Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

mimic-video: Video-Action Models for Generalizable Robot Control Beyond VLAs

Jonas Pai*[1,2,3], Liam Achenbach*[1,2,3] Oliver Sanchez[1], Stefanos Charalambous[1],
Victoriano Montesinos[1], Benedek Forrai[1], Oier Mees[2,5] and Elvis Nava[1,3,4]
*Core Contributors
[1]mimic robotics [2]Microsoft Zurich [3]ETH Zurich [4]ETH AI Center [5]UC Berkeley
https://mimic-video.github.io

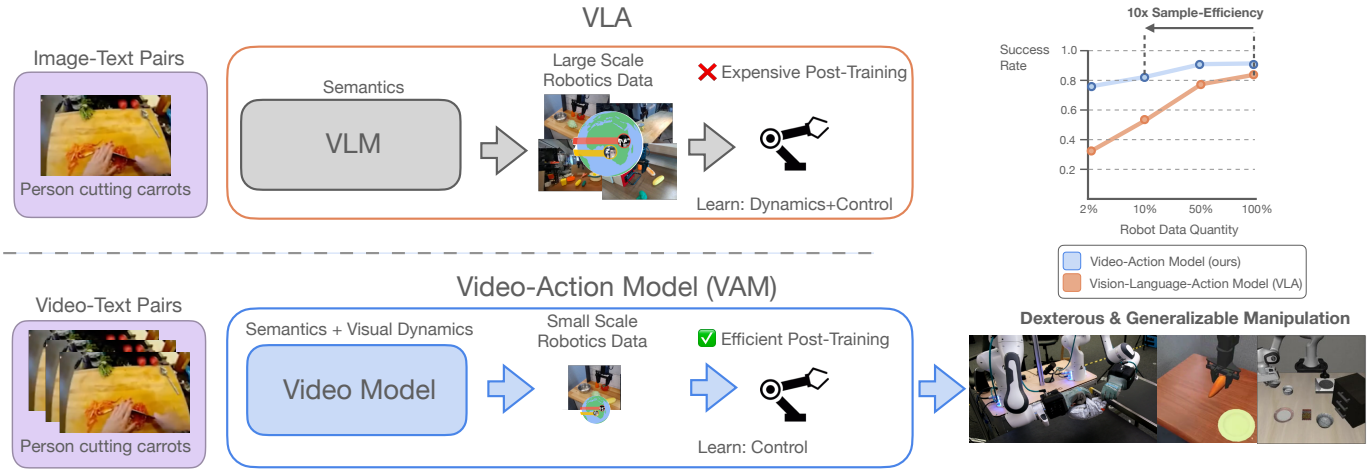

Fig. 1: We introduce mimic-video, a new class of Video-Action Model (VAM) that grounds robotic policies in pretrained video models. Unlike standard VLAs that must learn physical dynamics from scratch (top), mimic-video leverages the inherent visual dynamics of video backbones to isolate the control problem (bottom). This enables state-of-the-art performance on dexterous manipulation tasks, while achieving 10x greater sample efficiency compared to VLAs (right).

***Abstract*—Prevailing Vision-Language-Action Models (VLAs) for robotic manipulation are built upon vision-language backbones pretrained on large-scale, but disconnected static web data. As a result, despite improved semantic generalization, the policy must implicitly infer complex physical dynamics and temporal dependencies solely from robot trajectories. This reliance creates an unsustainable data burden, necessitating continuous, large-scale expert data collection to compensate for the lack of innate physical understanding. We contend that while vision-language pretraining effectively captures semantic priors, it remains blind to physical causality. A more effective paradigm leverages video to jointly capture semantics and visual dynamics during pretraining, thereby isolating the remaining task of low-level control. To this end, we introduce mimic-video, a novel Video-Action Model (VAM) that pairs a pretrained Internet-scale video model with a flow matching-based action decoder conditioned on its latent representations. The decoder serves as an Inverse Dynamics Model (IDM), generating low-level robot actions from the latent representation of video-space action plans. Our extensive evaluation shows that our approach achieves state-of-the-art performance on simulated and real-world robotic manipulation tasks, improving sample efficiency by 10x and convergence speed by 2x compared to traditional VLA architectures.**

I. Introduction

Building on the capabilities of pretrained Vision-Language Models (VLMs), Vision-Language-Action Models (VLAs) transfer semantic knowledge acquired through Internet-scale vision-language pretraining to the physical domain. By fine-tuning a VLM on diverse robot action data, VLAs learn generalist, natural language-conditioned robot manipulation policies that combine a wide range of skills and exhibit impressive generalization to unseen instructions, objects, and environments [52, 22, 3, 21].

However, this paradigm faces a fundamental limitation: the pretraining data, while massive in scale, is inherently static. Images and text lack explicit, temporally-grounded information about dynamics and physical procedures that are crucial for complex manipulation. Consequently, the burden of learning physical dynamics (how objects move, deform and interact) falls entirely on the post-training stage, where the model must infer these from scarce and expensive expert-teleoperated demonstrations. This heavy reliance on robot data creates a data-efficiency bottleneck that limits scalability. While prior works have explored augmenting VLA training with auxiliary video-derived signals such as language plans, affordances, or keypoints [48, 16, 20, 21], reducing dense video into such sparse representations creates an information bottleneck, failing to capture fine-grained dynamics.

In this work, we posit that the key to more sample-efficient and capable robot policies lies in leveraging a pretraining modality that inherently encodes dynamic, procedural information: video. Unlike static image-text pairs, internet-scale

video data provides rich knowledge on "how things are done", capturing the nuanced physics of interaction, how objects move, deform and react to forces. However, effectively harnessing this data remains a significant challenge. Prior approaches leveraging pretrained video models typically learn the joint distribution over video and actions, factorized such that the predicted actions are conditioned on synthesized future frames [26, 25, 12]. However, recovering the policy typically requires fully generating these future frames, necessitating prohibitive video synthesis at every control step.

To address these limitations, we propose a more direct paradigm: grounding robot policies directly in the latent representations of a generative, pretrained video model. We introduce mimic-video, a novel Video-Action Model (VAM) that unifies video modeling with robot control. Built upon a state-of-the-art video diffusion backbone, mimic-video functions by first synthesizing a visual plan: given an initial observation and language instruction, the video backbone predicts a future trajectory within a compact latent space. Rather than requiring full or even partial video generation, we extract intermediate video model representations to condition a downstream action decoder. This decoder operates as an Inverse Dynamics Model (IDM) to recover low-level motor commands. This formulation allows the video backbone to remain frozen, eliminating the need to train it on scarce robot action data. Fundamentally, this architecture decouples the inherent multi-modality of long-horizon planning, now offloaded to the video backbone, from the downstream control task. This effectively frees the action decoder from modeling complex future distributions, allowing it to dedicate its entire capacity to the far simpler, unimodal and non-causal problem of inverse dynamics [30, 31].

Our primary contribution is mimic-video, a novel generalist robot policy that integrates generative video pretraining with flow matching-based control, establishing a new class of methods we term Video-Action Models (VAMs). We evaluate our approach across a diverse suite of robotic embodiments ranging from standard single-arm manipulation to bimanual dexterous tasks, demonstrating state-of-the-art results in both simulated benchmarks and challenging real-world environments. Our mimic-video model achieves this performance while improving sample efficiency by 10x and convergence speed by 2x compared to traditional VLA architectures.

II. Related Work

a) Vision-Language-Action (VLA) Models: A major breakthrough in robot learning has been the paradigm of Vision-Language-Action (VLA) models, which are obtained by fine-tuning large, pretrained Vision-Language Models (VLMs) on robotics data. Models like RT-2 [52], OpenVLA [22], and the $\pi_0/\pi_{0.5}$ series [3, 37, 21] leverage the vast semantic knowledge embedded in their backbones from pretraining on internet-scale image-text data. This allows them to follow open-ended language instructions and generalize to novel environments and tasks in a zero-shot fashion. However, a fundamental limitation of VLAs is the lack an inherent model of dynamics, physics, or temporal progression, limiting their ability to reason about the physical consequences of actions.

Several works [48, 49] make use of techniques like Chain-of-Thought reasoning [46] in order to extract more useful grounded conditioning signals and representations [6, 20, 49] for VLAs. These approaches typically result in significantly slower inference due to the computation of autoregressive plans before action decoding.

b) Video Modeling for Policy Learning: The utilization of video prediction for robotic control has a long-standing history, primarily motivated by the potential to enable planning through visual foresight. Early works, such as the seminal approaches by Oh et al. [35], Watter et al. [45], Fragkiadaki et al. [15], Finn et al. [14], Finn and Levine [13], demonstrated how video prediction could enhance physical interaction. The use of action-conditioned video models (world models) for policy learning has recently seen significant adoption. World models can help select more optimal action sequences at runtime by "imagining" their outcome [1, 38], or be used as learned simulators for evaluation and DAgger-like [41] approaches [17, 42]. In this work, we consider non action-conditioned video models. One line of work fully generates pixel-space future video and obtains actions either via non-parametric methods [25], or learned pixel-based Inverse Dynamics Models [11, 12]. CoT-VLA [49] generates a subgoal image and actions in one autoregressive sequence. Another line of work learns to model video during training without predicting video in action sampling; Unified World Models [51] learns a model from scratch that can flexibly function as a policy, a video prediction model, or a forward or inverse dynamics model and LAPA [47] finetunes a VLM to predict visual "latent actions", re-training only the output layer to predict actions in a subsequent training stage. FLARE [50] aligns intermediate VLA representations with future vision-language embeddings. Similarly to our approach, Video Policy [26] explicitly models the joint video-action distribution and conditions a policy model on intermediate video model representations, but crucially does not allow for efficient sampling of the marginal action distribution.

Our proposed approach departs from most prior work by directly grounding control in the rich latent priors of internet-scale video models, rather than training from scratch or relying on pixel-level reconstruction. Additionally, by conditioning a lightweight inverse dynamics model on intermediate, noisy latent states, we bypass the computational cost of full video generation and the brittleness of heuristic tracking. This enables a scalable, end-to-end framework that effectively transfers broad physical understanding to downstream manipulation tasks, significantly reducing the reliance on expensive, large-scale robotic demonstrations.

III. Case Study: How does Video Generation Quality Affect Robot Policy Performance?

In this work, we argue that the internal representations arising in pretrained video model backbones are better suited for downstream robot learning compared to those in the

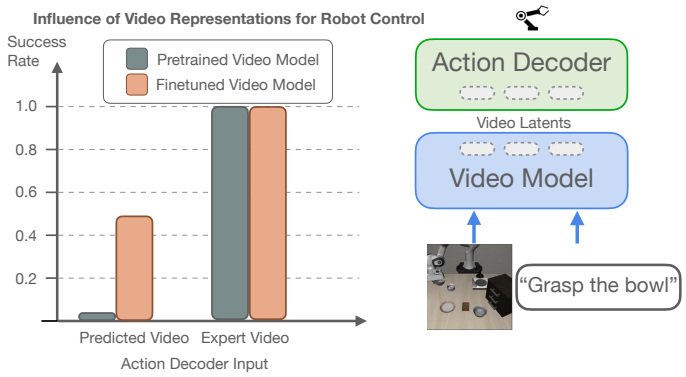

Fig. 2: We compare success rates when conditioning our action decoder on different visual inputs: video latents generated by either predictions or ground-truth (expert) video for both features from a standard pretrained video model (gray), as well as a video model finetuned on video data from the robot dataset (orange). The near-perfect performance with ground truth inputs confirms that control effectively reduces to visual prediction, implying policy performance scales directly with video model quality.

VLM backbones commonly used in current state of the art VLAs. Intuitively, video models jointly model images, physical dynamics, alongside visual action plans. A policy trained on such representations effectively reduces the role of the action decoder to a simple translator, mapping visual action plans into a low-dimensional robot action trajectory. If this hypothesis holds, the bulk of learning in Video-Action Models falls on the large-scale video pretraining and finetuning phases, while training the action decoder (the step requiring expensive, high-quality robot teleoperation data) becomes lightweight and data efficient.

We investigate this claim by conducting an "oracle" case study (see Fig. 2), where we disentangle the difficulty of *predicting* the future in robotic control tasks from *executing* it. Concretely, we train an action decoder on top of video representations and evaluate its performance under different conditioning regimes. We compare success rates when the decoder is conditioned on predicted video latents, from either a standard off-the-shelf video model or one finetuned on robotics data, versus "oracle" latents extracted from ground-truth future video frames. We observe a pronounced scaling behavior: while minimizing the domain gap via finetuning leads to improved performance when using *predicted* video, conditioning on *oracle* latents yields near perfect success rates regardless of whether the underlying backbone is finetuned on the target distribution or not. Notably, this finding suggests that a high-quality pretrained video model backbone provides extremely rich representations for action decoding, sufficient on their own to perfectly decode low-level action plans with a decoder trained on minimal low-level action finetuning data. Consequently, the burden for policy learning in VAMs effectively shifts away from low-level action decoding towards video model pretraining and finetuning.

IV. Video-Action Models

We introduce mimic-video, a generative Video-Action Model (VAM) capable of modeling the joint distribution of video and robot actions. Our architecture couples two Conditional Flow Matching (CFM) models: a pretrained, language-conditioned video backbone and a lightweight action decoder that functions as an Inverse Dynamics Model (IDM) by conditioning on the video model's latent representations.

A. Preliminaries: Flow Matching

Both the video and action prediction components are trained using the Flow Matching framework [27] to model a data distribution $p_0(x^0)$ by constructing a Continuous Normalizing Flow [5]. We use the conditional optimal transport path

$$x^\tau = (1-\tau)x^0 + \tau\varepsilon, \quad \tau \in [0,1] \tag{1}$$

which interpolates between clean data x^0 (at $\tau = 0$) and Gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$ (at $\tau = 1$) to define the conditional probability path $p_\tau\left(x^\tau \mid x^0\right)$. The model parameterizes an estimator v_θ to the intractable marginal generating vector field

$$u_\tau(x^\tau) = \mathbb{E}_{p(x^0|x^\tau)} u_\tau(x^\tau \mid x^0)$$

where $u_\tau(x^\tau \mid x^0) := \frac{d}{d\tau}x^\tau = \varepsilon - x^0$ is termed the conditional generating vector field and can be computed trivially for samples x^0, ε. The power of flow matching lies in learning v_θ by regressing to $u_\tau(x^\tau \mid x^0)$:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{\mathcal{T}(\tau),\, p_0(x^0),\, p_\tau(x^\tau|x^0)} \left\| v_\theta(x^\tau, \tau) - u_\tau(x^\tau \mid x^0) \right\|^2, \tag{2}$$

where the expectation is taken over a distribution $\mathcal{T}$ of flow times τ, which is $\mathcal{U}([0,1])$ in [27] and will take different values in this work.

Inference is performed by integrating the learned field v_θ from $\tau = 1$ to $\tau = 0$ to recover $\hat{x}^0 \sim p_0$:

$$\hat{x}^0 = \varepsilon + \int_1^0 v_\theta(\hat{x}^\tau, \tau) d\tau \tag{3}$$

Critically, this continuous time parameter τ allows us to define *partial denoising* (stopping at intermediate $\tau > 0$), which is central to our method.

B. Architecture Formulation

Formally, we aim to learn a generalist robot policy $\pi(\mathbf{A}_t \mid \mathbf{o}_t, l)$ that predicts a sequence of actions $\mathbf{A}_t = [\mathbf{a}_t, \ldots, \mathbf{a}_{t+H_a-1}]$ given observations consisting of multiple RGB images $\mathbf{I}_{t'}$, a language instruction l and the robot's proprioceptive state $\mathbf{q}_t$, such that $\mathbf{o}_t = [\mathbf{I}_{t-H_o+1}, \ldots, \mathbf{I}_t, l, \mathbf{q}_t]$.

Our model consists of two flow matching-based models trained using the objective defined in Eq. 2. Let $\mathbf{z}_t^0$ be the sequence of video encodings and $\mathbf{A}_t^0$ be the clean action chunk:

Video Model: $v_\phi(\mathbf{z}^0_{\text{past}}, \mathbf{z}^{\tau_v}_{\text{future}}, l, \tau_v)$ induces $p_\phi(\mathbf{z}^0_{\text{future}} \mid \mathbf{z}^0_{\text{past}}, l)$.

Action Policy: $\pi_\theta(\mathbf{A}_t^{\tau_a}, \mathbf{q}_t, \mathbf{h}^{\tau_v}, \tau_a, \tau_v)$ induces the action distribution $p_\theta(\mathbf{A}_t^0 \mid \mathbf{q}_t, \mathbf{h}_t^{\tau_v}, \tau_v)$.

Here, $\mathbf{h}^{\tau_v} = v_\phi^{(k)}(\mathbf{z}^0_{\text{past}}, \mathbf{z}^{\tau_v}_{\text{future}}, l, \tau_v)$ is the vector of hidden states extracted after the k^{th} layer of the video model when

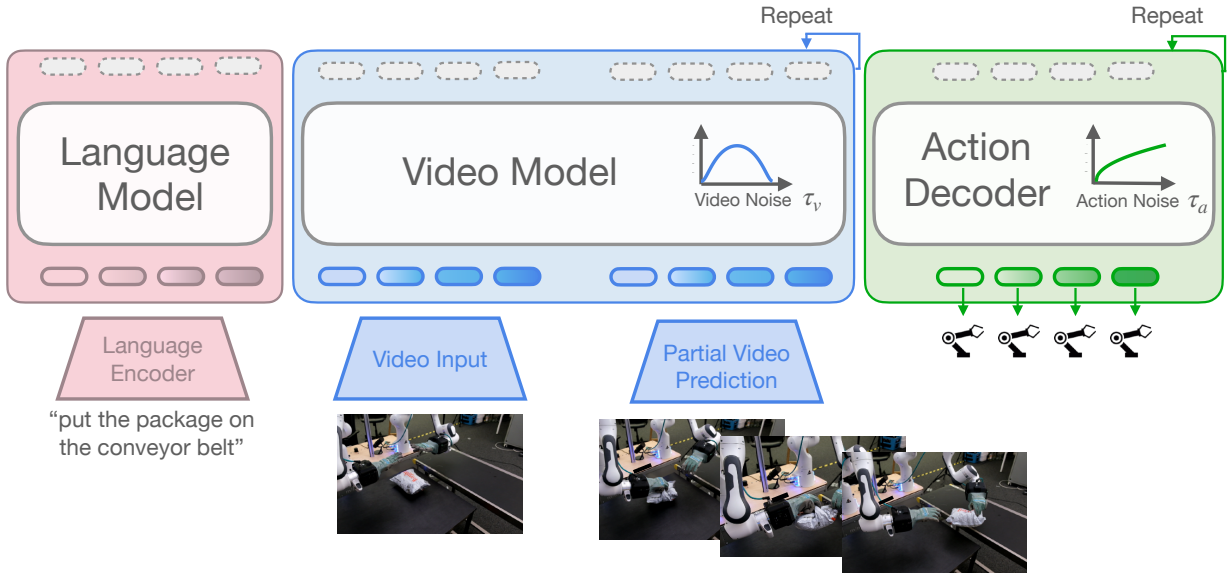

Fig. 3: mimic-video architecture: we instantiate our framework with a pretrained video generation backbone (Cosmos-Predict2 [34, 33]), which provides rich physical dynamics priors learned from large-scale video data. We adapt this model for control via a *partial denoising* strategy, where the video backbone follows the flow to an intermediate flow time τ_v to extract latent visual plans. These representations condition a smaller action decoder, which processes proprioceptive states and predicts action trajectories. The video and action components operate on independent flow schedules (τ_v and τ_a), allowing us to design the learning problem separately for each modality.

invoking it on "noisy" video input $\mathbf{z}^{\tau_v}_{\text{future}}$ (computed via Eq. 1) at flow-time τ_v. We illustrate our architecture in Fig. 3.

C. Video Model

While our Video-Action Model formulation can be instantiated with any flow matching-based video model, in practice we use Cosmos-Predict2 [34, 33] as our base model. Cosmos-Predict2 is an open-source 2B latent Diffusion Transformer (DiT) [36] model that operates on a sequence of video frames encoded by a pretrained autoencoder. The input to the model is a concatenation of clean latent frames that form a context prefix and "noisy" latent frames representing future video. Each transformer layer alternates between (1) self-attention over the full video sequence, (2) cross-attention to language instructions encoded by T5 [39], and (3) a two-layer MLP.

D. Action Decoder

The action decoder is instantiated as a DiT that encodes the robot's proprioceptive state $\mathbf{q}_t$ and a sequence of $\mathbf{A}_t$ future robot actions through two separate MLP networks and concatenates them to form the action decoder's sequence dimension. We use learned absolute positional encodings to add temporal information to each token. During training, we randomly replace the soft token encoding the proprioceptive state with a learned mask token to prevent overfitting on the low dimensional observation. Each action decoder layer consists of (1) cross-attention to intermediate video model representations $\mathbf{h}^{\tau_v}$, (2) self-attention over the action sequence, and (3) a two-layer MLP. Each component is bypassed by a residual path and each component's output is modulated via AdaLN [36], where the input to the AdaLN projections is a low-rank bilinear-affine encoding of both video and action flow times τ_v and τ_a. In our experiments, we approprioate 500M parameters to the action decoder, bringing the total to 2.46B.

E. Action Sampling

To enable real-time control, we formulate inference as efficient sampling from the marginal action policy. Although mimic-video is in principle capable of sampling from the joint video-action distribution (see Fig. 4 for an example), we can sample from the marginal action distribution more efficiently by bypassing the computational cost of full video reconstruction. We therefore propose a *partial denoising* strategy that extracts semantic features from intermediate flow states without resolving fine-grained pixel details. Our inference action sampling procedure is described in Algorithm 1. Given image observations $\mathbf{o}_t$, we integrate the video flow field from Gaussian noise to an intermediate flow time τ_v (see Eq. 3). This yields a partially denoised latent state $\mathbf{z}^{\tau_v}_{\text{future}}$ that retains sufficient structural information to guide the policy. We process this state with the first k layers of the video model and pass the resulting activations as conditioning information to the action decoder. The action decoder then performs a full denoising procedure to produce a chunk of robot actions $\mathbf{A}^0_t$.

At inference time, τ_v is a free hyperparameter. Its optimal value is task-dependent, but we show in Sec. V-C empirically that it is generally close to 1 (high noise). In the special case of $\tau_v = 1$, a single forward pass of the computationally intensive video backbone is sufficient to generate a chunk of actions (line 3 in Algorithm 1 becomes redundant), facilitating real-time inference in our experiments. We find that $\tau_v = 1$ is a good default value that balances policy performance and inference

Algorithm 1 Action Sampling(k, τ_v)

Input: $\mathbf{z}^0_{\text{past}}$, $\mathbf{q}_t$, l
$\mathbf{z}^1_{\text{future}}$, $\mathbf{A}^1_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{z}^{\tau_v}_{\text{future}} \leftarrow \mathbf{z}^1_{\text{future}} + \int_1^{\tau_v} v_\phi(\mathbf{z}^0_{\text{past}}, \mathbf{z}^{\tau'_v}_{\text{future}}, l, \tau'_v)\, d\tau'_v$
$\mathbf{h}^{\tau_v} \leftarrow v^{(k)}_\phi(\mathbf{z}^0_{\text{past}}, \mathbf{z}^{\tau_v}_{\text{future}}, l, \tau_v)$
$\mathbf{A}^0_t \leftarrow \mathbf{A}^1_t + \int_1^0 \pi_\theta(\mathbf{A}^{\tau_a}_t, \mathbf{q}_t, \mathbf{h}^{\tau_v}_t, \tau_a, \tau_v) d\tau_a$
return $\mathbf{A}^0_t$

speed. See Sec. D for a discussion on the motivation behind "noisy" video conditioning.

F. Training

Video-Action Model training proceeds in two distinct phases operating on disjoint sets of parameters. The first stage focuses on the video backbone. To align the generalist backbone with the specific visual domain and dynamics of our robotic tasks, we finetune it using Low-Rank Adapters (LoRA) [19] on robotics video datasets. This adaptation step ensures the model captures domain-specific semantics while preserving its pretrained temporal reasoning capabilities.

The second stage focuses on learning the action decoder π_θ while keeping the video backbone frozen. We train the decoder from scratch to regress the action flow field, conditioned on video representations $\mathbf{h}^{\tau_v}$ extracted from the frozen backbone. Crucially, to ensure robustness to varying noise levels during inference, we sample *independent* flow times τ_v (for video) and τ_a (for action) during each training iteration, as detailed in Algorithm 2. We employ a logit-normal distribution for $\mathcal{T}_v$ matching the video pretraining and $\mathcal{T}_a(\tau_a) \propto \sqrt{\tau_a - 0.001}$ for actions following [3]. This decoupled training scheme renders our approach significantly more sample-efficient and faster to converge than comparable VLA baselines (see Sec. V-B).

Algorithm 2 Action Decoder Training(k, $\mathcal{T}_v$, $\mathcal{T}_a$)

repeat
  $\mathbf{z}^{\text{past}}_0, \mathbf{z}^{\text{future}}_0, \mathbf{a}_0, \mathbf{s}_0, l \sim p_0(\mathbf{z}^{\text{past}}_0, \mathbf{z}^{\text{future}}_0, \mathbf{a}_0, \mathbf{s}_0, l)$
  $\tau_v \sim \mathcal{T}_v(\tau_v)$; $\tau_a \sim \mathcal{T}_a(\tau_a)$
  $\boldsymbol{\varepsilon}_v, \boldsymbol{\varepsilon}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{z}^{\text{future}}_{\tau_v} \leftarrow (1-\tau_v)\, \mathbf{z}^{\text{future}}_0 + \tau_v\, \boldsymbol{\varepsilon}_v$
  $\mathbf{a}_{\tau_a} \leftarrow (1-\tau_a)\, \mathbf{a}_0 + \tau_a\, \boldsymbol{\varepsilon}_a$
  $\mathbf{h}_{\tau_v} \leftarrow v^{(k)}_\phi(\mathbf{z}^{\text{past}}_0, \mathbf{z}^{\text{future}}_{\tau_v}, l, \tau_v)$
  Take gradient descent step on
  $\nabla_\theta \left\| \pi_\theta(\mathbf{a}_{\tau_a}, \mathbf{s}_0, \mathbf{h}_{\tau_v}, \tau_a, \tau_v) - u_{\tau_a}(\mathbf{a}_{\tau_a} \mid \mathbf{a}_0) \right\|^2$
until converged

V. Experiments

Our experiments provide an empirical analysis of mimic-video, evaluating the efficacy of leveraging a video backbone for robotic control across several axes:

1) Can mimic-video effectively control multiple embodiments?
2) Does conditioning on a generative video backbone yield superior sample efficiency and faster convergence for action decoder training, compared to conditioning on a VLM backbone?
3) Is fine-grained video reconstruction necessary for effective policy learning?

a) Evaluation setups: We evaluate mimic-video's capabilities across the simulated benchmarks SIMPLER [24] and LIBERO [28], as well as through real-world dexterous manipulation experiments using humanoid hands.

SIMPLER serves as a high-fidelity proxy for real-world performance, evaluating policies trained on the BridgeDataV2 [44] dataset of a Widow-X robot embodiment. By employing system identification and visual matching, it specifically tests the policy's ability to generalize to unseen tasks under realistic visual domain shifts.

LIBERO benchmark evaluates precision and multi-task capacity with a simulated tabletop Panda robot. We focus on the LIBERO-Goal, -Object, and -Spatial suites, each providing 50 expert demonstrations per task across 10 distinct tasks. This widely used benchmark assesses the model's ability to learn precise multi-task manipulation behaviors.

Real-World Dexterous Bimanual Manipulation. To validate mimic-video on high-dimensional, contact-rich tasks, we utilize a bimanual setup equipped with two 16-DoF "mimic" hands [32] mounted on Panda arms. The observation space includes a global workspace view, four wrist cameras, and full proprioception. We evaluate on two long-horizon tasks: Package Sorting (pick, handover, place) and Tape Stowing (pick, stow, move box). Critically, while the video backbone is finetuned on a broader 200-hour corpus, the respective action decoders are trained on extremely scarce task-specific data: just 1h 33m (512 episodes) for sorting and 2h 14m (480 episodes) for stowing.

b) Comparisons: We compare mimic-video's ability to control multiple robots against several state-of-the-art baselines.

$\pi_{0.5}$-style VLA (Knowledge-Insulating). To isolate the effect of video pretraining versus standard vision-language pretraining, we construct a VLM-based baseline following a similar architecture as $\pi_{0.5}$ [21, 10]. We employ the 3B-parameter PaliGemma [2] as backbone and couple it with an action decoder identical to that of mimic-video, creating a 3.42B-parameter VLA. Mirroring mimic-video (and departing from $\pi_{0.5}$), the action decoder cross-attends to one particular layer of the backbone for which we empirically find the optimal choice. This equivalent action decoder design, together with training on perfectly equivalent datasets, ensures that performance differences in our comparisons stem strictly from the quality of the conditioning representations (video vs. image-text). Adhering to the "Knowledge-Insulation" protocol [10], we employ a two-stage training process: the autoregressive backbone is trained via Next Token Prediction (discretizing and compressing actions with FAST [37]), while the action decoder is separately trained via flow matching. Notably,

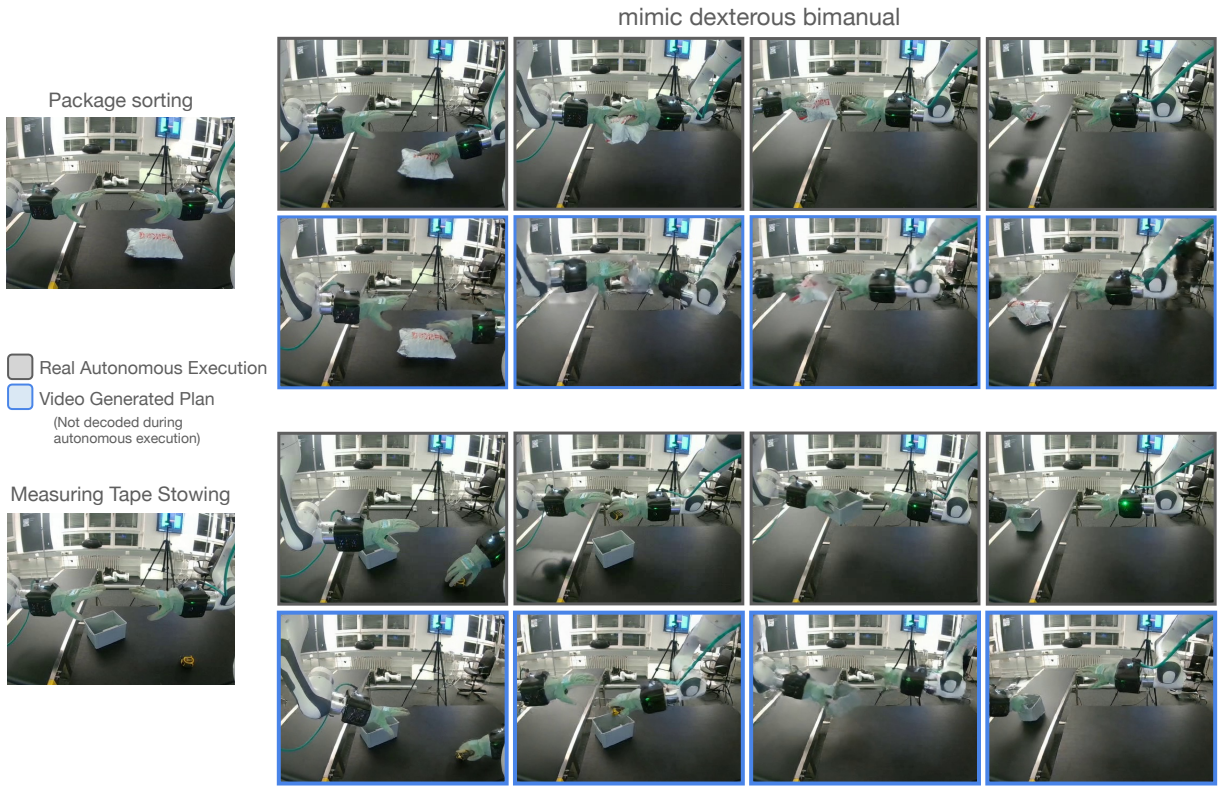

Fig. 4: We train and evaluate mimic-video on a real-world bimanual robot setup with Franka Emika Panda robot arms and mimic 16-DoF dexterous humanoid hands. We execute a real-world evaluation on the bimanual setup with two tasks: package sorting and pick and place of a measuring tape into a box. For each action chunk, mimic-video generates a latent video plan ($\tau_v = 1$) and then executes the actions on the real robot. We further fully denoise the predicted video for this visualization.

while the original $\pi_{0.5}$ leverages massive web and cross-embodiment pretraining, our baseline (denoted "$\pi_{0.5}$-style") trains the backbone from the original VLM checkpoint and the decoder from scratch. Additionally, the original $\pi_{0.5}$ is commonly finetuned using multiple robot camera views while in our experiments, $\pi_{0.5}$-style VLA is restricted to a workspace camera view only. Both changes standardize the data regime across methods and thus allow for a fair evaluation of the backbone prior.

DiT-Block Policy. For real-world bimanual evaluations, we compare against a strong single-task baseline: a DiT-Block Policy [8] following the action representation recipe from Nava et al. [32]. This model features a ViT-S DINO backbone [9, 4] (with separate encoders for each camera view) feeding into an 8-block, 8-head transformer diffusion policy. With approximately 155M parameters (in the multi-view setting), this architecture represents a competitive standard for imitation learning in low-data regimes, making it an ideal reference point for the utilized bimanual dexterous tele-operation datasets.

State-of-the-Art Published Baselines. We additionally include results reported for state-of-the-art competing approaches, namely Octo [43], ThinkAct [20], FLOWER [40], OpenVLA [22] and OpenVLA-OFT [23].

A. Direct Evaluation across Diverse Robot Platforms

a) SIMPLER-Bridge: We first evaluate mimic-video's cross-task generalization capabilities on the SIMPLER-Bridge benchmark, with full results detailed in Table I. Our model achieves the strongest average success rate across all four tasks, matching or surpassing the performance of state-of-the-art baselines, including our $\pi_{0.5}$-style VLA comparison. This strong performance validates that conditioning on the generative video prior yields more robust policy representations than those derived from vision-language-action (VLA) pretraining alone. Additionally, leveraging the partial denoising strategy, we demonstrate a novel form of *inference-time policy optimization*: by adjusting the flow parameter τ_v, the fixed trained model can be specialized to individual task dynamics, achieving further performance gains at the cost of modest increases in computation.

b) LIBERO: We evaluate mimic-video's multi-task manipulation capabilities on the LIBERO benchmark. Despite being trained from scratch on task-specific action data, mimic-video outperforms the majority of state-of-the-art methods finetuned from generalist models (see Table II). Notably, mimic-video achieves significantly higher success rates than the comparable $\pi_{0.5}$-style VLA baseline, indicating that the generative video prior facilitates more robust and efficient policy learning than the corresponding vision-language pretrained representations.

TABLE I: Benchmark scores on SIMPLER-Bridge. The training regimes denote the usage of robot action data: "pretrained" (large-scale external), "finetuned" (external → target), and "scratch" (target only). Note that all models leverage image or video pretraining. **Bold**: best overall; underline: best "scratch" score. We also report mimic-video with task-optimized τ_v.

Inputs: third-person image, language instruction, robot proprioceptive state (optional)					
Model	Put Carrot on Plate	Put Spoon on Towel	Stack Blocks	Eggplant	Average SR (%)
OpenVLA (finetuned) [22]	4.2	8.3	0.0	45.8	14.6
Octo (finetuned) [43]	8.3	12.5	0.0	43.1	16.0
ThinkAct (pretrained) [20]	**37.5**	58.3	8.7	70.8	43.8
FLOWER (finetuned) [40]	13.0	**71.0**	8.0	88.0	45.0
$\pi_{0.5}$-style VLA (scratch)	25.0	29.2	**20.8**	66.7	35.4
mimic-video (scratch)	**37.5**	37.5	12.5	**100.0**	**46.9**
mimic-video (scratch, per task τ_v-tuning)	54.2	41.7	29.2	100.0	56.3

c) Real-World Dexterous Bimanual System: To validate mimic-video on high-dimensional, contact-rich tasks under real-world data scarcity, we benchmark it on a bimanual setup comprising two Franka arms equipped with dexterous humanoid hands. In these experiments, we focus on dexterity rather than generality and train single-task expert models for two distinct tasks: Package Sorting and Measuring Tape Stowing. Fig. 4 illustrates the real world experiment while a detailed description of the tasks and evaluation procedure can be found in Appendix C.

Since no language conditioning is required for these tasks, we train our $\pi_{0.5}$-style VLA baseline without the KI protocol in the final stage of full model finetuning, improving its success rate compared to the knowledge-insulating VLA recipe used in our simulation experiments. We furthermore choose single-task DiT-Block Policies as an additional baseline.

The bimanual setup with dexterous hands presents a significant challenge due to heavy occlusions, particularly during grasping, where wrist camera observations play a critical role in guiding robot policies. This necessity is reflected by the performance gap between the DiT-Block Policy variant privileged with additional wrist camera views and its workspace camera-only counterparts shown in Table III. Remarkably, mimic-video significantly surpasses the performance of all baselines, despite only being conditioned on the single workspace camera view. This result confirms that the predictive capacity of the generative video prior allows mimic-video to effectively bridge the visual uncertainty caused by occlusion, leading to robust policies learned from minimal task-specific data.

TABLE II: Benchmark scores on LIBERO. "finetuned", "scratch", **bold**, and underline are defined as in Tab. I.

Inputs: third-person image, language instruction, proprioception (optional)				
Model	Spatial (%)	Object (%)	Goal (%)	Avg (%)
Diffusion Policy (scratch) [7]	78.3	92.5	68.3	79.7
Octo (finetuned) [43]	78.9	85.7	84.6	83.1
DiT Policy (finetuned) [8]	84.2	96.3	85.4	88.6
OpenVLA (finetuned) [22]	84.7	88.4	79.2	84.1
OpenVLA-OFT (finetuned) [23]	**96.2**	**98.3**	**96.2**	**96.9**
$\pi_{0.5}$-style VLA (scratch)	79.2	94.0	84.4	85.9
mimic-video (scratch)	94.2	96.8	90.6	93.9

TABLE III: Benchmark scores on real-world bimanual dexterous manipulation.

Model	Packing	Package handover
DiT-Block Policy [8]	11.0	30.0
$\pi_{0.5}$-style VLA (non-KI)	22.2	60.7
DiT-Block Policy [8] (+ wrist cams)	42.6	74.1
mimic-video	**72.2**	**92.6**

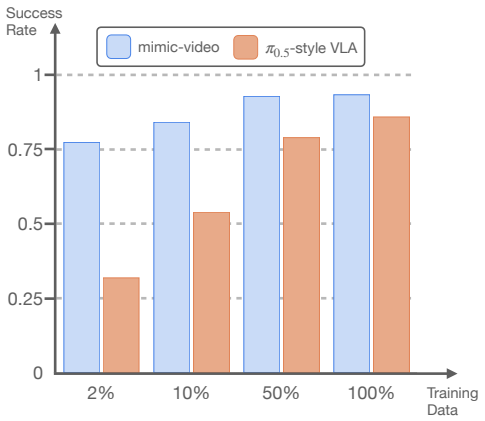

Fig. 5: Sample efficiency for action decoder training on LIBERO: mimic-video against the $\pi_{0.5}$-style VLA baseline. Both decoders are trained with their respective optimal batch size and learning rate and until their respective success rates reach their peaks.

B. Data Efficiency and Convergence Speed

We investigate the data efficiency of decoding actions from video model representations compared to the VLM representations by training mimic-video and $\pi_{0.5}$-style VLA action decoders on differently-sized subsets of the LIBERO-Goal, LIBERO-Spatial, and LIBERO-Object task suites. The result, shown in Fig. 5, demonstrates a remarkable *order-of-magnitude increase in sample efficiency* when conditioning on the video prior. Specifically, mimic-video's action decoder reaches the maximum success rate achieved by the VLM-conditioned decoder while requiring only 10% of the training data. Decreasing the dataset size to only one episode per task (a 98% reduction in action data), still yields a 77% average

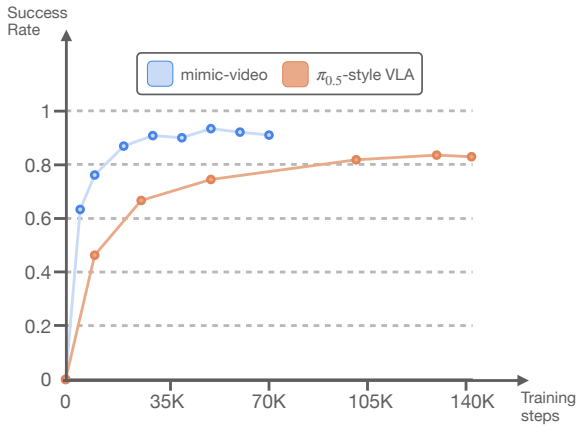

Fig. 6: Convergence speed for action decoder training. Both decoders are trained with a batch size of 128 (optimal for $\pi_{0.5}$-style VLA) and their respective optimal learning rate.

success rate, placing mimic-video trained on 2% of the action data competitive with our Diffusion Policy baseline.

Beyond sample efficiency, Fig. 6 shows that the mimic-video action decoder converges in significantly fewer gradient updates and to a higher asymptotic success rate than the $\pi_{0.5}$-style VLA decoder. Notably, this advantage persists despite the VLA baseline having been exposed to task-specific action data during FAST-pretraining.

The improved sample efficiency and optimization dynamics of mimic-video compared to the VLA baseline stem from a deliberate compute-for-data trade-off: by investing more compute per gradient step through a richer video conditioning signal, the model requires significantly less robot data to reach the same performance. As such, mimic-video will spend 3.8 PFLOPs per gradient update when training on LIBERO while $\pi_{0.5}$-style VLA will spend 55 TFLOPs. We argue that this trade-off is favorable in robotics, where expert teleop data is highly scarce while compute is a more scalable resource.

C. Trade-offs between Video Fidelity and Action Performance

mimic-video couples two separate flow matching models for video and actions, respectively. A key design element of our approach is the ability to control the video generation process via an inference-time hyperparameter: the video flow time $\tau_v \in [0, 1]$. This parameter dictates the extent to which future video latents are denoised during action sampling. To investigate the necessity of fine-grained video reconstruction for effective policy learning, we first note the intuitive hypothesis: a more resolved, higher-fidelity video signal should correlate with better policy performance. In order to study this question, we sweep τ_v across the SIMPLER-Bridge environments and visualize the resulting success rates in Fig. 7.

Counterintuitively, we find that the best autonomous policy performance in our SIMPLER experiments is achieved at the highest flow time $\tau_v = 1$. Theoretically, as τ_v progresses from 1 (pure noise) to 0 (full reconstruction), the underlying video signal grows, and the mutual information $I(\mathbf{z}^{\text{future}}_{\tau_v}; \mathbf{A}^0)$ between the future video latent and future actions increases. However, consistent with the case study in Sec. III, we hypothesize that imperfect video generation introduces artifacts. Consequently, fully denoised video latents may diverge from the training distribution, presenting out-of-distribution conditioning to the action decoder. To isolate the effect of these generation errors, we perform an additional sweep of τ_v where we condition the action decoder on "noisy" ground-truth video latents, $\mathbf{z}^{\text{future}}_{\tau_v}$, computed via Eq. 1. We report the resulting action reconstruction MSE on a held-out validation set of BridgeDataV2 in Fig. 8. We observe that the lowest action reconstruction error is achieved at an intermediate flow time of $\tau_v \approx 0.4$, corresponding to the perfect rollout performance observed in our Case Study (Sec. III). Action prediction error

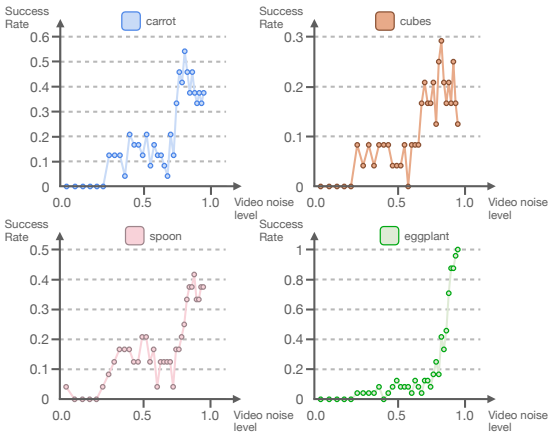

Fig. 7: Policy success rate across the SIMPLER-Bridge environments vs video flow time (τ_v, logit-scaled). Performance peaks at an intermediate noise level, confirming that high-fidelity video reconstruction is not required for performant robot policies.

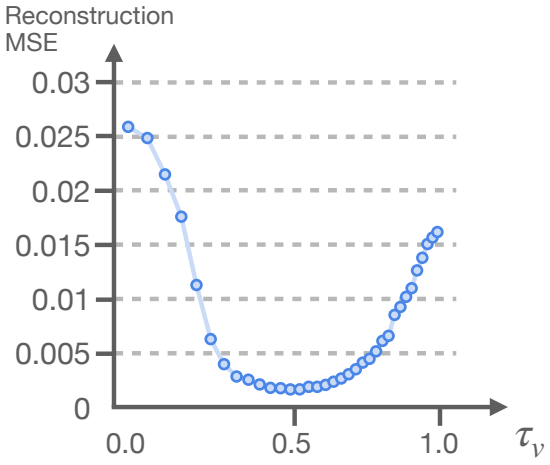

Fig. 8: Action reconstruction MSE of a decoder conditioned on "noisy" ground-truth video latents at varying flow times (logit-scaled) on BridgeDataV2. Reconstruction is best at intermediate flow times and increases towards clean and pure noise latents.

increases sharply as we move from this optimum towards $\tau_v = 0$ (full reconstruction). We provide a detailed discussion of these mechanisms in Appendix D.

This observation yields a significant practical advantage: operating at $\tau_v = 1$ requires only a single forward pass of the video backbone, resulting in both the highest average performance and the fastest inference speed.

TABLE IV: Inference latency of mimic-video at $\tau_v = 1$.

GPU	Inference latency
RTX 4090	1.3 s
H200	470 ms

D. Inference Latency

We measure the inference latency incurred when generating a chunk of actions with mimic-video on different compute hardware in Table IV. Following the results of the previous section, we focus on the case $\tau_v = 1$ where only a single (partial) forward pass of the video model suffices. We find that control at > 2 Hz is feasible using data center-grade GPUs, while even on consumer hardware action generation fits comfortably within less than half of the 3 second action prediction horizon used in our experiments.

VI. Discussion and Future Work

In this work, we introduce mimic-video, a new class of Video-Action Model (VAM) that grounds robotic policies in a pretrained video model. By leveraging the physical priors embedded in internet-scale video, mimic-video achieves an order-of-magnitude improvement in sample efficiency and significantly faster convergence compared to standard VLA baselines. These results strongly suggest that representations learned from large-scale generative video pretraining provide a significantly more robust signal for policy learning than those induced by vision–language-action pretraining. To achieve this, our approach operates by first partially generating a plausible video of a task's successful execution. We find that conditioning on these partially-denoised plans is critical, yielding a dual benefit: it mitigates the distribution shift between model predictions and the ground-truth data used for training, while simultaneously accelerating inference by significantly reducing the computational cost of video generation.

While mimic-video achieves strong performance across both simulated and real-world evaluations, we find that the current model still has several shortcomings. First, we rely on a single-view video backbone, which restricts our policies to a fixed, single workspace view. Exploring a wider range of video architectures, particularly natively multi-view models, would likely enhance spatial reasoning and occlusion robustness. Second, we have not yet applied the VAM recipe to train a unified, large-scale, cross-embodiment model, a step we believe is necessary to unlock the full generalization capabilities of video foundation models. Finally, our current real-world experiments are limited to a focused set of tasks; scaling this approach to a broader diversity of manipulation behaviors remains a key objective for future work.

Acknowledgments

This work was supported under project ID #36 as part of the Swiss AI Initiative, through a grant from the ETH Domain and computational resources provided by the Swiss National Supercomputing Centre (CSCS) under the Alps infrastructure. We thank mimic robotics for providing experimental infrastructure, real-world robot platforms and additional compute resources. Primary work by the lead authors was performed during their internships at mimic robotics, with continued development supported during their internships at Microsoft.

We thank Benjamin Estermann, Erik Bauer, German Rodriguez, Sigmund Hennum Høeg, Irvin Totic, and Benedict Wüest for their help with the project.

References

[1] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning, June 2025. URL http://arxiv.org/abs/2506.09985. arXiv:2506.09985 [cs].

[2] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. pi0: A Vision-Language-Action Flow Model for General Robot Control, November 2024. URL http://arxiv.org/abs/2410.24164. arXiv:2410.24164 [cs].

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *arXiv:2104.14294 [cs]*, May 2021. URL http://arxiv.org/abs/2104.14294. arXiv: 2104.14294.

[5] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019. URL https://arxiv.org/abs/1806.07366.

[6] William Chen, Suneel Belkhale, Suvir Mirchandani, Oier Mees, Danny Driess, Karl Pertsch, and Sergey Levine. Training strategies for efficient embodied reasoning. In *Conference on Robot Learning*, 2025.

[7] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion, March 2024. URL http://arxiv.org/abs/2303.04137. arXiv:2303.04137 [cs].

[8] Sudeep Dasari, Oier Mees, Sebastian Zhao, Mohan Kumar Srirama, and Sergey Levine. The ingredients for robotic diffusion transformers. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, 2025.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. URL http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs].

[10] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.

[11] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation, 2023. URL https://arxiv.org/abs/2302.00111.

[12] 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, October 2023. URL http://arxiv.org/abs/2310.10625. arXiv:2310.10625 [cs].

[13] Chelsea Finn and Sergey Levine. Deep Visual Foresight for Planning Robot Motion, March 2017. URL http://arxiv.org/abs/1610.00696. arXiv:1610.00696 [cs].

[14] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016.

[15] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning Visual Predictive Models of Physics for Playing Billiards, January 2016. URL http://arxiv.org/abs/1511.07404. arXiv:1511.07404 [cs].

[16] Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Haoran Yin, Xiangyu Li, Xinbang Zhang, Ying Zhang, Wenyu Liu, Qian Zhang, and Xinggang Wang. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning, 2025. URL https://arxiv.org/abs/2502.13144.

[17] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation, 2025. URL https://arxiv.org/abs/2510.10125.

[18] Kyle Beltran Hatch, Ashwin Balakrishna, Oier Mees, Suraj Nair, Seohong Park, Blake Wulfe, Masha Itkina, Benjamin Eysenbach, Sergey Levine, Thomas Kollar, and Benjamin Burchfiel. Ghil-glue: Hierarchical control with filtered subgoal images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, 2025.

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[20] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning, 2025. URL https://arxiv.org/abs/2507.16815.

[21] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization, April 2025. URL http://arxiv.org/abs/2504.16054. arXiv:2504.16054 [cs].

[22] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model, September 2024. URL http://arxiv.org/abs/2406.09246. arXiv:2406.09246 [cs].

[23] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

[24] 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, 2024. URL https://arxiv.org/abs/2405.05941.

[25] 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, June 2024. URL http://arxiv.org/abs/2406.16862. arXiv:2406.16862 [cs].

[26] Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video Generators are Robot Policies, August 2025. URL http://arxiv.org/abs/2508.00795. arXiv:2508.00795 [cs].

[27] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023. URL http://arxiv.org/abs/2210.02747. arXiv:2210.02747 [cs].

[28] 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.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

[30] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. Pmlr, 2020.

[31] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):11205–11212, 2022.

[32] Elvis Nava, Victoriano Montesinos, Erik Bauer, Benedek Forrai, Jonas Pai, Stefan Weirich, Stephan-Daniel Gravert, Philipp Wand, Stephan Polinski, Benjamin F. Grewe, and Robert K. Katzschmann. mimic-one: a Scalable Model Recipe for General Purpose Robot Dexterity, June 2025. URL http://arxiv.org/abs/2506.11916. arXiv:2506.11916 [cs].

[33] 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.

[34] NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World Simulation with Video Foundation Models for Physical AI, October 2025. URL http://arxiv.org/abs/2511.00062. arXiv:2511.00062 [cs].

[35] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-Conditional Video Prediction using Deep Networks in Atari Games, December 2015. URL http://arxiv.org/abs/1507.08750. arXiv:1507.08750 [cs].

[36] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

[37] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. In *Proceedings of Robotics: Science and Systems*, Los Angeles, USA, 2025.

[38] Han Qi, Haocheng Yin, Aris Zhu, Yilun Du, and Heng Yang. Strengthening Generative Robot Policies through Predictive World Modeling, May 2025. URL http://arxiv.org/abs/2502.00622. arXiv:2502.00622 [cs].

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

[40] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies, 2025. URL https://arxiv.org/abs/2509.04996.

[41] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686.

[42] Gemini Robotics Team, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas

Kipf, Sean Kirmani, Fangchen Liu, Anirudha Majumdar, Andrew Marmon, Carolina Parada, Yulia Rubanova, Dhruv Shah, Vikas Sindhwani, Jie Tan, Fei Xia, Ted Xiao, Sherry Yang, Wenhao Yu, and Allan Zhou. Evaluating gemini robotics policies in a veo world simulator, 2025. URL https://arxiv.org/abs/2512.10675.

[43] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy, May 2024. URL http://arxiv.org/abs/2405.12213. arXiv:2405.12213 [cs].

[44] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024. URL https://arxiv.org/abs/2308.12952.

[45] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images, November 2015. URL http://arxiv.org/abs/1506.07365. arXiv:1506.07365 [cs].

[46] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[47] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent Action Pretraining from Videos, May 2025. URL http://arxiv.org/abs/2410.11758. arXiv:2410.11758 [cs].

[48] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic Control via Embodied Chain-of-Thought Reasoning, March 2025. URL http://arxiv.org/abs/2407.08693. arXiv:2407.08693 [cs].

[49] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. CoT-VLA: Visual Chain-of-Thought Reasoning for Vision-Language-Action Models, March 2025. URL http://arxiv.org/abs/2503.22020. arXiv:2503.22020 [cs].

[50] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. Flare: Robot learning with implicit world modeling, 2025. URL https://arxiv.org/abs/2505.15659.

[51] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets, May 2025. URL http://arxiv.org/abs/2504.02792. arXiv:2504.02792 [cs].

[52] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, pages 2165–2183. PMLR, December 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html. ISSN: 2640-3498.

CONTRIBUTIONS

Jonas Pai: Led project ideation, implementation, and evaluation. Contributed to tech report writing.
Liam Achenbach: Led baseline model development, training, and evaluation. Helped with dataset integration and tech report writing.
Oliver Sanchez: Contributed real-world experiments for the VLA baseline and performance improvements for the mimic-video implementation.
Stefanos Charalambous: Contributed to development of the mimic bimanual system. Helped with real-world experiments for the VLA baseline and performance improvements for the mimic-video implementation.
Victoriano Montesinos: Contributed the diffusion policy baseline implementation and training, as well as data collection for the bimanual mimic robot experiments.
Benedek Forrai: Oversaw development for the mimic bimanual system, contributed data collection for the bimanual mimic robot experiments.
Oier Mees: Supervised the project from its inception, mentored the lead authors during their research internships, guided the technical direction and experimental strategy, and led the writing and visualization of this tech report.
Elvis Nava: Supervised the project from early conception to implementation and evaluations, contributed to project ideation. Supervised the lead authors throughout their research internship and oversaw the technical development of supporting infrastructure, robot systems and compute resources. Contributed to manuscript writing, website and video editing. Contributed data collection for the bimanual mimic robot experiments.

APPENDIX

A. Training Hyperparameters

We summarize mimic-video training hyperparameters for each dataset in Tab. V.

B. Data Preprocessing

All orientations are expressed as 6-dimensional vectors corresponding to the top two rows of the matrix representation. Images are extracted or rendered in a resolution of 480 x 640 px.

a) BridgeDataV2:

- Observation Space: Absolute end-effector pose and absolute continuous gripper joint state.
- Action Space: Future end effector pose (relative to the proprioceptive pose for the entire action chunk) and the continuous (but practically mostly binary) gripper *action*.

We remove 3046 non-informative language labels as well as the first state and null-action of each episode.

b) LIBERO:

- Observation Space: Absolute end-effector pose and absolute continuous gripper joint state.
- Action Space: End effector pose action (relative to the proprioceptive pose for the entire action chunk) and binary gripper action.

We follow the preprocessing procedure used in Kim et al. [23] and remove episodes not leading to a successful rollout when replaying their actions.

c) Real-World Dexterous Bimanual:

- Observation Space: Absolute end-effector poses, absolute continuous hand joint states. Relative end-effector poses with respect to each other. Previous end-effector and hand actions.
- Action Space: End effector pose action (relative to the proprioceptive pose for the entire action chunk) and absolute hand joints action.

C. Real-World Experiments Evaluation Details

Both real-world manipulation tasks involve pick-and-place like movements involving one (in the package handover task) or two (in the measuring tape packing task) objects. We perform 54 rollouts per task, varying the initial configuration of objects while starting the robot at an invariant home configuration.

a) Package Handover: The robot moves a package from a table to a moving conveyor belt. We vary the initial position of the package over a grid of nine areas covering the table, vary the orientation of the package over three angles offset by 45 degrees, and flip the package both ways showing either a white surface or a red-colored logo.

b) Measuring Tape Packing: The robot moves a measuring tape from a table into a box on the same table, then places the box onto a moving conveyor belt. We vary the initial position of the box over three areas covering the short side of the table adjacent to the conveyor belt, vary the initial position of the measuring tape over a grid of nine areas covering the table, and run each configuration twice.

D. Video Denoising Analysis

A key finding of our work is that the choice of the video model's cutoff flow time, τ_v, is a critical inference hyperparameter. We empirically observe that stopping the video generation process early and conditioning the action decoder on a "noisy" visual plan yields substantially better performance than allowing the video model to fully denoise its prediction. We find two main reasons for this phenomenon:

a) Distribution Mismatch and Noise as Augmentation: The action decoder is trained by conditioning on the video model's representations of *ground truth* future video. A fully denoised video generated by the video model at inference time could represent an incorrect action plan due to the video model's weakness. Even if accurate, it will still likely be subtly out-of-distribution compared to the ground truth data seen during training. By intentionally leaving noise in the visual plan, we perform a kind of train and test-time augmentation. This prevents the action decoder from relying on spurious ground truth visual cues that may not be present in the video model's own generations. This is analogous to findings in goal-conditioned policies [18], where augmenting predicted future target images with simple transformations improved robustness and performance.

TABLE V: Hyperparameters used during training of mimic-video.

Hyperparameter	**Video finetuning**			**Action Decoder training**		
	BridgeDataV2	**LIBERO**	**mimic**	**BridgeDataV2**	**LIBERO**	**mimic**
Learning Rate	1.778e-4			1e-4		
Warmup Steps	1000					
Training Steps	70043	7k-8k	27300	14112	50k	26k
LR Scheduler	Constant			Linear		
Weight Decay Factor	0.1					
Gradient Clip Threshold	10.0					
Batch Size	256	128	32	256	128	128
Optimizer	AdamW [29]					
Video Model Source Layer (k)	19					
Video Observation Horizon	5 frames					
Video Prediction Horizon	56 frames					
Video Frequency	5 Hz					
Proprioception Observation Horizon	—			1 step		
Action Prediction Horizon	—			15 steps	60 steps	45 steps
Action Frequency	—			5 Hz	20 Hz	15 Hz
Proprioception masking ratio	—			0.2		

The results shown in Fig. 2 give credence to this hypothesis, as we observe that conditioning the action decoder on ground truth data at inference time leads to perfect performance, so less-than-perfect performance during regular inference has to be attributed to the fully denoised video plans being imperfect or out of distribution. The results shown in Fig. 7 also illustrate how optimal performance on benchmarks occurs at video flow times close to $\tau_v = 1$, where "noise as augmentation" is larger.

b) Information Content of Intermediate Representations: Another reason lies in the nature of the flow matching model's internal representations throughout the denoising process. In intermediate steps, the hidden states of the video model must encode rich information about scene dynamics and the necessary transformations to reach the final, clean video. However, as the denoising process approaches $\tau_v = 0$, the input is already very close to the target. To minimize the training loss, the video model layers, when conditioned on the final noise values, are incentivized to learn a close-to-identity mapping, making minimal changes to the nearly-perfect input. Consequently, these final-step hidden states become less informative for downstream tasks. Cross-attending to the richer representations from earlier flow times τ_v provides the action decoder with a more useful conditioning signal for generating actions. The results in Fig. 8 indeed distinctly show that, when approaching $\tau_v = 0$, reconstruction error strongly increases, which is compatible with this hypothesis.