent} discrete codes selected from the shared codebook $\mathcal{C}$, which serve as the latent action tokens z_t.

The forward decoder mirrors this architecture with a factorized spatio-temporal transformer, but applies causal temporal attention so that predictions depend only on the past. It jointly attends to the latent action sequence $z_{0:t}$ and the observation history $o_{0:t}$ to reconstruct the next frame o_{t+1}. In addition, we integrate the action-conditioning modules proposed by (He et al., 2025) before each spatio-temporal block in the decoder to better align action tokens with visual dynamics.

Algorithm 1 Latent Action Learning (Training Step)

```
Require: Video clip of $L+1$ observations $o_{0:L} \in \mathbb{R}^{(L+1)\times H\times W\times 3}$
Require: Hyperparameters: LPIPS weight $\lambda_{\text{LPIPS}}$, Flow weight $\lambda_{\text{flow}}$, Flow start step $\alpha_{\text{flow}}$
Require: Codebook $\mathcal{C} \in \mathbb{R}^{K\times D}$ with $K$ codes of dimension $D$
Extract visual features: $f_{0:L} \leftarrow \text{DINOv2}(o_{0:L})$
Compute contextual embeddings: $h_{0:L} \leftarrow I_\beta(f_{0:L})$
for $t = 0$ to $L-1$ do
    Quantize embedding to latent: $z_t \leftarrow \text{NSVQ}(h_t, \mathcal{C})$
    Decode next frame: $\hat{o}_{t+1} \leftarrow F_\alpha(o_{0:t}, z_{0:t})$
end for

$\mathcal{L}_{\text{rec}} \leftarrow \sum_{t=0}^{L-1} \|\hat{o}_{t+1} - o_{t+1}\|_1$
$\mathcal{L}_{\text{LPIPS}} \leftarrow \sum_{t=0}^{L-1} \text{LPIPS}(\hat{o}_{t+1}, o_{t+1})$
if step $> \alpha_{\text{flow}}$ then

    $\mathcal{L}_{\text{flow}} \leftarrow \frac{1}{L}\sum_{t=1}^{L}\Big(\|\text{OF}(\hat{o}_t, \hat{o}_{t-1}) - \text{OF}(o_t, o_{t-1})\|_1 + \|\text{OF}(\hat{o}_t, \hat{o}_{t+1}) - \text{OF}(o_t, o_{t+1})\|_1\Big)$
else
    $\mathcal{L}_{\text{flow}} \leftarrow 0$
end if

$\mathcal{L}_{\text{latent}} \leftarrow \mathcal{L}_{\text{rec}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{flow}}\mathcal{L}_{\text{flow}}$
Update parameters via AdamW optimizer
```

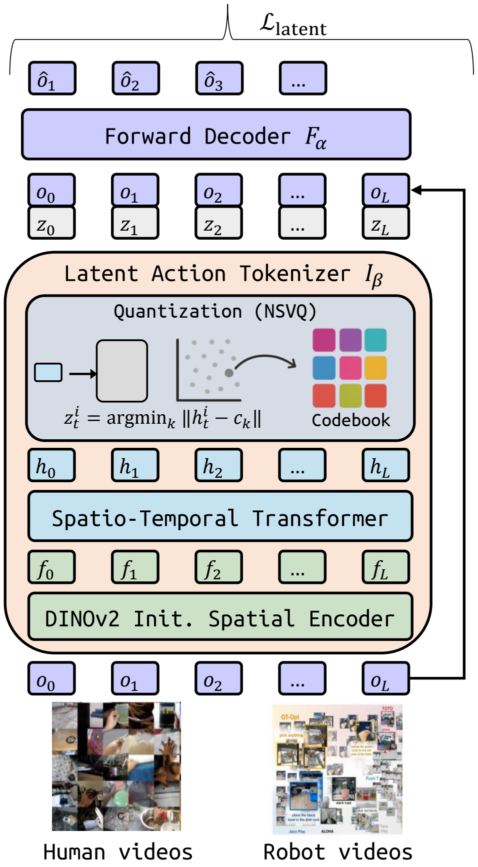


Figure 7: **Latent action learning framework.** Given a sequence of frames $o_{0:L}$, an inverse dynamics encoder $I_\beta(z_t \mid o_{0:L})$ maps the observation clip into discrete latent tokens z_t via vector quantization. A forward decoder $F_\alpha(\hat{o}_{t+1} \mid o_{0:t}, z_{0:t})$ then reconstructs the next frame $\hat{o}_{t+1}$ conditioned on the observation history and latent action sequence. Training combines reconstruction, perceptual (LPIPS), and optical flow consistency losses to ensure that the latent tokens capture physically grounded and temporally localized action information.

Hyperparameter	**Value**
Training Configuration	
Optimizer	AdamW
Base Learning Rate	1e-4
DINO Enc. Learning Rate	1e-5
Optimizer Momentum	$\beta_1, \beta_2 = 0.9, 0.99$
Batch Size	128
Grad. Norm Clip	4.0
Total Steps	240000
Image Augmentation	RandomResizeCrop
Flow Start Step α_{flow}	60000
Losses	$\mathcal{L}_{\text{rec}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{flow}}\mathcal{L}_{\text{flow}}$
LPIPS Weight λ_{LPIPS}	0.5
Flow Weight λ_{flow}	0.1 (after α_{flow})
GPU	8 Nvidia H100 (168 hours)
Inverse Dynamics Encoder I	
Backbone Init.	DINOv2 (Oquab et al., 2023)
Embedding Dim	768
Spatio-temporal Layers	6
Attention Heads	16
Attention Head Dim	64
Latent Action Quantization	
Codebook Size $\|\mathcal{C}\|$	8
Quantized Token Dim	32
Quantization Method	NSVQ (Vali & Bäckström, 2022)
Codebook Refresh Interval	Every 10 till 100, every 100 till 1000, every 1000 till 10000
Codebook Refresh Strategy	Re-init Unused, Re-shuffle Used
Forward Decoder F_α	
Embedding Dim	768
Spatio-temporal Layers	8
Attention Heads	16
Attention Head Dim	64

Table 3: Hyperparameters for latent action learning.

C Multimodal Video Pretraining with Latent Actions

We augment a pretrained multimodal video model G_θ (LWM-Chat-1M (Liu et al., 2024)) with an embedding layer E_ϕ and a decoder H_ψ for latent action processing. The model jointly predicts future visual tokens and latent action sequences, conditioned on past frames and task context.

The latent action embedding head E_ϕ maps each code $z_t \in \mathcal{C}$ into the token space of G_θ, and the decoder head H_ψ predicts next-token logits over the latent vocabulary. Training uses teacher forcing for both video tokens (from a frozen VQ-VAE) and latent tokens with cross-entropy loss. The complete training procedure and hyperparameters are detailed in Algorithm 2 and Table 4.

Algorithm 2 Multimodal Video Pretraining via Video and Latent Action Prediction

```
Require: History frames: $(o_{t-1}, o_t)$
Require: Target frame: $o_{t+H}$
Require: Task description: $c$ (text string)
Require: Labels i.e. latent action chunk $z_{t:t+H-1} \in \mathcal{C}^H$
Require: Pretrained models: VQ-VAE encoder $E_{\text{VQ}}$, video model $G_\theta$
Require: Trainable components: random initialized embedding layer $E_\phi$, decoder head $H_\psi$
Tokenize input frames: $x_{t-1}, x_t \leftarrow E_{\text{VQ}}(o_{t-1}), E_{\text{VQ}}(o_t)$
Tokenize target frame: $x_{t+H} \leftarrow E_{\text{VQ}}(o_{t+H})$
Encode text prompt: $\tilde{c} \leftarrow \texttt{Tokenizer}(c)$
Embed latent actions: $\tilde{z}_{t:t+H-1} \leftarrow E_\phi(z_{t:t+H-1})$
for $i = 1$ to $N_{\text{tokens}}$ do
    $\hat{x}^i_{t+H} \leftarrow G_\theta(\tilde{c}, x_{t-1}, x_t, \hat{x}^{<i}_{t+H})$ {Autoregressive prediction}
end for
$\mathcal{L}_{\text{img}} \leftarrow \sum_{i=1}^{N_{\text{tokens}}} \text{CE}(\hat{x}^i_{t+H}, x^i_{t+H})$ {Image token loss}
for $k = t$ to $t + H - 1$ do
    for $i = 1$ to $N_{\text{latent}}$ do
        $\hat{z}^i_k \leftarrow H_\psi(G_\theta(\tilde{c}, x_{t-1}, x_t, x_{t+H}, z_{<k}, z^{<i}_k))$
    end for
end for
$\mathcal{L}_{\text{act}} \leftarrow \sum_{k=t}^{t+H-1} \sum_{i=1}^{N_{\text{latent}}} \text{CE}(\hat{z}^i_k, z^i_k)$ {Action token loss}
$\mathcal{L}_{\text{pretrain}} \leftarrow \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{act}}$ {Total loss}
Update $G_\theta$, $E_\phi$, and $H_\psi$ using gradient descent
```

Hyperparameter	**Value**
Model Setup	
Video Model	LWM-Chat-1M (Liu et al., 2024) (initialized)
Tokenization Backbone	VQ-VAE (frozen)
Prompt Tokenizer	BPE tokenizer
Latent Action Vocabulary Size $\|\mathcal{C}\|$	8
Latent Embedding Dim (E_ϕ)	4096
Latent Decoder Type (H_ψ)	MLP
Latent Decoder Layers	1
Latent Decoder Hidden Dim	2048
Training Configuration	
Optimizer	AdamW
Learning Rate	4e-5
Weight Decay	0.0
Optimizer Betas	$(0.9, 0.95)$
Batch Size	512
Total Steps	50,000
Dropout	0.1
Gradient Clipping	1.0
Mixed Precision	`bfloat16`
GPU	8 Nvidia H100 (144 hours)
Prediction Targets	
Prediction Horizon H	14
Image Token Loss	Cross Entropy
Latent Action Loss	Cross Entropy

Table 4: Hyperparameters for multimodal video pretraining to jointly predict future visual state and latent action sequence. The video model is initialized from LWM-Chat-1M and trained jointly with lightweight latent action modules.

D Flow Matching Decoder for Continuous Control

We extend the pretrained video model G_θ with two components for continuous control. Specifically, we introduce an action encoder head E_γ that maps each continuous noisy action x_s into the model's embedding space, and an action decoder head H_η that predicts the flow field used to recover the full action chunk.

The resulting model, denoted g_ω, consists of three key components: the pretrained video model G_θ, the action encoder E_γ, and the flow decoder H_η. During training, we sample a noisy interpolation between a standard Gaussian vector and the ground-truth action chunk, and supervise the predicted flow toward the true actions using visual and task context. The full training procedure is detailed in Algorithm 3. Corresponding neural design choices and hyperparameters are include in Table 5.

Algorithm 3 Flow Matching for Continuous Control

```
Require: Image history frames $(o_{t-1}, o_t)$,
Require: Task description: $c$ (text string)
Require: Action chunk $a_{t:t+H-1} \in \mathbb{R}^{H \times D}$ {D-dimensional actions}
Require: Pretrained models: VQ-VAE encoder $E_{\text{VQ}}$, video model $G_\theta$
Require: Trainable components: action encoder $E_\gamma$, flow decoder $H_\eta$
Sample timestep $s \sim \text{Beta}(1.5, 1.0)$
Sample noise $x_0 \sim \mathcal{N}(0, I)$
Compute interpolation: $x_s \leftarrow s \cdot x_0 + (1-s) \cdot a_{t:t+H-1}$
Tokenize input frames: $x_{t-1}, x_t \leftarrow E_{\text{VQ}}(o_{t-1}), E_{\text{VQ}}(o_t)$
Encode text prompt: $\tilde{c} \leftarrow \texttt{Tokenizer}(c)$
Encode noisy actions: $f_s \leftarrow E_\gamma(x_s, s)$
Predict flow field: $\hat{g} \leftarrow H_\eta\left(G_\theta(\tilde{c}, x_{t-1}, x_t, f_s)\right)$
$\mathcal{L}_{\text{FM}} \leftarrow \|a_{t:t+H-1} - x_0 - (1-s) \cdot \hat{g}\|_2^2$ {Flow matching loss}
Update parameters of $E_\gamma$, $G_\eta$, and $G_\theta$ via gradient descent
```

Hyperparameter	**Value**
Noisy Action Encoder Head (E_γ)	
Architecture	2-layer MLP
Hidden Dim	4096
Embedding Dim (d_a)	4096
Activation	GELU
Dropout	0.1
Flow Decoder Head (G_η)	
Input Dim	7 (End-Effector Deltas) or 8 (Absolute Joint States)
Architecture	Single linear projection
Flow Matching Setup	
Interpolation Timestep s	Beta$(1.5,\ 1.0)$
Noise Distribution $\mathbf{x}_0$	Standard normal $\mathcal{N}(0, I)$
Prediction Horizon H	14
Integration Method (Inference)	Forward Euler, $N = 10$ steps
Training Configuration (follows Table 4)	
Total Steps	12000 (SIMPLER)

Table 5: Architecture and hyperparameters used for continuous control. Training settings (optimizer, schedule, etc.) match those used during pretraining.

E Simulation Benchmarks

E.1 SIMPLER Benchmark

Following prior latent action works (Ye et al., 2024b; Bu et al., 2025), we benchmark ViPRA in SIMPLER (Li et al., 2024c), an open-source suite for evaluating generalist manipulation policies. We evaluate on four Bridge tasks with a 7-DoF WidowX arm, a benchmark designed to test generalization across diverse manipulation goals. Since SIMPLER does not provide finetuning data, we collect 100 diverse multi-task trajectories using a pretrained VLA model (Ye et al., 2024b) to adapt policies before evaluation. The tasks are as follows:

- **Spoon2Cloth**: The instruction is `put the spoon on the towel`. The spoon is placed on a vertex of a 15 cm square on the tabletop, and the towel on another vertex. The spoon's orientation alternates between horizontal and vertical, requiring the robot to re-orient its gripper. This task evaluates both grasp selection and orientation adjustment.
- **Carrot2Plate**: The instruction is `put carrot on plate`. Same setup as Spoon2Cloth, but with a carrot and a plate. While similar in layout, this introduces a different geometry and surface, requiring adaptation in grasping and placement.
- **StackG2Y**: The instruction is `stack the green block on the yellow block`. A green block is placed on a vertex of a tabletop square (10 cm and 20 cm edges) and a yellow block on another. Success requires precise alignment and careful release, making it a fine-grained manipulation task that stresses stability and accuracy.
- **Eggplant2Bask**: The instruction is `put eggplant into yellow basket`. An eggplant is dropped into the right basin of a sink and a yellow basket in the left basin. The eggplant is randomized in pose but ensured to be graspable. This task evaluates robustness to shape variability and placement under uncertainty, as the object must be reliably picked and transferred across workspace regions.

We evaluate performance using two metrics: **success rate** and **partial success rate (grasp rate)**. Success rate measures whether the full task goal is completed (e.g., spoon placed on towel, block stacked without falling, eggplant deposited into basket). Grasp rate captures whether the robot is at least able to establish a successful grasp on the object, even if the subsequent placement or stacking is not achieved. This distinction is important: grasping reflects a fundamental capability for initiating manipulation, while successful completion requires the integration of grasping with precise transport and placement. Together, these metrics provide a more comprehensive view of policy competence, distinguishing between failures due to perception/grasping versus those arising from downstream control and placement.

E.2 SIMPLER Results

We report both end-to-end *success rate* and *grasp rate* in Table 1. Across **discrete actions** setting, **ViPRA-AR** attains the best average success (69.8%), exceeding LAPA (53.1%), VPT (51.0%) and OpenVLA (38.6%). It leads on precision-heavy `StackG2Y` (66.7% vs. 54.2% Scratch-AR, 45.8% VPT) and `Carrot2Plate` (62.5%), and remains competitive on `Spoon2Cloth` (66.7%, near VPT's 70.8%). On `Eggplant2Bask`, ViPRA-AR (83.3%) significantly outperforms other methods, demonstrating strong transport and placement.

In the **continuous setting**, **ViPRA-FM** achieves the highest average success (62.5%), outperforming Scratch-FM (41.7%), π_0 (27.1%), and UniVLA (42.7%). It is the strongest continuous model on `StackG2Y` (54.2%), `Carrot2Plate` (50.0%) and `Spoon2Cloth` (66.7%) while remaining competitive (79.2%) with π_0 (83.3%) on `Eggplant2Bask`. UniPI frequently generates action sequences that diverge from the given instruction in longer-horizon settings, while VPT provides only limited gains, indicating that IDM-derived pseudo-labels are sensitive to environment shifts. In contrast, ViPRA's joint use of latent action prediction and future state modeling yields stronger cross-environment transfer and more reliable task execution.

ViPRA converts grasps into task completion more reliably. On `StackG2Y`, OpenVLA achieves 70.8% grasp but only 25.0% success (a 45.8 pt gap), indicating post-grasp placement failures. ViPRA-AR maintains 66.7% grasp and 66.7% success (0 pt gap), and ViPRA-FM 62.5% grasp

Method	Success Rate
UniPI (Du et al., 2023b)	0.00
OpenVLA Kim et al. (2024)	0.54
π_0-FAST (Black et al., 2024)	0.60
π_0 (Black et al., 2024)	0.85
UniVLA (human) (Bu et al., 2025)	0.79
UniVLA (all) (Bu et al., 2025)	0.92
ViPRA	0.79

Table 6: Success rates on LIBERO-10 benchmark.

vs. 54.2% success (8.3 pt gap), evidencing stable transport and release. On `Eggplant2Bask`, OpenVLA's 91.7% grasp falls to 58.3% success (33.4 pt drop), whereas ViPRA-AR (100% → 83.3%) and ViPRA-FM (91.7% → 79.2%) show markedly smaller drops, consistent with smoother post-grasp control and accurate instruction following.

E.3 LIBERO Long Benchmark

We also evaluate on LIBERO Long (a.k.a LIBERO 10) (Liu et al., 2023), the most challenging subset of the LIBERO simulation benchmark. Unlike the Spatial, Object, or Goal subsets, LIBERO Long focuses on long-horizon manipulation tasks that require sequencing multiple sub-goals with heterogeneous objects, layouts, and task dependencies. This setting stresses robustness and temporal compositionality, since errors can accumulate across long horizons.

The evaluation consists of a suite of 10 long-horizon tasks, each paired with a natural language goal description. For example, one of the task instructions includes `"put the white mug on the plate and put the chocolate pudding to the right of the plate"`, requiring reasoning over both symbolic relations (object identities, spatial references) and low-level control. For each task, the environment is initialized with objects placed in varied locations, increasing the difficulty of generalization.

Each task is evaluated across 10 runs with 5 different random seeds, and results are reported as the average reward over all 10 tasks (500 episodes in total). This protocol provides a stringent test of both semantic grounding and long-horizon policy execution, making LIBERO Long a valuable complement to SIMPLER's shorter-horizon manipulation tasks.

E.4 LIBERO Long Results

On LIBERO-10, which emphasizes long-horizon, multi-stage manipulation, ViPRA achieves a 79% success rate. This is substantially higher than OpenVLA (54%) and π_0-FAST (60%), and close to UniVLA (90%), which is specifically optimized for LIBERO. These results demonstrate that ViPRA's motion-centric latent pretraining transfers effectively to simulated long-horizon tasks, outperforming methods trained primarily with labeled actions or direct policy supervision.

We observe that ViPRA performs reliably on coarse manipulations (e.g., cups, bowls, books), which are easy to grasp, but struggles with precision grasps such as cylindrical cans that require diameter-aligned control. We attribute this to *delta-EEF drift*: since LIBERO's action space is delta end-effector, small prediction biases can accumulate over time, leading to imprecise grasps in the absence of absolute cues to re-anchor the trajectory. For instance, π_0 mitigates this issue by conditioning on proprioceptive state history and wrist-camera inputs. Despite lacking such additional signals, ViPRA surpasses OpenVLA under the same sensing setup (image-only, delta-EEF), underscoring the benefits of motion-centric latent pretraining for long-horizon manipulation.

F ACTION OUTPUT ANALYSIS

We provide a more in-depth analysis of the action outputs of various policies introduced in Section 5.5, highlighting their differences in smoothness, consistency, and suitability for real world deployment. Policies differ in their action representations and control spaces:

- **Absolute Joint Space.** ViPRA-FM and LAPA (Ye et al., 2024a) output full 7D joint positions (Franka), directly supervised in joint space.
- **Delta End-Effector Space.** OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), and operate in 7D Cartesian delta commands (position, Euler rotations, gripper), decoded from visual inputs.
- **Continuous vs Discrete.** ViPRA-FM and π_0 (Black et al., 2024) predict continuous actions via a flow matching decoder, whereas LAPA (Ye et al., 2024a) and OpenVLA (Kim et al., 2024) use quantized logits over discretized action bins.

To better understand the behavioral differences between discrete and continuous policies, we analyze the predicted action trajectories across different models during closed-loop visual rollout on real robot observations. We evaluate policies by loading their finetuned checkpoints into our inference pipeline and simulating replay on the training trajectories from the finetuning dataset. This allows us to visualize their motor command trends without introducing new generalization factors. In particular, we compare ViPRA-FM (ours), LAPA (Ye et al., 2024a), OpenVLA (Kim et al., 2024), and π_0 (Black et al., 2024).

LAPA (Ye et al., 2024b) and OpenVLA (Kim et al., 2024) rely on a discretization scheme in which each dimension of the robot's action space is uniformly quantized into 255 bins using equal-sized quantiles over the training distribution. That is, for each joint or end-effector dimension, bin boundaries are chosen so that each bin contains roughly the same number of training points. This quantile-based discretization ensures equal data coverage across bins but introduces two key limitations in how actions are represented and learned:

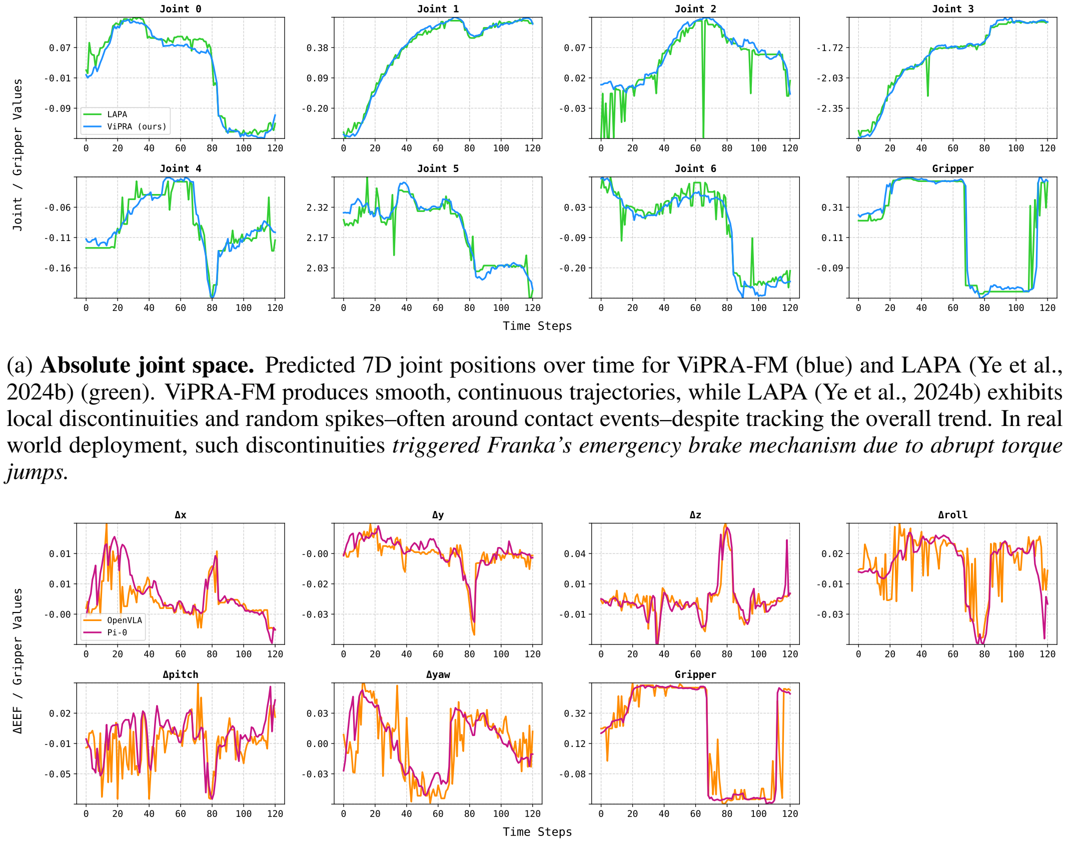
1. **Contact-Sensitive Flipping:** At test time, small perturbations in the input (e.g., due to occlusions or slight viewpoint drift) may cause the model to flip from one bin to another near the quantile boundary–especially at contact points. Since adjacent bins can correspond to different action magnitudes, these minor visual shifts can lead to abrupt discontinuities in motor output.
2. **Loss Granularity:** The cross-entropy loss used for training treats each action bin as a distinct class label. As a result, all incorrect predictions are penalized equally, regardless of how close they are to the ground-truth bin. For example, predicting bin 127 instead of 128 incurs the same loss as predicting bin 0. This is fundamentally at odds with the structure of continuous action spaces, where the cost of an error should scale with its magnitude.

We hypothesize that the combination of bin boundary flipping and non-metric loss leads to the spiky or erratic behavior seen in discrete action models, particularly around moments of contact or high-frequency motion. These effects are amplified in high-dimensional control settings, where discretization artifacts can arise independently in each action dimension–compounding into visibly unstable or jerky behaviors across the full joint trajectory.

By contrast, continuous policies such as ViPRA-FM and π_0 (Black et al., 2024) operate directly in $\mathbb{R}^D$ using flow matching. These losses naturally reflect the structure of the action space–penalizing predictions in proportion to how far they deviate from the ground truth. As a result, the output trajectories tend to be smoother, better aligned with demonstrations, and more robust to perceptual jitter.

We note that it may be possible to mitigate some of the above issues by increasing the number of bins or by using non-uniform binning schemes (e.g., higher resolution in frequently visited regions). However, these approaches increase model complexity and still inherit the fundamental limitation of using classification loss in a regression setting. Continuous decoders trained with distance-aware objectives offer a more natural and principled solution for low-level control.

To gain deeper insight into how different action representations influence control behavior, we examine the temporal structure of predicted actions across several rollout trajectories. We organize



(a) **Absolute joint space.** Predicted 7D joint positions over time for ViPRA-FM (blue) and LAPA (Ye et al., 2024b) (green). ViPRA-FM produces smooth, continuous trajectories, while LAPA (Ye et al., 2024b) exhibits local discontinuities and random spikes–often around contact events–despite tracking the overall trend. In real world deployment, such discontinuities *triggered Franka's emergency brake mechanism due to abrupt torque jumps.*

(b) **Delta end-effector space.** Predicted 7D delta actions (position, rotation, gripper) for π_0 (Black et al., 2024) (magenta) and OpenVLA (Kim et al., 2024) (orange). Although delta control provides structured low-level modulation, OpenVLA exhibits sharp fluctuations due to discretized output. Notably, the gripper signal shows large, momentary switches during contact events–resulting in failed grasps or premature object drops. In contrast, π_0 maintains stable gripper behavior during fine manipulation.

Figure 8: Visualization of predicted actions across different control spaces. Discrete policies often produce sharp discontinuities due to binning artifacts and classification loss, whereas continuous policies exhibit smoother, dynamics-consistent behavior.

the analysis by control space–absolute joint angles vs. delta end-effector motions–and visualize per-dimension action trends across time in Figure 8.

These visualizations support our hypothesis: *discrete policies, trained with cross-entropy over fixed bins, tend to produce abrupt transitions around perceptually sensitive regions*–especially near bin boundaries or occlusions. This manifests as random spikes, high-frequency jitter, or contact-time instability, all of which can destabilize robot behavior in deployment.

In contrast, *continuous policies like* ViPRA-FM *and* π_0*, trained with flow matching losses, yield consistently smooth, physically plausible actions that better reflect real world constraints.* The ability to interpolate naturally between states–not just classify them–proves critical for robust closed-loop performance in contact-rich manipulation.

G REAL WORLD EXPERIMENTS: SETUP, CHALLENGES, AND OBSERVATIONS

To complement our real world results in Section 5.4, we provide additional details on our hardware setup, task design, and policy behavior under realistic sensing and control constraints. We also analyze generalization to unseen objects, retry behavior, and how chunked continuous actions support efficient closed-loop control.

G.1 HARDWARE AND DATA COLLECTION SETUP

All experiments are conducted on a real world robotic platform with two 7-DOF Franka Emika Panda arms. The workspace is observed by a single front-mounted ZED stereo camera. There are no wrist-mounted or side-view cameras, so all perception is monocular and from a fixed third-person viewpoint. We use the GELLO teleoperation system (Wu et al., 2023b) to collect human demonstrations at 15Hz. Demonstrations are collected directly in task-relevant environments, with each policy trained using only a single camera view.

Our decision to use image history as part of the observation is motivated by the inherently temporal nature of the video model architecture, as well as the absence of auxiliary views. Stacking observations over time allows the model to internally infer dynamics and compensate for occlusions or ambiguous single-frame cues.

G.2 TASK DESCRIPTIONS AND CHALLENGES

We evaluate policies on three real world single-arm tasks, each with unique control and perception challenges (Figure 9):

1. **Cover-Object:** The robot must pick up a piece of cloth and drape it over a specified object. This task is challenging due to the deformable nature of cloth, which requires reliable grasping from the table surface. Slight changes in cloth configuration or object geometry can affect dynamics drastically. Generalization requires reasoning over unseen cloth textures and novel target objects.
2. **Pick-Place:** The robot must pick up a named object (e.g., sponge, bowl, duck) and place it on a destination surface (plate or board). Object shapes vary significantly, leading to different grasp affordances. Grasping a wide bowl vs. a narrow-handled cup requires distinct motor strategies. The task is highly multimodal–there are multiple correct ways to perform the task, depending on object shape, pose, and placement surface.
3. **Stack-Cups:** The robot must follow language instructions to stack a cup of color1 onto a cup of color2. Success requires grounding object properties and executing precise stacking. Evaluation setups include unseen cup types, color shades, and geometries to test language understanding and spatial generalization.

G.3 GENERALIZATION TO NOVEL OBJECTS

A core goal of our real world evaluation is to assess how well the policy generalizes to unseen object instances and configurations not encountered during training. We design test-time setups that introduce meaningful variation across tasks:

- **Cover-Object:** Test scenarios include cloths of varying texture, size, and stiffness, as well as new target objects such as jars, boxes, and toys. These variations require the policy to generalize grasp strategies and adapt to deformable material dynamics.
- **Pick-Place:** We evaluate on previously unseen objects with diverse geometries and affordances (e.g., bowls, mugs, fruits), and destination surfaces of varying size and texture. The task requires flexible grasping and reliable placement across a range of object shapes and destination surfaces.
- **Stack-Cups:** Evaluation includes new cup types with unseen shapes, rim sizes, and fine-grained color variations. The policy must generalize language grounding to new color references and execute precise stacking across novel physical configurations.

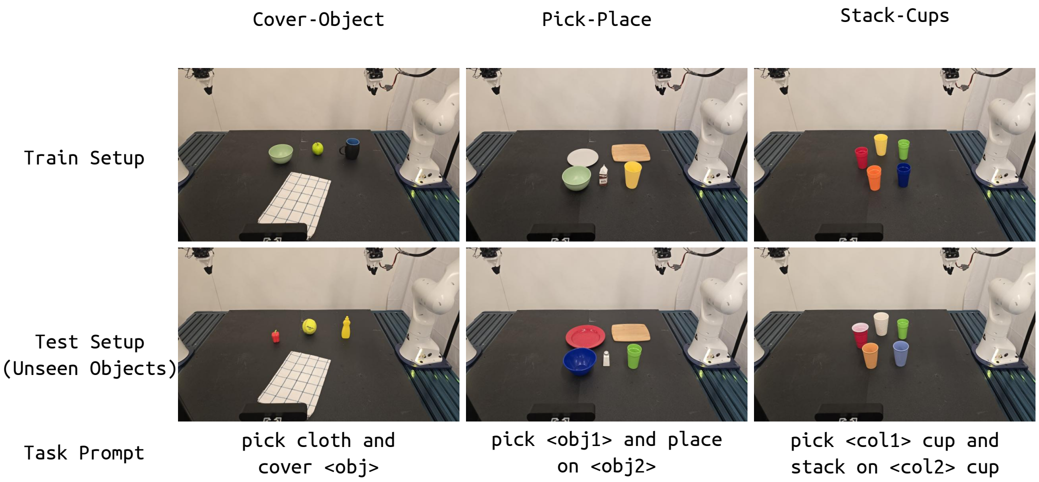

Cover-Object
Pick-Place
Stack-Cups
Train Setup
Test Setup
(Unseen Objects)
Task Prompt
pick cloth and cover <obj>
pick <obj1> and place on <obj2>
pick <col1> cup and stack on <col2> cup

Figure 9: **Task Setup Overview.** (Top row) Training environments for each of the three single-arm manipulation tasks: `Cover-Object`, `Pick-Place`, and `Stack-Cups`. (Bottom row) Evaluation environments featuring novel objects, textures, or placements not seen during training. Note the variety in cloth shape, object geometry, plate type, and cup color/size combinations.

Despite these shifts, *our method consistently exhibits robust generalization across all tasks*. We attribute this to the combination of latent dynamics pretraining, language-conditioned perception, and a unified architecture that integrates semantic, spatial, and temporal cues. Pretraining on diverse unlabeled videos teaches the model general priors about object motion and interaction. Conditioning on task instructions guides object selection and interpretation even in ambiguous or unfamiliar contexts. Finally, the architectural design ensures that learned representations capture not just appearance, but how objects behave across time, enabling transfer to new instances that were not explicitly seen during supervised finetuning.

G.4 Retrying Behavior Enabled by Temporal Pretraining

Our method consistently *exhibits robust retry behavior*: when an initial grasp attempt fails, due to occlusion, misalignment, or object shift, the policy often reattempts until successful. This is especially evident in `Cover-Object`, where the robot frequently retries grasping if the cloth slips, and in `Pick-Place`, where wide or irregularly shaped objects like bowls may require multiple grasp attempts from different angles.

We attribute this robustness to our temporal pretraining objective. By learning to predict future video frames and latent actions over multiple steps, the model develops a sense of longer-horizon dynamics and recoverability. Rather than depending on single-step feedback, it implicitly plans through extended temporal context–enabling it to course, correct and persist through partial failures.

G.5 Action Chunking and Inference Efficiency

ViPRA produces continuous actions using a chunked flow matching decoder, generating sequences of 14 actions per inference step. At test time, we cap control frequency by evaluating two rollout strategies: **7/14 rollout**, where the first 7 actions of each chunk are executed before re-planning, and **14/14 rollout**, where all 14 actions are executed before the next inference. The former corresponds to an effective closed-loop update rate of $\sim$3.5 Hz, while the latter doubles this to 7 Hz. Because predicted action trajectories are smooth and temporally coherent, ViPRA remains stable even under open-loop execution within each chunk. This property is particularly beneficial for contact-rich phases that demand reactive yet jitter-free behavior.

KV caching for fast inference We further optimize inference with key-value (KV) caching. Language and image attention states are cached once and reused across flow matching Euler steps, so only action tokens are recomputed during integration. This reduces redundant computation, enabling the entire 14-step chunk to be produced in 510 ms ($\sim$1.95 Hz), which corresponds to a robot-side control frequency of up to 22 Hz. Our setup can stably support control rates approaching 20 Hz, to our knowledge matched only by one other 7B-parameter model (Kim et al., 2025).

Comparison with baselines. Table 7 summarizes model sizes, action rollout lengths, and inference times. Unlike prior approaches that also use a 7B model (e.g., LAPA and OpenVLA) and operate at ∼200 ms per step but predict only single actions, ViPRA amortizes inference across long, smooth action chunks, enabling high frequency reactive control.

Method	**Model Size**	**Action Steps**	**Inference Time (ms)**
LAPA (Ye et al., 2024b)	7B	1	220
OpenVLA (Kim et al., 2024)	7B	1	190
π_0 (Black et al., 2024)	3.3B	16	90
UniPI (Du et al., 2024)	–	16	24000
UVA (Li et al., 2025)	0.5B	16	230
ViPRA (ViPRA-FM)	7B	14	510

Table 7: Inference speed comparison across models. ViPRA achieves high effective control frequencies by amortizing computation over action chunks.

H ViPRA-FM on Challenging Bimanual Tasks

Bimanual manipulation introduces significant complexity beyond single-arm control. The combined action space spans 14 degrees of freedom, and inter-arm coordination requires precise spatial alignment, collision avoidance, and timing consistency. The solution space is also highly multimodal–there are many valid ways to execute a task depending on object geometry, initial configurations, and movement variability. These challenges make bimanual tasks a strong test of a policy's ability to generalize and coordinate under real world constraints.

H.1 Bimanual Setup

We test our framework using both arms of the Franka Panda robot. While only the right arm performs active grasping, both arms are controlled jointly using a single policy conditioned on shared language instructions. The system receives monocular observations from a front-mounted ZED camera and generates chunked continuous actions for both arms in a synchronized control loop.

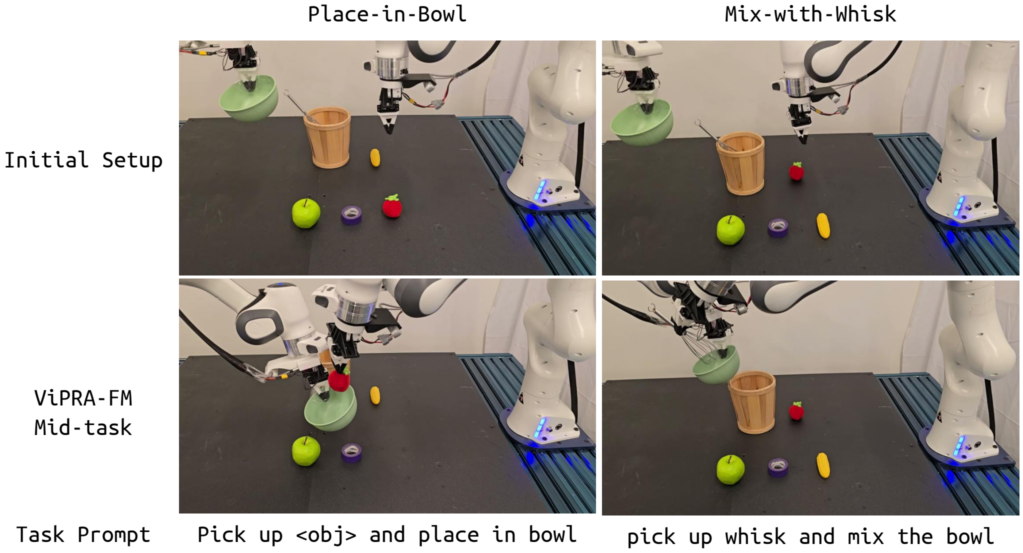


Figure 10: **Bimanual task execution by ViPRA-FM.** (Top row) Initial setup for the two tasks: placing a tomato into a bowl and mixing with a whisk. (Bottom row) Mid-execution rollout of ViPRA-FM: the right arm transports the tomato toward the bowl held by the left arm (left), and mixes the contents using the whisk while the left arm maintains bowl stability (right). These examples highlight coordinated two-arm control and fluent execution of tool- and object-handling behaviors.

We evaluate two bimanual tasks of increasing complexity:

(1) Place-in-Bowl: The right arm must grasp a target object (e.g., a fruit or kitchen item) and place it into a bowl held by the left arm. Success requires fine-grained spatial alignment above the bowl, smooth object transfer, and collision-free approach and retreat trajectories in close proximity to the support arm.

(2) Mix-with-Whisk: The right arm retrieves a whisk from a nearby basket, mixes the contents of the bowl, and returns the whisk to its original location. This task involves tool use, curved and sustained motion, and close-proximity coordination with the left arm, which dynamically maintains the bowl pose throughout the sequence.

These tasks pose significant challenges for real world bimanual coordination. Both arms must operate in close proximity, requiring precise spatial alignment to avoid collisions–especially during approach and retreat phases. With only a single fixed camera and no wrist-mounted sensors, the policy must infer depth and object interactions purely from visual input. Timing mismatches or calibration drift between the arms can further compound errors, making successful execution sensitive to both perception and control stability.

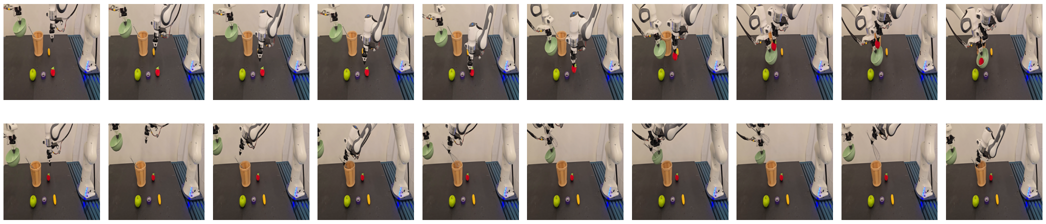


Figure 11: **ViPRA-FM rollouts in real world bimanual tasks.** Top: `Place-in-Bowl` - the robot picks up a tomato and places it into a bowl held by the left arm. Bottom: `Mix-with-Whisk` - the robot retrieves a whisk, stirs the bowl contents, and returns the tool. Each sequence shows 10 evenly spaced frames sampled from real world executions.

H.2 Bimanual Results

ViPRA-FM is deployed using 14-step action chunks, executed at 7Hz control frequency. This high-frequency chunked control allows the policy to maintain smooth, temporally coherent trajectories while remaining responsive to changing visual inputs. The model also receives short history windows as input, which helps stabilize motion during contact-heavy transitions and multi-step interactions.

In `Place-in-Bowl`, the robot completes 10 out of 18 trials. Failures were primarily due to unsuccessful grasps caused by the limited span and compliance of our custom 3D-printed gripper, not the bimanual coordination itself. In all successful grasps, the object was consistently placed into the bowl without collision or instability. This suggests that the policy reliably handles the spatial reasoning and coordination demands of the task, with grasp robustness being the primary bottleneck, a limitation that could be mitigated with a more capable gripper design.

In `Mix-with-Whisk`, the robot completes 8 out of 12 trials. The task involves sustained, curved motion in close proximity to the left arm, requiring continuous spatial alignment between the whisk and bowl. The policy leverages temporal history to stay anchored to the mixing target and uses its chunked control output to produce smooth stirring behavior. The whisk's small, symmetric handle makes it easier to grasp, allowing the policy to focus on trajectory accuracy and contact stability throughout the sequence.

Together, these results demonstrate that ViPRA-FM is capable of executing complex bimanual tasks using a single vision-conditioned policy and continuous action generation. Additional results, comparisons, and rollout videos will be shared on our project website. `https://vipra-robot.github.io`.