IRONMAN: INFORMATION-CONSTRAINED VIDEO-ACTION LEARNING FOR ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video Action Models (VAMs) couple visual dynamics modeling with action generation for robot manipulation. However, video representations are not naturally suited to action generation, as exposing the action policy to excessive visual detail can impair its generalization ability. Therefore, we introduce **IronMan** (**I**nformation-const**R**ained vide**O**-actio**N** learning for robot **MAN**ipulation), a robust video-action learning framework built on information bottleneck principle. The core principle of this framework is to impose information constraints that suppress irrelevant visual information while preserving action-relevant dynamics cues. IronMan employs a dynamics-aware bottleneck that distills noisy, entangled one-step video features into compact world representations. Extensive simulation and real-world experiments demonstrate strong in-distribution (ID) performance and out-of-distribution (OOD) robustness while maintaining efficient inference. IronMan achieves success rates of 99.0% on LIBERO and 79.4% on RoboTwin clean2clean, outperforming all the evaluated baselines. Under OOD shifts, IronMan achieves a success rate of 79.1% on LIBERO-Plus, exceeding the strongest baseline by 10.4 percentage points.

1 INTRODUCTION

Vision-Language-Action (VLA) models have emerged as a prominent class of foundation models for robot manipulation (Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025; Intelligence et al., 2025; 2026; Wu et al., 2026; Liu et al., 2026). However, most VLAs inherit priors from Vision-Language Models (VLMs) pretrained primarily on static image–text data, and therefore lack dynamics modeling. This limitation has motivated a video-action learning paradigm that couples visual dynamics with action generation, enabling these Video-Action Models (VAMs) to learn inverse dynamics within predictive representations (Tian et al., 2025; Hu et al., 2025; Bi et al., 2026; Ye et al., 2026; Li et al., 2026a; Yuan et al., 2026; NVIDIA et al., 2025; Zhang et al., 2026b). These advances reveal the great potential of jointly modeling video dynamics and actions.

However, video and action generation inherently impose different information requirements. A video backbone needs to preserve rich visual details to reconstruct future frames. But an action policy should primarily leverages action-relevant cues. Therefore, the rich information from video backbones is not naturally aligned with the needs of action generation. Existing approaches typically connect video features to the action policy through learned projections or attention mechanisms (Yuan et al., 2026; Hu et al., 2025; Yan et al., 2026). Such interfaces do not explicitly suppress features irrelevant to action generation.

This can undermine the robustness of action policy. When excessive visual cues are exposed to the policy, the model may exploit incidental correlations, which are unreliable under visual distribution shifts. For example, camera noise, light conditions, and background may become shortcuts, leading to degraded generalization in unseen scenes. Therefore, a key challenge is how to transform rich video features into compact representations that capture the dynamics essential for manipulation while avoiding spurious visual correlations.

The information bottleneck (IB) theory offers a principled way to regulate the video-action interface by constraining how much information can pass from input features, while encouraging the latent representation to retain sufficient information for prediction target (Tishby et al., 1999; Alemi et al., 2017). Prior work applies IB-style objectives to regularize visuomotor representations for robot

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

manipulation (Bai et al., 2025; Zhang et al., 2026a; Du et al., 2022). However, these methods do not address the more challenging video-action interface, where features from a generative video backbone entangle predictive dynamics with visual details.

Motivated by this, we introduce **IronMan** (**I**nformation-const**R**ained vide**O**-actio**N** learning for robot **MAN**ipulation), a robust video-action learning framework. As illustrated in Figure 1, unlike inverse dynamics model (IDM)-based VAMs, IronMan extracts one-step video features from the video backbone, substantially reducing inference overhead. Although these features already encode physical dynamics, they remain highly noisy and entangled with redundant visual cues. Applying an information constraint on these features alone does not specify how spatial and temporal cues should be used for action generation. Therefore, we introduce a **dynamics-aware bottleneck** in Figure 2 which first aggregates the video features with a Qformer-style module (Li et al., 2023), and then maps the aggregated features to a parameterized latent distribution through a Gaussian head. These latents constitute world tokens, forming an efficient and compact video-action interface. To instantiate the information bottleneck in our video-action learning framework, we adopt a variational formulation (Alemi et al., 2017) that penalizes the KL divergence between the latent distribution and a standard Gaussian prior, and combine this regularization with video and action supervision to form our joint learning objective.

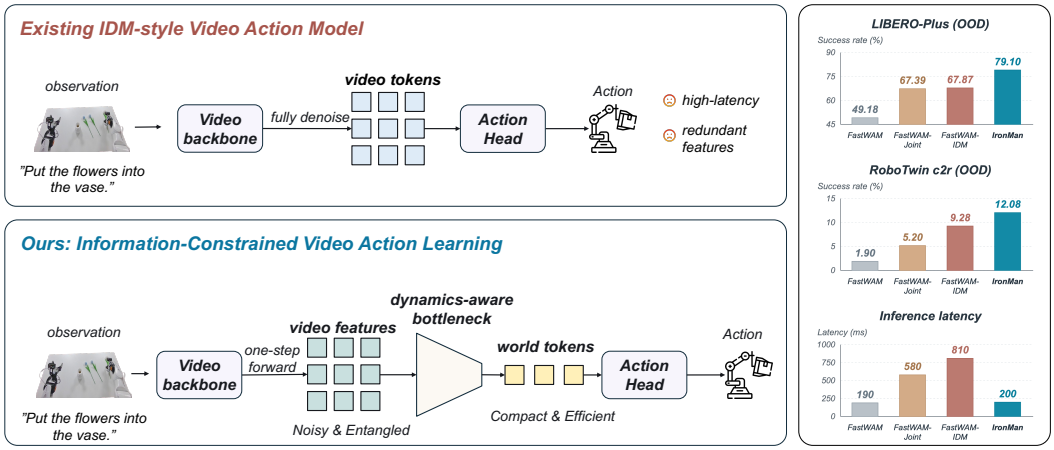


Figure 1: Motivation of IronMan. Existing video–action interfaces allow action policy to freely attend to irrelevant visual signals, increasing its sensitivity to visual details. IronMan instead uses an information-constrained bottleneck designed to retain the predictive dynamics needed for action while suppressing redundancy.

To evaluate the effectiveness of IronMan, we conduct simulation experiments across two datasets and four benchmarks covering both in-distribution (ID) and out-of-distribution (OOD) settings: LIBERO, LIBERO-Plus, RoboTwin clean2clean, and RoboTwin clean2random. We further evaluate three real-world manipulation tasks under ID and OOD settings with background and layout shifts. Across these evaluations, IronMan achieves strong ID performance and OOD robustness while maintaining efficient inference.

2 Related Work

2.1 Vision Language Action Models

Vision–Language–Action (VLA) models, such as RT-2 (Brohan et al., 2023), OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), $\pi_{0.5}$ (Physical Intelligence et al., 2025), and GR00T (NVIDIA et al., 2025), have emerged as a powerful paradigm for robot manipulation by transferring the rich visual and semantic knowledge of pretrained vision–language models (VLMs) to embodied control. Leveraging large-scale pretraining on diverse image–text data, VLAs enable robots to better understand instructions, recognize a wide range of objects and scenes, and generalize across tasks and environments beyond the limited coverage of robotic datasets. This has substantially improved the versatility and generalization of robotic policies, establishing VLAs as a prominent direction toward general-purpose robot manipulation. Despite these achievements, modeling physical dynamics re-

mains a fundamental limitation of current VLAs. Because their VLM backbones are pretrained predominantly on static image–text pairs, they receive little direct supervision about temporal evolution. Recent approaches seek to mitigate this limitation during post-training by aligning VLM representations with features of future observations extracted by vision foundation models, thereby injecting dynamics into a VLM backbone (Zheng et al., 2025b; Su et al., 2026; Xu et al., 2026; Luo et al., 2026; Cen et al., 2025b;a). In contrast, our work directly uses a pretrained video generative model as the source of physical dynamics priors.

2.2 Video Action Models

Recent Video-Action Models (VAMs) increasingly leverage generative video models to provide rich physical dynamics priors (Tian et al., 2025; Hu et al., 2025; Bi et al., 2026; Ye et al., 2026; Li et al., 2026a; Yuan et al., 2026). A paradigm is explicit future prediction, where future frames are generated to guide action generation (Li et al., 2026a; Ye et al., 2026; Bharadhwaj et al., 2025; Liang et al., 2025; Li et al., 2026b; Wang et al., 2026a). However, these methods require full video denoising, introducing substantial inference overhead. This has motivated a more efficient line of work based on implicit future prediction, which bypasses full video denoising and directly extracts one-step features from video backbones (Hu et al., 2025; Ma et al., 2026; Yan et al., 2026; Pai et al., 2026). One-step video features are obtained at the start of denoising process and thus retain redundant visual cues for reconstruction, making them noisy and highly entangled. Existing video-action interfaces allow the action policy to attend to these redundant visual details without explicit information constraints: Ma et al. (2026); Pai et al. (2026) directly connect one-step features to action tokens through cross-attention. Hu et al. (2025); Yan et al. (2026) transform them with learned projection modules. To address this limitation, we introduce an information-constrained interface that explicitly limits the information transferred from video to action.

2.3 Information Bottleneck

The Information Bottleneck (IB) principle provides a framework for learning compact representations that preserve task-relevant information while discarding irrelevant variations (Tishby et al., 1999). Its extensions to deep learning, particularly the Variational Information Bottleneck (VIB) (Alemi et al., 2017), make such information constraints tractable and have been widely adopted to improve representation compactness, invariance, and generalization. In robot manipulation, recent works further apply IB-style objectives to suppress redundant information in visuomotor representations and retain features relevant to action prediction (Bai et al., 2025; Zhang et al., 2026a; Du et al., 2022). Building upon this theoretical foundation, we address a more challenging and specialized problem: learning compact and manipulation-relevant representations from highly noisy, redundant, and entangled video features. Such representations jointly encode rich visual details and dynamics, making it difficult to isolate information useful for control. This problem has remained largely unexplored in prior IB-based methods for robot manipulation.

3 Methodology

As illustrated in Figure 2, we introduce IronMan, an information-constrained video-action learning framework that transforms one-step video features into compact world tokens for action generation. We first introduce the theoretical preliminaries in Section 3.1. We then present the IronMan architecture in Section 3.2. Finally, Section 3.3 describes the joint training objective.

3.1 Preliminaries

Problem Formulation. Formally, given the current observation $\boldsymbol{o}_t$ and language instruction ℓ, the video backbone $p_{\boldsymbol{\theta}_\mathrm{v}}$ parameterized by $\boldsymbol{\theta}_\mathrm{v}$ generates future frames $\boldsymbol{o}_{t+1:t+T_v}$ and video features $\boldsymbol{h}_{t:t+T_v}$. The video-action interface defines a latent distribution $q_{\boldsymbol{\theta}_\mathrm{i}}$ parameterized by $\boldsymbol{\theta}_\mathrm{i}$. During training, a sample from this distribution conditions the action policy $\pi_{\boldsymbol{\theta}_\mathrm{a}}$, parameterized by $\boldsymbol{\theta}_\mathrm{a}$:

$$
\begin{aligned}
&\boldsymbol{o}_{t+1:t+T_v},\, \boldsymbol{h}_{t:t+T_v} \sim p_{\boldsymbol{\theta}_\mathrm{v}}\big(\cdot \mid \boldsymbol{o}_t, \ell\big),\\
&\boldsymbol{z}_t \sim q_{\boldsymbol{\theta}_\mathrm{i}}\big(\cdot \mid \boldsymbol{h}_{t:t+T_v}\big),\\
&\mathbf{a}_{t:t+T_a-1} \sim \pi_{\boldsymbol{\theta}_\mathrm{a}}\big(\cdot \mid \ell, \boldsymbol{z}_t\big).
\end{aligned}
\tag{1}
$$

Here, $\boldsymbol{z}_t$ denotes the latent features used to condition the action policy, and T_v and T_a denote the video and action chunk lengths, respectively. Accordingly, the overall training task jointly models

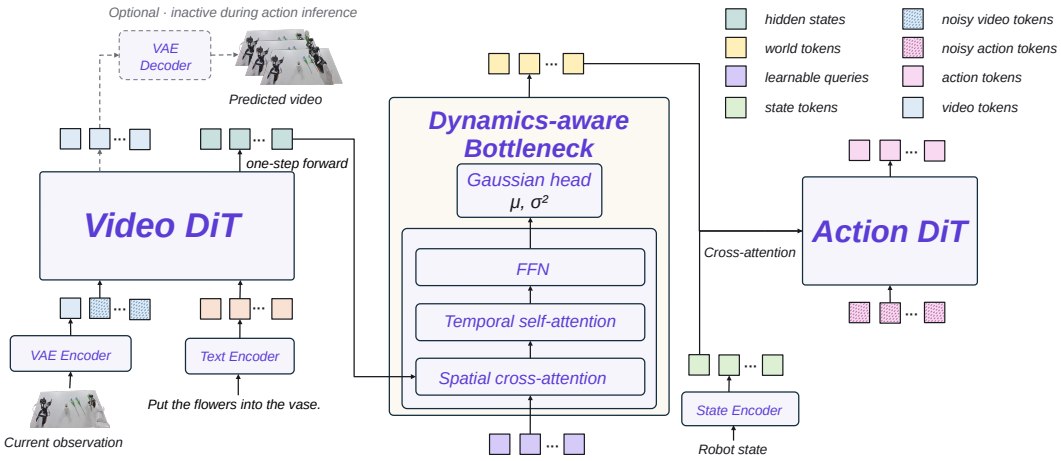


Figure 2: Overview of IronMan. A dynamics-aware bottleneck transforms one-step video features into compact world tokens using a spatio-temporal transformer and a Gaussian head. The action policy attends to world tokens and robot state tokens to generate actions; video decoding is optional and inactive during action inference.

visual dynamics and robot actions as

$$\boldsymbol{o}_{t+1:t+T_v},\, \mathbf{a}_{t:t+T_a-1} \sim p_\Theta\big(\cdot \mid \boldsymbol{o}_t, \ell\big), \tag{2}$$

where $\Theta = \{\boldsymbol{\theta}_\mathrm{v}, \boldsymbol{\theta}_\mathrm{i}, \boldsymbol{\theta}_\mathrm{a}\}$ collects all jointly optimized model parameters, and p_Θ denotes the resulting joint video action model.

Flow Matching. We cast both video and action prediction as flow-matching problems (Lipman et al., 2023). Given a data sample $\boldsymbol{x}_0 \sim p_0$ and independent Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we define

$$\boldsymbol{x}_\tau = (1-\tau)\boldsymbol{x}_0 + \tau\epsilon, \qquad \tau \in [0,1]. \tag{3}$$

We train a neural vector field to predict the velocity $\epsilon - \boldsymbol{x}_0$:

$$\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}_0,\epsilon,\tau}\left[\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_\tau,\tau) - (\epsilon - \boldsymbol{x}_0)\|_2^2\right]. \tag{4}$$

At inference, we integrate the learned vector field backward from Gaussian noise at $\tau = 1$ to 0.

Information Bottleneck. The Information Bottleneck (IB) principle (Tishby et al., 1999) seeks a compact representation $\mathcal{Z}$ of an input $\mathcal{X}$ that preserves information relevant to a target $\mathcal{Y}$ while discarding irrelevant information. Formally, the IB objective is

$$\min_{\mathcal{Z}} \mathrm{IB} = \beta \underbrace{I(\mathcal{X};\mathcal{Z})}_{\text{compression term}} - \underbrace{I(\mathcal{Z};\mathcal{Y})}_{\text{prediction term}}. \tag{5}$$

where $I(\cdot;\cdot)$ denotes mutual information and $\beta > 0$ balances compression and prediction. The compression term limits how much input information $\mathcal{Z}$ retains, while the prediction term encourages $\mathcal{Z}$ to preserve information useful for predicting $\mathcal{Y}$. Together, they favor a compact, target-relevant representation. We apply this principle to video-action learning in Section 3.3.

3.2 IronMan Architecture

As shown in Figure 2, IronMan comprises three key components: a video diffusion transformer (Video DiT) that models visual dynamics; a dynamics-aware bottleneck that compresses the resulting features into compact world tokens under an information constraint; and an action diffusion transformer (Action DiT) that generates robot actions conditioned on these world tokens.

Video DiT. We use Wan2.2-5B (Wan et al., 2025) as our video backbone $p_{\boldsymbol{\theta}_\mathrm{v}}$. The VAE encodes the current observation $\boldsymbol{o}_t$ into conditioning latents, while a text encoder embeds the language instruction ℓ. Conditioned on these inputs, the Video DiT processes noisy future video latents and predicts the velocity field introduced in Section 3.1. To obtain video features, we extract the hidden states $\boldsymbol{h}_{t:t+T_v}$ from layer L_{feat} in a one-step forward process.

Dynamics-aware Bottleneck. The dynamics-aware bottleneck extracts action-relevant dynamics from highly noisy video features then compresses them into compact world tokens. It consists of a spatio-temporal aggregation module and a Gaussian head that parameterizes $q_{\boldsymbol{\theta}_\mathrm{i}}(\boldsymbol{z}_t \mid \boldsymbol{h}_{t:t+T_v})$.

The aggregation module uses learnable queries $\boldsymbol{Q}$ to summarize video features through stacked Qformer-style blocks (Li et al., 2023). Each block performs spatial cross-attention to video tokens, temporal self-attention across query tokens, and a feed-forward update:

$$\boldsymbol{Q}^{(l+1)} = \mathrm{FFN}\Big(\mathrm{SelfAttn}_{\mathrm{temp}}\Big(\mathrm{CrossAttn}_{\mathrm{spatial}}\Big(\boldsymbol{Q}^{(l)}, \boldsymbol{h}_{t:t+T_v}\Big)\Big)\Big), \tag{6}$$

where $\boldsymbol{Q}^{(0)} = \boldsymbol{Q}$. A shared Gaussian head maps the final query tokens $\boldsymbol{Q}^{(L)}$, where L is the number of aggregation blocks, to a mean $\boldsymbol{\mu}_t$ and a diagonal covariance $\boldsymbol{\Sigma}_t$, defining

$$q_{\boldsymbol{\theta}_\mathrm{i}}(\boldsymbol{z}_t \mid \boldsymbol{h}_{t:t+T_v}) = \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t). \tag{7}$$

During training, world tokens are sampled via reparameterization, $\boldsymbol{z}_t = \boldsymbol{\mu}_t + \boldsymbol{\Sigma}_t^{1/2}\epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$; at inference, we use $\boldsymbol{z}_t = \boldsymbol{\mu}_t$.

Action DiT. We instantiate the action policy $\pi_{\boldsymbol{\theta}_\mathrm{a}}$ as a separate diffusion transformer that generates an action chunk of length T_a. Conditioned on the action flow timestep τ_a, it applies self-attention to noisy action tokens and cross-attention to world tokens $\boldsymbol{z}_t$ and robot state embeddings. The instruction ℓ is incorporated through the instruction-conditioned video features encoded in $\boldsymbol{z}_t$. The model predicts an action velocity field introduced in Section 3.1, which is integrated from Gaussian noise at $\tau_a = 1$ to 0 to generate $\mathbf{a}_{t:t+T_a-1}$.

3.3 Training Objective

Action Learning with Variational Information Bottleneck. We instantiate the information bottleneck objective in equation 5 with video features $\boldsymbol{h}_{t:t+T_v}$ as the input, world tokens $\boldsymbol{z}_t$ as the compressed latents, and the action chunk $\mathbf{a}_{t:t+T_a-1}$ as the prediction target. The resulting objective balances retaining action-relevant information against compressing video features. Using a standard Gaussian prior $p(\boldsymbol{z}_t) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the compression term admits the variational upper bound (Alemi et al., 2017)

$$I(\boldsymbol{h}_{t:t+T_v}; \boldsymbol{z}_t) \le \mathbb{E}_{\boldsymbol{h}_{t:t+T_v}}\left[D_{\mathrm{KL}}(q_{\boldsymbol{\theta}_\mathrm{i}}(\boldsymbol{z}_t \mid \boldsymbol{h}_{t:t+T_v}) \,\|\, p(\boldsymbol{z}_t))\right]. \tag{8}$$

Maximizing the prediction term $I(\boldsymbol{z}_t; \mathbf{a}_{t:t+T_a-1})$ corresponds to minimizing conditional action uncertainty. We use the action flow-matching loss as a tractable surrogate for this prediction term, yielding the action-centric objective

$$\mathcal{L}_{\mathrm{ac}}(\boldsymbol{\theta}_\mathrm{i}, \boldsymbol{\theta}_\mathrm{a}) = \mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}_\mathrm{a}) + \beta\, \mathbb{E}_{\boldsymbol{h}_{t:t+T_v}}\left[D_{\mathrm{KL}}(q_{\boldsymbol{\theta}_\mathrm{i}}(\boldsymbol{z}_t \mid \boldsymbol{h}_{t:t+T_v}) \,\|\, p(\boldsymbol{z}_t))\right]. \tag{9}$$

The action-centric loss captures the fundamental trade-off between action prediction and compressing visual-dynamics information.

Information-Constrained Video-Action Learning. To achieve simultaneous modeling of latent representations for both video and action, we complement the action-centered objective with dense visual supervision. We jointly optimize $\boldsymbol{\theta}_\mathrm{v}$, $\boldsymbol{\theta}_\mathrm{i}$, and $\boldsymbol{\theta}_\mathrm{a}$ using

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}_\mathrm{v}) + \mathcal{L}_{\mathrm{ac}}(\boldsymbol{\theta}_\mathrm{i}, \boldsymbol{\theta}_\mathrm{a}). \tag{10}$$

This joint objective trains the video backbone to model rich visual dynamics, while the information bottleneck balances action prediction against visual information compression, encouraging world tokens to preserve control-relevant dynamics and discard details unnecessary for action prediction. The training and inference procedures are detailed in Appendix A.1.1.

4 Simulation Experiments

We evaluate IronMan on RoboTwin Clean dataset (Chen et al., 2025) and the standard LIBERO dataset (Liu et al., 2023), each with an in-domain (ID) benchmark and a corresponding out-of-distribution (OOD) benchmark, to assess its ID performance and OOD robustness. We further conduct ablations to validate individual modules, empirically analyze the bottleneck mechanism in IronMan, and provide visualizations to illustrate the effectiveness of this paradigm.

Table 1: In-domain performance on LIBERO and RoboTwin clean2clean (c2c). Success rates are reported in percentages. Em. PT. denotes embodied pretraining. Bold and underlined values indicate the best and second-best results in each column, respectively.

Models	Em. PT.	LIBERO					RoboTwin c2c
		Spatial	Object	Goal	Long	Avg.	
$\pi_{0.5}$	✓	<u>98.8</u>	98.2	98.0	92.4	96.9	70.7
X-VLA	✓	98.2	98.6	97.8	97.6	98.1	68.0
Abot-M0	✓	<u>98.8</u>	<u>99.8</u>	**99.0**	96.6	<u>98.6</u>	57.4
starVLA	✗	98.7	99.7	<u>98.6</u>	94.2	97.8	46.5
FastWAM	✗	98.2	**100.0**	97.0	95.2	97.6	<u>77.8</u>
LingBot-VA	✓	98.5	99.6	97.2	<u>98.5</u>	98.5	–
Motus	✓	96.8	<u>99.8</u>	96.6	97.6	97.7	–
IronMan	✗	**99.0**	**100.0**	97.8	**99.0**	**99.0**	**79.4**

4.1 SETUP

Datasets. We use two simulation datasets: RoboTwin Clean and LIBERO. RoboTwin Clean comprises 50 manipulation tasks from RoboTwin 2.0, with 50 demonstrations per task collected under the `demo_clean` setting. Standard LIBERO datasetcomprises 40 manipulation tasks across four suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. Training follows a multi-task setting, with a single policy trained per dataset.

Benchmarks. We evaluate IronMan on four benchmarks. For RoboTwin, clean2clean (c2c) evaluates policies trained on clean demonstrations under the same `demo_clean` setting as training, while clean2random (c2r) evaluates them under the `demo_randomized` setting, with 100 evaluation episodes per task across all 50 tasks. For LIBERO, we use the standard benchmark for ID evaluation, with 50 evaluation episodes per task across all 40 tasks, and LIBERO-Plus (Fei et al., 2026) for OOD evaluation, with 10,030 evaluation episodes in total. We report the average success rate for each benchmark.

Baselines. We compare IronMan with two categories of baselines: Vision-Language-Action Models (VLAs), including $\pi_{0.5}$ (Physical Intelligence et al., 2025), X-VLA (Zheng et al., 2025a), Abot-M0(Yang et al., 2026), and StarVLA (StarVLA Community, 2026); and Video Action Models (VAMs), including FastWAM, FastWAM-Joint, FastWAM-IDM (Yuan et al., 2026), Motus (Bi et al., 2026), LingBot-VA (Li et al., 2026a), OpenWAM-Joint (Wang et al., 2026b), and DiT4DiT (Ma et al., 2026).

Implementation Details. We initialize the Video DiT from Wan2.2-5B (Wan et al., 2025), which is then finetuned on the two simulation datasets. Our dynamics-aware bottleneck uses a three-layer spatio-temporal transformer with a hidden dimension of 512. Following the architecture design of Yuan et al. (2026), our Action DiT contains approximately 1B parameters, with a hidden dimension of 1024. We set the action horizon T_a to 32 and temporally downsample the video frames by a factor of 4. The latency of our model is measured on a NVIDIA RTX 5090 (24G) GPU. More implementation details are provided in Appendix A.1.2.

4.2 IN-DOMAIN EVALUATION RESULTS

We evaluate IronMan's in-domain performance on LIBERO and RoboTwin clean2clean (c2c) in Table 1. Without embodied pretraining, IronMan achieves the highest success rate on LIBERO. Notably, IronMan achieves 99.0% on LIBERO-Long, highlighting its effectiveness on long-horizon manipulation tasks. On RoboTwin c2c, IronMan achieves 79.4%, exceeding the strongest reported baseline, FastWAM. These results demonstrate IronMan's strong in-domain performances.

4.3 OUT-OF-DISTRIBUTION EVALUATION RESULTS

For a fair comparison, we compare IronMan with VAM baselines without large-scale embodied pretraining and report inference latency without engineering optimizations. As shown in Table 2, IronMan achieves an overall success rate of 79.1% on LIBERO-Plus, surpassing the strongest baseline, DiT4DiT, by 10.4%. It ranks first across all seven perturbation categories, with the largest

gains over the best competing results in sensor noise (+18.3%) and background textures (+10.2%). On RoboTwin c2r, IronMan ranks second with a success rate of 12.1%, trailing OpenWAM-Joint by 1.1% with less than half its inference latency. These results demonstrate IronMan's strong generalization under visual and physical shifts while maintaining inference efficiency.

Table 2: Performance on LIBERO-Plus and RoboTwin clean2random (c2r), and inference latency. Success rates are reported in percentages. Bold and underlined success rates indicate the best and second-best results in each column, respectively.

Models	LIBERO-Plus								RoboTwin c2r	Latency (ms)
	Camera	Robot	Language	Light	Background	Noise	Layout	Overall		
DiT4DiT	<u>59.7</u>	61.1	88.4	90.8	33.7	<u>60.0</u>	<u>83.6</u>	<u>68.7</u>	–	180
FastWAM	16.8	43.7	68.1	77.2	51.5	36.5	59.5	49.2	1.9	190
FastWAM-Joint	41.3	65.5	<u>91.0</u>	86.4	53.8	56.2	79.2	67.4	5.2	580
FastWAM-IDM	37.7	<u>67.2</u>	90.3	<u>92.7</u>	<u>54.2</u>	56.6	79.2	67.9	9.3	810
OpenWAM-Joint	1.2	53.4	71.1	77.3	43.8	18.4	52.2	43.7	**13.2**	429
IronMan	**64.4**	**75.9**	**92.7**	**93.4**	**64.4**	**78.3**	**83.7**	**79.1**	<u>12.1</u>	200

4.4 Ablation Study

Table 3 disentangles the contributions of the dynamics-aware bottleneck (DB) architecture and the information bottleneck (IB) objective. The baseline directly conditions action generation on hidden states extracted from the fixed feature layer L_{feat}. The IB-only variant applies a shared Gaussian head to each video token, with a wider MLP at the same depth to ensure comparable model capacity. The DB-only variant sets $\beta = 0$, removing the information constraint. Applying IB directly to raw video features yields no improvement over the baseline (73.2% vs. 73.4%), suggesting that token-wise compression alone is insufficient to improve OOD performance. DB alone achieves a modest gain to 74.3%, while combining DB with IB raises the overall score to 79.1%. These results support the complementary roles of structured dynamics extraction and explicit information constraints.

Table 3: Ablation study of the dynamics-aware bottleneck (DB) architecture and information bottleneck (IB) objective across seven types of perturbations in LIBERO-Plus. Bold values indicate the best result in each column.

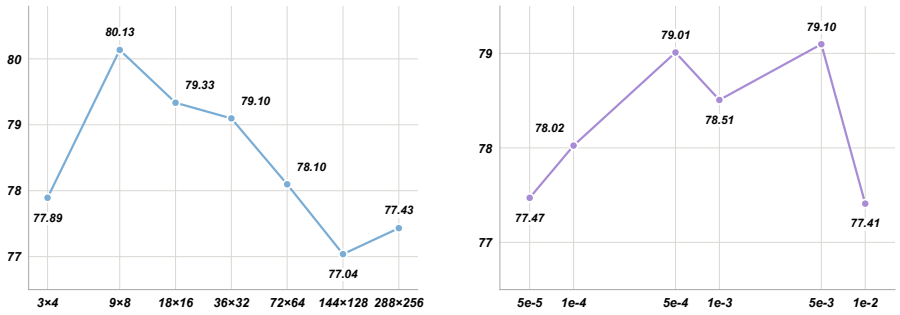
DB arch.	IB obj.	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
✗	✗	49.8	**77.1**	93.2	87.0	63.3	61.3	82.9	73.4
✗	✓	56.0	73.6	90.8	83.1	**65.5**	61.7	82.0	73.2
✓	✗	54.7	75.9	**93.4**	85.2	63.9	64.2	82.8	74.3
✓	✓	**64.4**	75.9	92.7	**93.4**	64.4	**78.3**	**83.7**	**79.1**

4.5 Bottleneck Mechanism Analysis

We examine how bottleneck size ($Q \times D$) and constraint strength (β) affect IronMan's OOD performance, where Q and D denote the number and dimensionality of world tokens, respectively. All configurations outperform the 73.4% baseline in Table 3. Figure 3a shows that smaller bottleneck limits the model's representation capacity, whereas larger bottlenecks may expose the action head to excessive irrelevant visual details. Similarly, Figure 3b shows that intermediate β values perform best, suggesting that weak constraints allow irrelevant information to pass through, while overly strong constraints discard information needed for control. Together, these results suggest a trade-off between retaining sufficient information for action prediction and compressing irrelevant visual details, consistent with the principle of bottleneck mechanism.

4.6 Visualization Analysis

Figure 4 compares attention heatmaps of IronMan and the baseline without the dynamics-aware bottleneck under a background shift from LIBERO to LIBERO Plus. Across the illustrated reach-



(a) Bottleneck size $Q \times D$. (b) Constraint strength β.

Figure 3: Bottleneck mechanism analysis on LIBERO-Plus. Success rates (%) are shown for different bottleneck capacities $Q \times D$ and information bottleneck constraint strengths β.

ing and grasping stages, IronMan focuses more selectively on the gripper–object interaction region, whereas the baseline also attends broadly to task-irrelevant backgrounds. This pattern is consistent with the goal of information-constrained video-action learning: preserving control-relevant information while limiting reliance on irrelevant visual details. Additional visualizations are provided in Appendix A.2, and intervention experiments further assess robustness to visual perturbations in Appendix A.4.1.

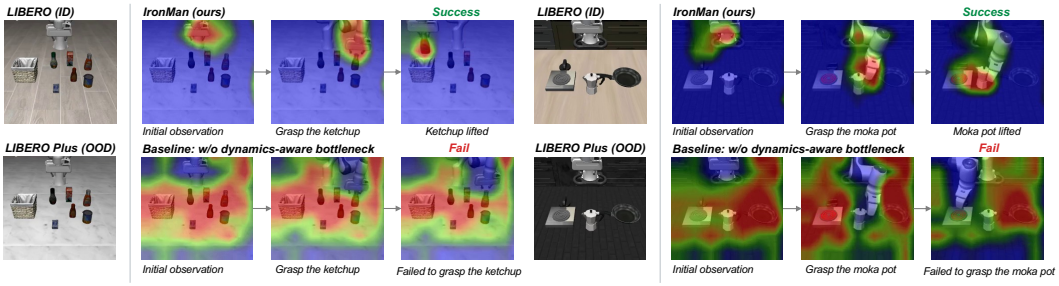


Figure 4: Comparison of attention heatmaps for IronMan and the baseline without the dynamics-aware bottleneck under a background shift from LIBERO (ID) to LIBERO Plus (OOD).

5 Real-World Experiments

Setup. To evaluate IronMan's capabilities on real-world tasks, we design three tasks across the AGIBOT G2 and ROBOTERA M7 platforms: **insert flowers**, **arrange fruits**, and **organize parcels**. These tasks evaluate the model's capabilities in spatial perception, fine-grained manipulation, and long-horizon task execution. For each task, we collect 100 demonstrations via teleoperation. Task demonstrations can be found in Appendix A.3.1.

Baselines. We select $\pi_{0.5}$ (Physical Intelligence et al., 2025) and FastWAM (Yuan et al., 2026) as our baselines. All baselines are trained using their official training codes.

Evaluation Protocol. For each task, we design three evaluation settings: **ID**, where the scene configuration follows the same distribution as the training data; **Background**, where we alter the tabletop texture to introduce strong visual distractions; and **Layout**, where we modify the positions and orientations of objects. For each task under each setting, we conduct 10 trials. Each trial receives a score from 0 to 10 according to the score protocol in Appendix A.3.2.

Results. Figure 5 summarizes the real-world results. Under ID setting, IronMan achieves the highest average normalized score of 91.11, comparable to $\pi_{0.5}$ at 90.28 and exceeding FastWAM by 23.89. Background perturbations substantially degrade FastWAM's performance, particularly on flower insertion task requiring precise manipulation. IronMan achieves an average score of 56.67 and outperforms FastWAM by 50.28, supporting greater robustness under background shift. Under layout shifts, IronMan achieves the highest average score of 66.11, surpassing $\pi_{0.5}$ and FastWAM by

4.72 and 30.28, respectively. These results show that, even without the $\pi_{0.5}$-scale embodied pretraining, IronMan achieves comparable real-world robustness and substantially outperforms FastWAM.

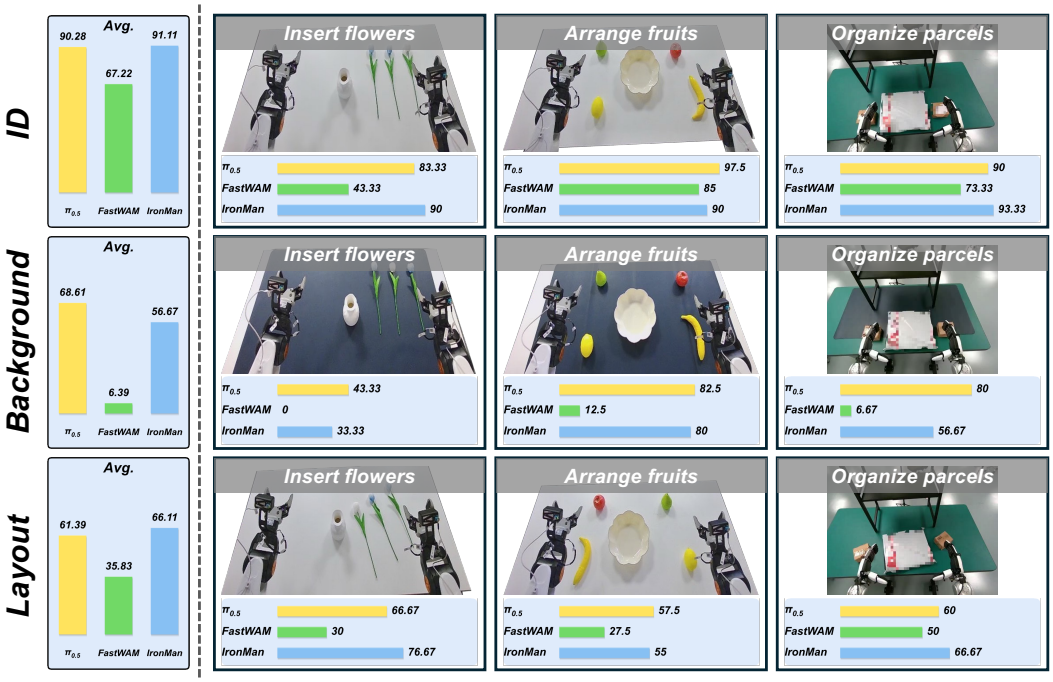


Figure 5: Real-world performance under ID, Background, and Layout settings.

Case Analysis. As shown in Figure 6, IronMan accurately grasps the target flowers and transfers them to the vase. In contrast, FastWAM fails to establish a secure grasp. The result indicates our model maintains more accurate grasping and placement than FastWAM under this background shift in this example.

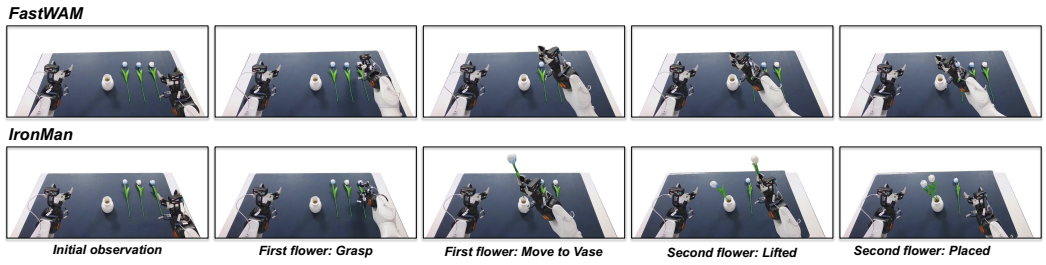


Figure 6: Comparison of the flower insertion task under background shift.

6 Conclusions

We propose IronMan, an information-constrained video-action learning framework that aligns visual representations with robot control. Its dynamics-aware bottleneck combines spatio-temporal aggregation and information regularization to distill one-step video features into compact world tokens. IronMan achieves strong in-domain performance and also demonstrates strong OOD robustness while maintaining efficient inference. Ablations validate the complementary roles of aggregation and regularization, and real-world evaluations further support its robustness.

AI USE STATEMENT

Generative AI tools were used to assist with outlining, and language editing. The authors retain full responsibility for the final manuscript and the associated implementation.

REPRODUCIBILITY STATEMENT

We confirm that the experimental results reported in this paper are reproducible. To facilitate future reproduction and further research, all code and model checkpoints will be publicly released upon acceptance of the paper.

REFERENCES

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Shuanghao Bai, Wanqi Zhou, Pengxiang Ding, Wei Zhao, Donglin Wang, and Badong Chen. Rethinking latent redundancy in behavior cloning: An information bottleneck approach for robot manipulation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 2560–2580. PMLR, 2025.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3936–3951. PMLR, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Kevin Black et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. RynnVLA-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025b.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Yuqing Du, Daniel Ho, Alex Alemi, Eric Jang, and Mohi Khansari. Bayesian imitation learning for end-to-end mobile manipulation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5531–5546. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/du22b.html`.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 2025.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a vla that learns from experience, 2025. URL https://arxiv.org/abs/2511.14759.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

Moo Jin Kim et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026a.

Sizhe Lester Li, Evan Kim, Xingjian Bai, Tong Zhao, Tao Pang, Max Simchowitz, and Vincent Sitzmann. Turning video models into generalist robot policies, 2026b. URL https://arxiv.org/abs/2605.27817.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, 2023.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, Xiao Liu, Dong Ke, Changxun Pan, Chenru Wu, Tailai Cheng, Xiaoshu Ren, Xinlei Zhang, Jianning Cui, Zijie Zhao, Haoyu Zhang, Kaiming Xu, Haodong Yang, Bowen Zhang, Jiahui Niu, Shaoting Zhu, Shiduo Zhang, and Hang Zhao. G0.5: One autoregressive stream for robot reasoning and action, 2026. URL https://arxiv.org/abs/2608.11739.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0.7: A latent world-action model from egocentric videos, 2026. URL https://arxiv.org/abs/2605.00078.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

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.

NVIDIA, Johan Bjorck, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Jonas Pai, Liam Achenbach, Oliver Sanchez, Stefanos Charalambous, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.077.

Physical Intelligence et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026. doi: 10.48550/arXiv.2604.05014. URL https://arxiv.org/abs/2604.05014.

Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. *arXiv preprint arXiv:2602.22010*, 2026. URL https://arxiv.org/abs/2602.22010.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 92033–92052, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e5b5c402bb7bd5e60bede6961d6fe39e-Paper-Conference.pdf.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pp. 368–377, 1999.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Mingxin Wang, Bin Hu, Bin Qian, Kaitao Jiang, Haoning Wu, Feng Yan, Bowen Jing, Ruiyang Hao, Enyi Wang, Kangning Niu, Yandan Yang, Mu Xu, Yan Wang, Houde Liu, and Tianlun Li. St-wam: Semantic-temporal world action model for robust manipulation under visual distribution shifts, 2026a. URL https://arxiv.org/abs/2607.28993.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, Yu-Kai Wang, Yuhan Rui, Shenzhe Yao, Zhen Yuan, Zhenhao Shen, Kefei Zhu, Zijie Zhu, Ning Gao, Xiaowei Chi, Guanqi He, Shanghang Zhang, Hao Dong, Lin Shao, and Hang Zhao. Openwam: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv: 2609.07398*, 2026b.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic VLA foundation model. *arXiv preprint arXiv:2601.18692*, 2026. doi: 10.48550/arXiv.2601.18692. URL https://arxiv.org/abs/2601.18692.

Xiaoxu Xu, Hao Li, Jinhui Ye, Yilun Chen, Jia Zeng, Xinyi Chen, Linning Xu, Dahua Lin, Weixin Li, and Jiangmiao Pang. FutureVLA: Joint visuomotor prediction for vision-language-action model. *arXiv preprint arXiv:2603.10712*, 2026. URL https://arxiv.org/abs/2603.10712.

Haodong Yan, Zhide Zhong, Jiaguan Zhu, Junjie He, Weilin Yuan, Wenxuan Song, Xin Gong, Yingjie Cai, Guanyi Zhao, Xu Yan, Bingbing Liu, Ying-Cong Chen, and Haoang Li. S-VAM: Shortcut video-action model by self-distilling geometric and semantic foresight. *arXiv preprint arXiv:2603.16195*, 2026.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning, 2026. URL https://arxiv.org/abs/2602.11236.

Seonghyeon Ye et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Jinhao Zhang, Wenlong Xia, Yaojia Wang, Zhexuan Zhou, Huizhe Li, Yichen Lai, Haoming Song, Youmin Gong, and Jie Mei. Information filtering via variational regularization for robot manipulation. *arXiv preprint arXiv:2601.21926*, 2026a.

Wenyao Zhang, Bozhou Zhang, Zekun Qi, Wenjun Zeng, Xin Jin, and Li Zhang. Disentangled robot learning via separate forward and inverse dynamics pretraining. *arXiv preprint arXiv:2604.16391*, 2026b.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model, 2025a. URL https://arxiv.org/abs/2510.10274.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loïc 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. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3952–3971. PMLR, 2025b. URL https://proceedings.mlr.press/v305/zheng25a.html.

A Appendix

A.1 Model and Training Details

A.1.1 Training and inference algorithms

Algorithm 1 summarizes the training process, and algorithm 2 describes action inference. We use $\boldsymbol{x}_0$ for clean video latents, $\mathbf{a}_{t:t+T_a-1}$ for the target action chunk, $\boldsymbol{c}_t$ for current-observation latents, and $\boldsymbol{s}_t$ for robot state embeddings. VAE, $\mathrm{Enc}_{\mathrm{text}}$, and $\mathrm{Enc}_{\mathrm{state}}$ denote the corresponding encoders. The video and action velocity predictors are $\boldsymbol{v}_{\boldsymbol{\theta}_\mathrm{v}}$ and $\boldsymbol{v}_{\boldsymbol{\theta}_\mathrm{a}}$. The dynamics-aware bottleneck combines the spatio-temporal aggregation in equation 6 with the Gaussian head in equation 7.

During training, video flow matching uses noisy ground-truth video latents, while a separate one-step forward pass extracts action-conditioning features from Gaussian future latents and the current observation. Both passes share the Video DiT parameters. The latter pass is identical during action inference; world tokens are then computed once and reused throughout action denoising. The conditioning frame is excluded from the video loss. The implementation averages the Gaussian KL over world tokens.

Algorithm 1 IronMan training

Require: Dataset $\mathcal{D}$; parameters $\Theta = \{\boldsymbol{\theta}_\mathrm{v}, \boldsymbol{\theta}_\mathrm{i}, \boldsymbol{\theta}_\mathrm{a}\}$; feature layer L_{feat}; IB weight β

while not converged **do**
  Sample a minibatch $(\boldsymbol{o}_{t:t+T_v}, \ell, \text{robot state}, \mathbf{a}_{t:t+T_a-1})$ from $\mathcal{D}$
  $\boldsymbol{x}_0 \leftarrow \mathrm{VAE}(\boldsymbol{o}_{t:t+T_v})$; $\boldsymbol{c}_t \leftarrow$ first temporal slice of $\boldsymbol{x}_0$
  $\boldsymbol{e}_\ell \leftarrow \mathrm{Enc}_{\mathrm{text}}(\ell)$; $\boldsymbol{s}_t \leftarrow \mathrm{Enc}_{\mathrm{state}}(\text{robot state})$
  Sample video training time τ and $\epsilon_\mathrm{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{x}_\tau \leftarrow (1-\tau)\boldsymbol{x}_0 + \tau\epsilon_\mathrm{v}$; replace its first temporal slice with $\boldsymbol{c}_t$
  $\hat{\boldsymbol{v}}_\mathrm{v} \leftarrow \boldsymbol{v}_{\boldsymbol{\theta}_\mathrm{v}}(\boldsymbol{x}_\tau, \tau \mid \boldsymbol{e}_\ell)$
  $\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}_\mathrm{v}) \leftarrow \|\hat{\boldsymbol{v}}_\mathrm{v} - (\epsilon_\mathrm{v} - \boldsymbol{x}_0)\|_2^2$ on future latents
  Sample Gaussian video latents; replace their first temporal slice with $\boldsymbol{c}_t$
  $\boldsymbol{h}_{t:t+T_v} \leftarrow$ extract hidden states from L_{feat} in a one-step forward on these latents,
  $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \leftarrow \mathrm{DynamicsAwareBottleneck}_{\boldsymbol{\theta}_\mathrm{i}}(\boldsymbol{h}_{t:t+T_v})$
  Sample $\epsilon_\mathrm{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$; $\boldsymbol{z}_t \leftarrow \boldsymbol{\mu}_t + \boldsymbol{\Sigma}_t^{1/2}\epsilon_\mathrm{z}$
  $\mathcal{L}_{\mathrm{KL}} \leftarrow D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \,\|\, \mathcal{N}(\mathbf{0}, \boldsymbol{I}))$
  Sample action training time τ_a and $\epsilon_\mathrm{a} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{a}_{\tau_a} \leftarrow (1-\tau_a)\mathbf{a}_{t:t+T_a-1} + \tau_a\epsilon_\mathrm{a}$
  $\hat{\boldsymbol{v}}_\mathrm{a} \leftarrow \boldsymbol{v}_{\boldsymbol{\theta}_\mathrm{a}}(\boldsymbol{a}_{\tau_a}, \tau_a \mid \boldsymbol{z}_t, \boldsymbol{s}_t)$
  $\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}_\mathrm{a}) \leftarrow \|\hat{\boldsymbol{v}}_\mathrm{a} - (\epsilon_\mathrm{a} - \mathbf{a}_{t:t+T_a-1})\|_2^2$
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}_\mathrm{v}) + \mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}_\mathrm{a}) + \beta\mathcal{L}_{\mathrm{KL}}$
  Update Θ by backpropagating the minibatch mean of $\mathcal{L}$ through both passes
end while
return Θ

Algorithm 2 IronMan action inference

Require: Observation $\boldsymbol{o}_t$; instruction ℓ; robot state; trained Θ; feature layer L_{feat}
Require: Horizons T_v, T_a; action schedule $1 = \tau_a^{(0)} > \cdots > \tau_a^{(N_{\text{infer}})} = 0$
$\boldsymbol{c}_t \leftarrow \text{VAE}(\boldsymbol{o}_t)$; $\boldsymbol{e}_\ell \leftarrow \text{Enc}_{\text{text}}(\ell)$; $\boldsymbol{s}_t \leftarrow \text{Enc}_{\text{state}}(\text{robot state})$
Sample Gaussian video latents for horizon T_v; replace their first temporal slice with $\boldsymbol{c}_t$
$\boldsymbol{h}_{t:t+T_v} \leftarrow$ extract hidden states from L_{feat} in a one-step forward on these latents
$(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \leftarrow \text{DynamicsAwareBottleneck}_{\boldsymbol{\theta}_{\text{i}}}(\boldsymbol{h}_{t:t+T_v})$
$\boldsymbol{z}_t \leftarrow \boldsymbol{\mu}_t$
Sample $\epsilon_{\text{a}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for T_a actions; $\hat{\boldsymbol{a}}^{(0)} \leftarrow \epsilon_{\text{a}}$
for $k = 0, \ldots, N_{\text{infer}} - 1$ **do**
    $\hat{\boldsymbol{v}}_{\text{a}} \leftarrow \boldsymbol{v}_{\boldsymbol{\theta}_{\text{a}}}(\hat{\boldsymbol{a}}^{(k)}, \tau_a^{(k)} \mid \boldsymbol{z}_t, \boldsymbol{s}_t)$
    $\hat{\boldsymbol{a}}^{(k+1)} \leftarrow \hat{\boldsymbol{a}}^{(k)} + (\tau_a^{(k+1)} - \tau_a^{(k)})\hat{\boldsymbol{v}}_{\text{a}}$
end for
return Action chunk $\hat{\mathbf{a}}_{t:t+T_a-1} \leftarrow \hat{\boldsymbol{a}}^{(N_{\text{infer}})}$

A.1.2 Training Configurations

We report the video generation training and the subsequent video-action learning stage separately for each dataset. We first finetune the Wan2.2-5B (Wan et al., 2025) video backbone using the video flow-matching loss on the simulation dataset. Then we train the Video DiT, dynamics-aware bottleneck, and Action DiT together using the objective in equation 10.

LIBERO. We jointly train on the four LIBERO suites. The external and wrist camera images are each resized to 224×224 and concatenated horizontally into a 224×448 video frame. The details are shown in Table 4a.

RoboTwin Clean. We train on the 50-task dataset collected under `demo_clean` setting. The head, left-wrist, and right-wrist camera images are resized to 240×320 before the dataset's multi-camera composition produces a 384×320 video frame. Table 4b gives the full configurations.

Table 4: Training configurations on LIBERO and RoboTwin Clean.

(a) LIBERO

Setting	video generation	video-action learning
Global batch size	128	256
Training steps	10k	30k
Learning rate	10^{-4}	10^{-4}
Video resolution	224×448	224×448
Sampled video frames	9	9
Action chunk length T_a	—	32
Number of queries Q	—	36
World token dimension D	—	32
IB weight β	—	5×10^{-3}
Feature layer L_{feat}	—	17
Bottleneck hidden dimension	—	512

(b) RoboTwin Clean

Setting	video generation	video-action learning
Global batch size	128	256
Training steps	10k	30k
Learning rate	10^{-4}	10^{-4}
Video resolution ($H \times W$)	384×320	384×320
Sampled video frames	9	9
Action chunk length T_a	—	32
Number of queries Q	—	36
World token dimension D	—	32
IB weight β	—	5×10^{-3}
Feature layer L_{feat}	—	17
Bottleneck hidden dimension	—	512

A.2 Visualization Analysis

Figures 7–10 extend the visualization analysis in Figure 4 with four examples involving a cup, a ketchup bottle, a moka pot, and a soup can. These examples cover different object arrangements and background shifts from LIBERO to LIBERO Plus, including changes in surface texture, color, and pattern. Across all four examples, IronMan exhibits more spatially selective attention than the baseline without the dynamics-aware bottleneck. Its attention is initially concentrated around the gripper and shifts toward the gripper–object interaction region as the robot approaches and grasps the target. During lifting, the high-attention region remains localized around the gripper and manipulated object, indicating that the spatial focus evolves with the manipulation rather than remaining fixed on background structures.

In contrast, the baseline exhibits broad attention over surrounding surfaces and non-target objects throughout the illustrated sequences. This difference is particularly visible in the cluttered ketchup-bottle and soup-can examples, where the baseline highlights large regions containing multiple objects and the basket, while IronMan maintains a more localized focus around the ongoing interaction. A similar pattern appears in the cup and moka-pot examples despite their different scene layouts: substantial baseline attention falls on surrounding structures or nearby objects, whereas IronMan emphasizes the region involved in grasping. In all four displayed rollouts, IronMan successfully grasps and lifts the target, while the baseline fails to grasp it. Together, these observations provide additional qualitative evidence consistent with the intended role of the dynamics-aware bottleneck: retaining manipulation-relevant information while reducing sensitivity to task-irrelevant visual variation.

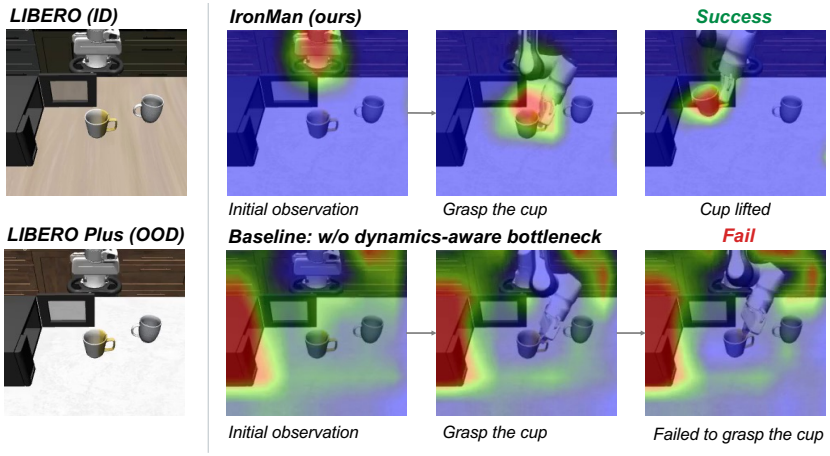


Figure 7: Attention visualization for grasping and lifting the cup.

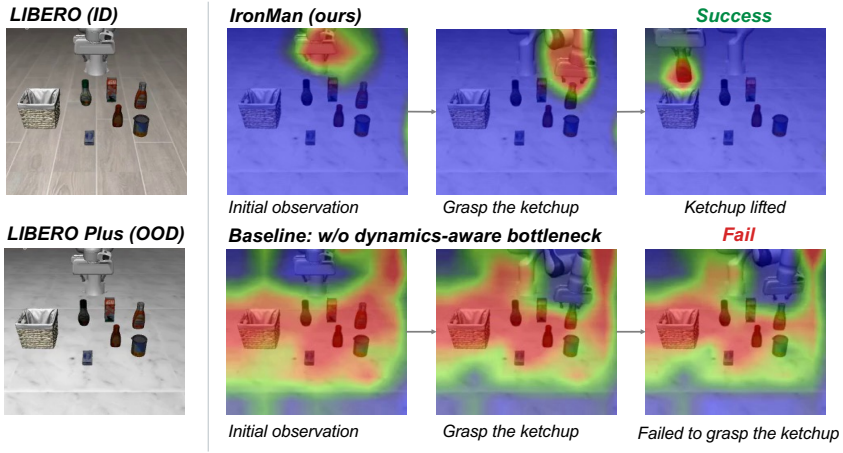


Figure 8: Attention visualization for grasping and lifting the ketchup bottle.

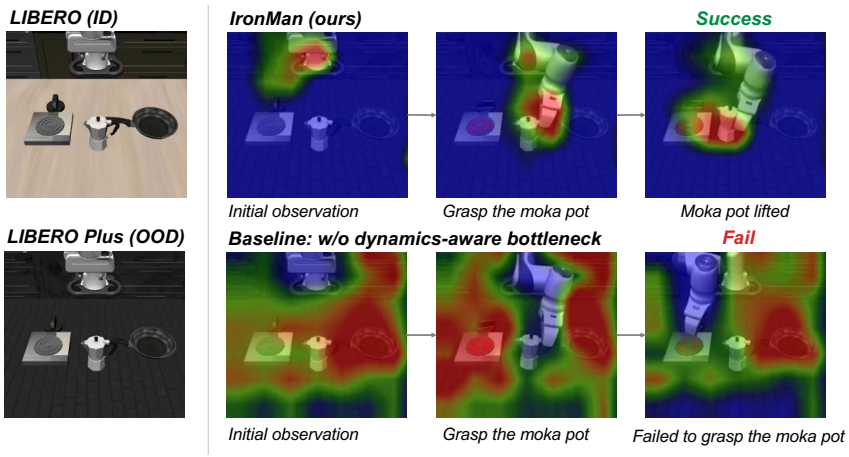


Figure 9: Attention visualization for grasping and lifting the moka pot.

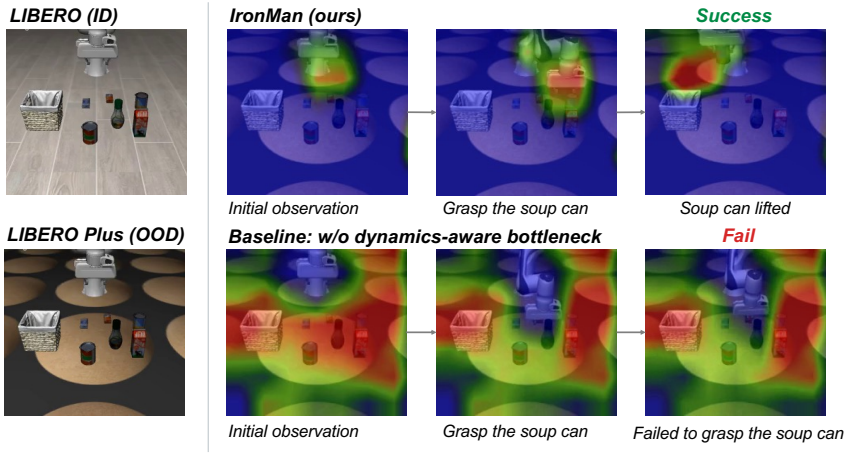


Figure 10: Attention visualization for grasping and lifting the soup can.

A.3 Real-World Experiments Details

A.3.1 Task Demonstrations

As shown in Figure 11, we evaluate our method on three real-world robotic manipulation tasks across two robbot platforms. The first two tasks use an AGIBOT G2 robot equipped with Omini Picker grippers, while the third uses a ROBOTERA M7 robot equipped with XHAND 1 dexterous hands. Both robots are equipped with three cameras: one head-mounted camera and two wrist-mounted cameras, one on each wrist. These cameras provide complementary views of the workspace and the regions around the end effectors. In Task 1, the robot must pick up three flowers and place them into a vase, requiring it to grasp each flower and align it with the vase opening before insertion. In Task 2, the robot must pick up fruits from the table and place them onto a plate, requiring object grasping and controlled placement within the target area. In Task 3, the robot must grasp parcels and place them on a shelf, requiring stable grasping during transport and accurate placement on the shelf.

Each task is evaluated under three settings: in-distribution (ID), Background, and Layout. The ID setting follows the training distribution and measures performance under familiar scene conditions. The Background setting introduces an out-of-distribution (OOD) shift by changing the tabletop background, testing robustness to changes in visual appearance. The Layout setting introduces a second OOD shift by changing the initial positions and orientations of the objects, testing generalization to unfamiliar spatial arrangements. These two OOD settings evaluate the effects of visual background changes and object layout changes, respectively.

We conduct 10 trials for each task under each setting, yielding 30 trials per task and 90 trials across the three tasks. Each trial is scored out of 10 points according to the task-specific scoring criteria detailed below.

A.3.2 SCORE PROTOCOL

Table 5 shows the score protocol. We score each trial according to the highest sequential completion stage achieved. The flower task follows the order of the first, second, and third flowers. The fruit task follows the order of banana, apple, lemon, and pear. The parcel task requires placing the first small parcel on the second shelf level, followed by the second small parcel on the same level, and finally the large parcel on the first shelf level. Each score level requires completion of all preceding stages. The same scoring criteria apply to the ID, Background, and Layout settings. We conduct 10 trials for each task under each setting and report normalized task scores on a scale from 0 to 100.

Table 5: Sequential scoring criteria for the three real-world tasks. Entries are raw per-trial scores on a 0–10 scale; reported task scores are normalized to a 0–100 scale.

Task	Score	Completion criterion
insert flowers	0	The first flower is not placed in the vase.
	3.33	The first flower is placed in the vase.
	6.67	The second flower is placed in the vase.
	10	The third flower is placed in the vase.
arrange fruits	0	The banana is not placed on the plate.
	2.5	The banana is placed on the plate.
	5	The apple is placed on the plate.
	7.5	The lemon is placed on the plate.
	10	The pear is placed on the plate.
organize parcels	0	No parcel is placed on shelf level 2.
	3.33	The right small parcel is placed on shelf level 2.
	6.67	The left small parcel is placed on shelf level 2.
	10	The large parcel is placed on shelf level 1.

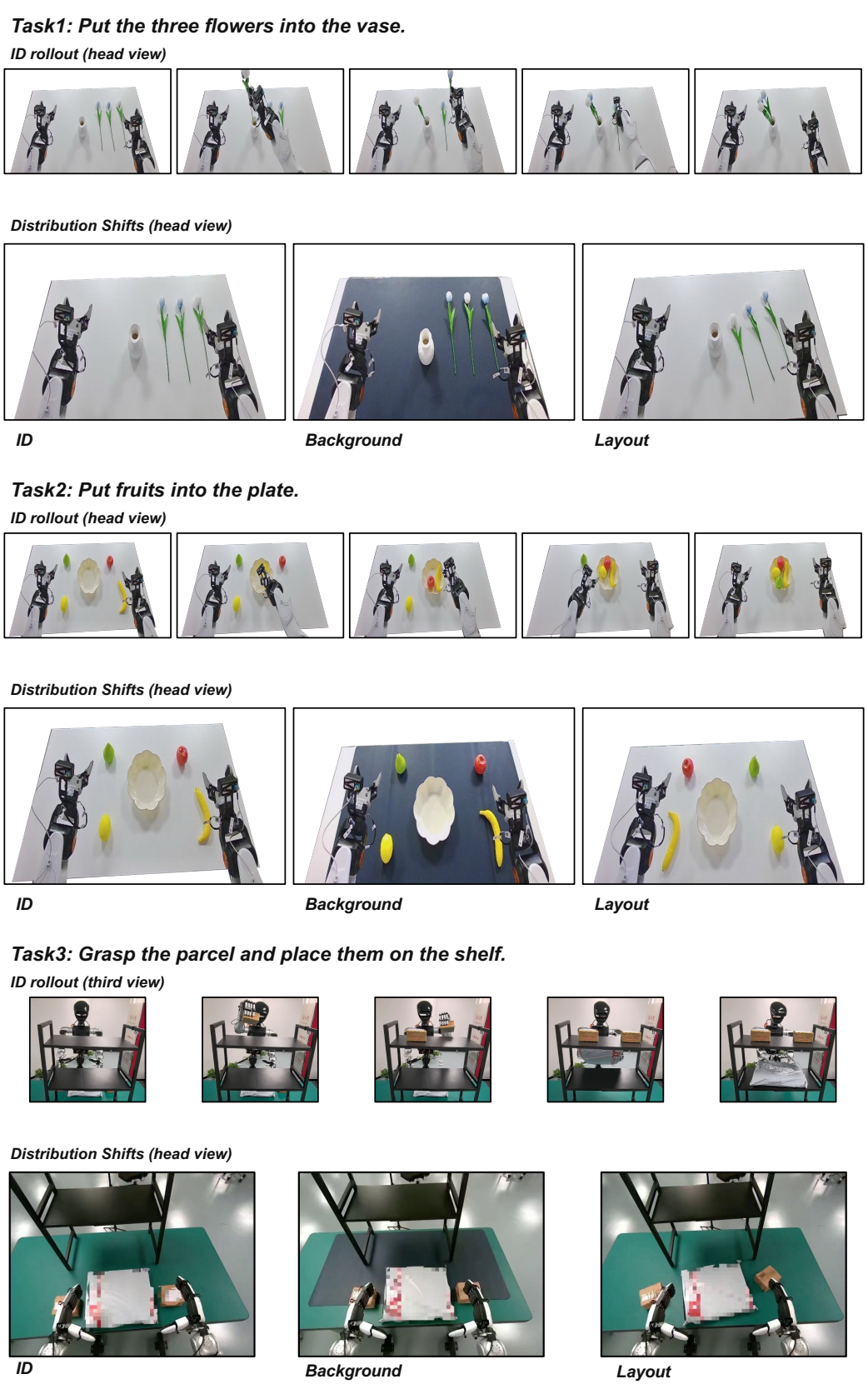


Figure 11: Real-world task demonstrations.

A.4 Additional Experiments

A.4.1 Intervention Experiments

We evaluate both models' sensitivity to visual perturbations on one task from each of four LIBERO suites—Spatial, Object, Goal, and LIBERO-Long—with a fixed initial state per task. Each model generates and executes a 32-step action chunk open-loop under clean conditions, yielding eight reference trajectories. For each task, we sample 10 configurations without replacement from each of four LIBERO-plus perturbation categories—Camera Viewpoints, Light Conditions, Background Textures, and Sensor Noise—using a fixed random seed, yielding 320 perturbed trajectories across both models. All rollouts use the same task-specific physical initial state, instruction, initial proprioceptive input, and physical parameters; both models receive identical images for corresponding conditions, with inference seeds fixed. We measure the mean absolute difference (L1 error) relative to the same model's clean reference for the same task.

We show the results in Figure 12. Across all four visual perturbation categories, IronMan exhibits lower median action L1 deviations and narrower interquartile ranges than the baseline. The reductions are particularly pronounced under changes in camera viewpoint, lighting, and sensor noise, while background texture perturbations yield a smaller reduction in the median. These results suggest that IronMan reduces both typical action deviations and their variability across the evaluated task–configuration pairs. Overall, IronMan improves action consistency under visual perturbations.

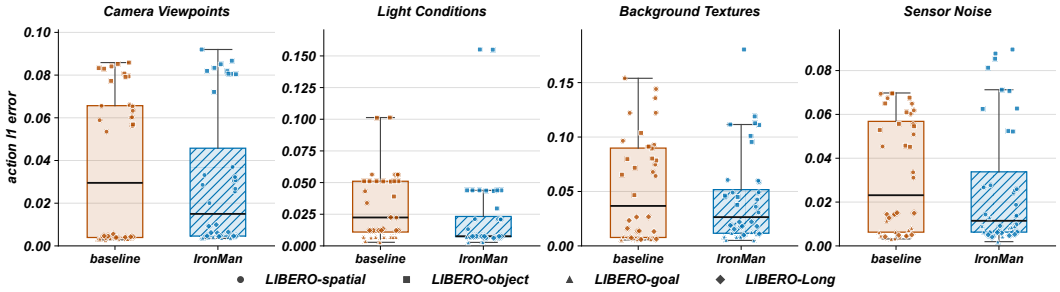


Figure 12: Action prediction error to visual perturbations. Box plots show the L1 error relative to each model's clean reference under four categories of visual perturbation, with 40 values per category and model across four LIBERO tasks and 10 perturbation configurations per task.