ENFOLD: FOLDING WORLD MODEL IMAGINATION INTO PREDICTIVE REPRESENTATIONS FOR ULTRA-EFFICIENT EMBODIED CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

World generative models are typically used through their outputs: a rendered future, a video-conditioned action, or latent context from a costly generative branch. We argue that their more reusable asset is the computation that constructs a future. As a generator transforms a corrupted future into a coherent trajectory, its intermediate states organize appearance, spatial layout, and interaction across abstraction levels. Can this computation be internalized in a representation inferred from the present alone? We present Enfold, which transfers it into a representation predicted from the current visual context and language instruction. During training, multi-level states exposed while the generator processes an observed future supervise a current-only encoder. The representation conditions future generation and is read by task heads without allowing task gradients to reshape the encoder. At deployment, action prediction bypasses the generator. This asymmetry encourages the encoder to retain transition structure predictable from the present rather than stochastic, future-dependent nuisance variation. Across LIBERO, RoboTwin2.0, and real-robot tasks, Enfold supports strong control and cuts action latency by $3.7\times$ relative to Fast–WAM; Enfold-Flash reaches $10.1\times$. Analyses show that it suppresses nuisance variation and preferentially captures changes emerging over longer horizons. When a human alters the current scene, both the generated continuation and executed actions adapt, inconsistent with fixed trajectory replay. These results recast a world generator as a source of predictive control representations: its future need not be materialized at every step if its internal structure can be enfolded into the present.

1 INTRODUCTION

Effective control requires more than recognizing the current scene: a robot must anticipate how the scene will change as an interaction unfolds. World generative models provide a compelling substrate for this capability. Trained to synthesize future observations, they capture long-range motion, object interaction, and multiple plausible continuations in observation space (Agarwal et al., 2025; 2026). This progress has led to robotic systems that imagine a visual future before acting or jointly model videos and actions (Du et al., 2024; Lv et al., 2025; Bi et al., 2026; Kim et al., 2026).

Figure 1 summarizes the dominant output-level interfaces. Imagine-then-act systems materialize a visual future before recovering an action, while unified world-action models couple video and action prediction in one decoding process. Recent work has weakened this dependence on rendered video: Fast–WAM retains video co-training while skipping explicit future generation at test time (Yuan et al., 2026), predictive policies and asynchronous WAMs consume internal video-model features or reusable latent context (Hu et al., 2024; Cai et al., 2026), and ImageWAM reads image-editing caches without decoding the edited frame (Zhang et al., 2026c). In parallel, VLA–JEPA predicts future target-encoder features from a current-only pathway (Sun et al., 2026). These results show that predictive information need not require a rendered future, but not whether the computation that constructs one can be transferred to a current-only representation. We therefore ask: *can a world generator's internal computation be internalized to support both control and future generation?*

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.

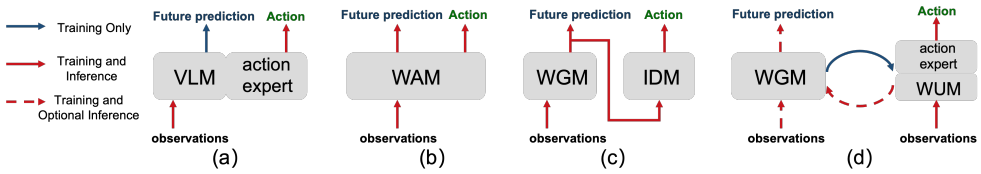


Figure 1: **Different interfaces between future modeling and action prediction.** (a) VLA methods augment a vision–language representation with future prediction while an action expert produces controls, (b) unified WAMs jointly predict future observations and actions, (c) imagine-then-act methods generate a visual future with a world generative model (WGM) and recover actions through inverse dynamics (IDM), and (d) Enfold instead internalizes the WGM's future-generative dynamics in a world understanding model (WUM), whose representation conditions both generation and a downstream action expert. This representation-level interface avoids requiring generated video rollout for action prediction.

The internal states of a video generator are a promising but nontrivial answer. As a corrupted future becomes a coherent trajectory, successive blocks organize appearance, spatial layout, and interaction at different scales. These states offer structured supervision beyond pixels, but depend on a future unavailable at deployment, vary with generation noise, and change across depth and corruption. Useful representations must retain transition structure predictable from current context.

We call the resulting framework **Enfold**: it enfolds future-conditioned generative dynamics into a current-only representation from which the model can unfold a visual future or directly read out an action. Enfold couples a predictive encoder and video generator in two directions. In *generation-to-representation* (G2R), multi-level features exposed as the generator processes the observed future supervise an encoder that sees only the current visual context and instruction. In *representation-to-generation* (R2G), the same representation conditions future generation. Detached task heads read it without sending task gradients into the encoder. The generator is therefore a training-time supervisor and optional future decoder, not a mandatory part of the control loop. At deployment, action prediction executes only the encoder and action head.

A central choice is which generator states supervise the encoder. No layer is universally best: the most informative block shifts with corruption level. Deeper states increasingly suppress appearance variation and emphasize global layout, but can also become more sensitive to generation noise. Enfold therefore predicts a multi-level target through a timestep-conditioned head. Unlike pixel prediction or distillation from a frozen visual encoder, this target describes how a generative model organizes the observed transition across both depth and corruption.

This is not merely a cheaper interface. Without embodied pre-training, Enfold obtains 97.8% success on LIBERO and 91.77% on RoboTwin2.0; Enfold-Flash reaches 97.5% and 92.02%, respectively. At 134 ms per action chunk, Enfold is $3.7\times$ faster than Fast–WAM (Yuan et al., 2026); Enfold-Flash reaches 49 ms and is $10.1\times$ faster. Generator-state supervision outperforms both future-pixel and action-only targets, while feeding the learned representation back to the generator improves future-video prediction. Representation probes further show that it suppresses nuisance variation present in the generator features and becomes increasingly predictive of scene regions that change over longer horizons. Finally, when a human changes the scene while the goal and preceding history remain fixed, the imagined continuation and executed actions both adapt. This coordinated response is inconsistent with trajectory replay and qualitatively supports adaptation to an alternative present.

Our contributions are:

- We formulate multi-level generator computation as predictive supervision and characterize its utility and stability across depth and corruption.
- We develop Enfold, distilling generator computation into a current-only representation shared by future generation and detached task readouts.
- We show competitive simulated and real-world control without executing the generator in the action path, improved future prediction, and much lower action latency.

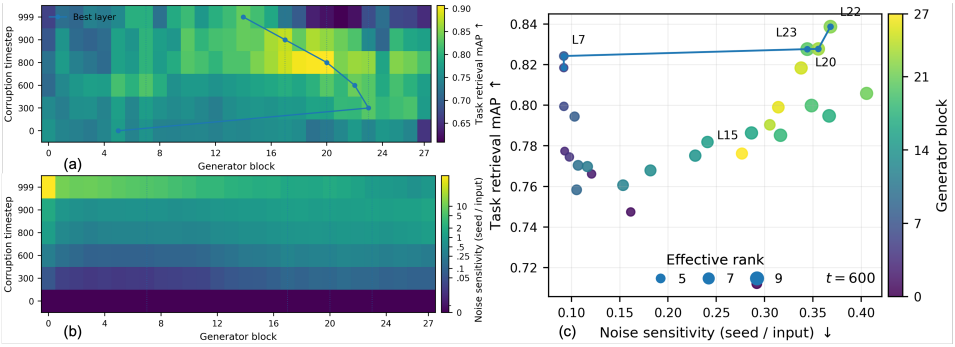


Figure 2: **Generator-state utility and stability. (a)** Task/scene retrieval across blocks and corruption timesteps; the trajectory marks the best block and dotted lines the selected layers. **(b)** Noise sensitivity relative to input variation. **(c)** Utility–stability at $t = 600$; marker size denotes effective rank and the curve marks the Pareto frontier.

Related work. Figure 1 contrasts Enfold with future-aware VLA models, generative world-action models, and predictive representation learning; Appendix A gives details. Enfold transfers multi-level states from a teacher-forced generator into a current-only representation, rather than retaining a generative control path or predicting only a future encoder's endpoint features.

2 Motivation: Why Generator States?

Transferring generative computation into a current-only representation requires choosing what the encoder predicts. Actions are executable targets but reveal scene evolution only indirectly. Future pixels provide dense temporal supervision but preserve visual detail potentially irrelevant to control. Frozen future-encoder features compactly describe an endpoint but not the generative process.

Generator states offer a different supervision space. During a teacher-forced pass, the generator processes an observed future across corruption levels and network depths. Intermediate states expose how appearance, spatial configuration, and interaction are organized into a coherent trajectory. They are grounded in the realized transition without weighting every output pixel equally. Predicting these states from current context yields future-oriented structure without control-time generator execution.

These states are not ready-made targets. They depend on the realized future and sampled noise, and their content changes across the generator. A useful target must therefore be informative, predictable enough from the present, and non-degenerate. We examine this trade-off before defining Enfold.

Let c denote the observed video context, e the language instruction, y the realized future video chunk, and y_t its corrupted latent at timestep t. For this diagnostic, we write the generator before introducing Enfold's representation conditioning. A teacher-forced forward pass exposes the hierarchy

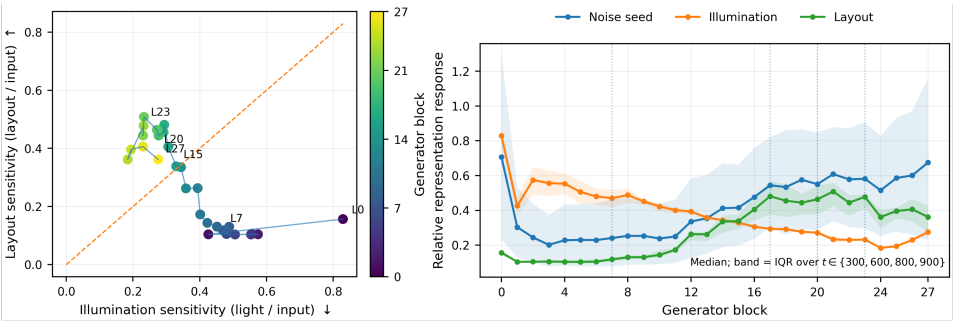
$$\mathcal{H}_\theta(y_t, t, c, e) = \left\{h^G_{\ell,t}\right\}_{\ell \in \mathcal{L}}. \tag{1}$$

We evaluate task and scene retrieval, effective rank, sensitivity to sampled noise, and controlled responses to illumination and spatial layout. Together, these diagnostics measure utility, stability, selectivity, and feature diversity. Appendix D provides the complete protocol.

Figure 2 shows that no block dominates throughout corruption. The most discriminative block changes with t, and depths occupy different points on the utility–stability frontier. Some late states remain informative yet noise-sensitive; semantic utility alone cannot select a supervision layer.

Depth also changes what is preserved. Figure 3 shows that deeper states suppress illumination yet retain task layout, but couple more to stochastic generation. Thus selectivity and predictability are not jointly monotonic; levels are complementary.

These observations motivate a target constructed from several generator depths $A\left(\mathcal{H}_\theta(y_t, t, c, e)\right)$, where $A(\cdot)$ selects and concatenates the chosen states. Concatenation preserves complementary levels, while a timestep-conditioned head accounts for how their structure is expressed at each t. The following section incorporates these choices into the G2R and R2G objectives.



(a) (b)

Figure 3: **Depth-dependent selective invariance.** **(a)** Illumination and layout sensitivity across blocks, summarized over $t \in \{300, 600, 800, 900\}$. **(b)** Noise, illumination, and layout sensitivity across depth; bands denote the interquartile range and dotted lines the selected layers.

3 METHOD

3.1 OVERVIEW

Enfold contains a predictive encoder U_ϕ, a generative world model G_θ, and lightweight prediction and task heads. For a current video context c and language instruction e, the encoder produces

$$z = U_\phi(c, e), \qquad \bar{z} = \mathrm{sg}(z), \tag{2}$$

where sg denotes stop-gradient. The representation complements the visual context. G2R determines which future-conditioned dynamics are enfolded into z; R2G tests whether they can be unfolded into a future. Task heads evaluate the same representation without sending gradients into U_ϕ. The paths share forward computation but are separated in optimization.

3.2 GENERATOR-TO-REPRESENTATION LEARNING

Let y_0 be the latent encoding of the real future chunk and $\epsilon \sim \mathcal{N}(0, I)$. At timestep t, we construct $y_t = \alpha_t y_0 + \sigma_t \epsilon$. While processing y_t under the current context, instruction, and detached representation, the generator produces

$$\mathcal{H}_\theta\big(y_t, t, c, e, \bar{z}\big) = \big\{h^G_{\ell,t}\big\}_{\ell \in \mathcal{L}}, \qquad r^G_t = \mathrm{LN}\left[A\left(\mathcal{H}_\theta\big(y_t, t, c, e, \bar{z}\big)\right)\right], \tag{3}$$

where $\mathcal{L}$ contains the selected layers and $A(\cdot)$ concatenates their spatio-temporal tokens along the channel dimension. Layer normalization places the combined target on a stable scale.

This target is extracted from a teacher-forced forward pass over the real future, not from a sampled video rollout. It captures how the generator organizes the observed transition at the sampled corruption level. Because z itself does not depend on t, a timestep-conditioned head maps the shared representation to the generator's timestep-specific coordinate system:

$$\hat{r}^G_t = F_\omega\big(z, \gamma(t)\big), \qquad \mathcal{L}_{\mathrm{G2R}} = \mathrm{SmoothL1}\left(\hat{r}^G_t, \mathrm{sg}(r^G_t)\right), \tag{4}$$

where $\gamma(t)$ is the timestep embedding. The same t corrupts the future, indexes the teacher features, and conditions F_ω. Gradients through the target are stopped. Thus U_ϕ must predict future-dependent structure from the present, while F_ω absorbs variation specific to the generative trajectory.

3.3 REPRESENTATION-TO-GENERATION LEARNING

We instantiate G_θ as a conditional latent video generator and write the objective using flow matching (Lipman et al., 2022). The generator predicts the velocity field of the corrupted future conditioned on c, e, and $\bar{z}$:

$$\mathcal{L}_{\mathrm{R2G}} = \mathbb{E}_{c,e,y_0,\epsilon,t}\left[\left\|v_\theta\big(y_t, t, c, e, \bar{z}\big) - u_t(y_0, \epsilon)\right\|_2^2\right], \tag{5}$$

where u_t is the target velocity of the interpolation path. Detaching z ensures that this objective teaches the generator to use the representation without allowing the generation loss to shape the encoder directly. R2G is therefore also a functional test: if z contains predictive information beyond the visible context, conditioning on it should improve future modeling.

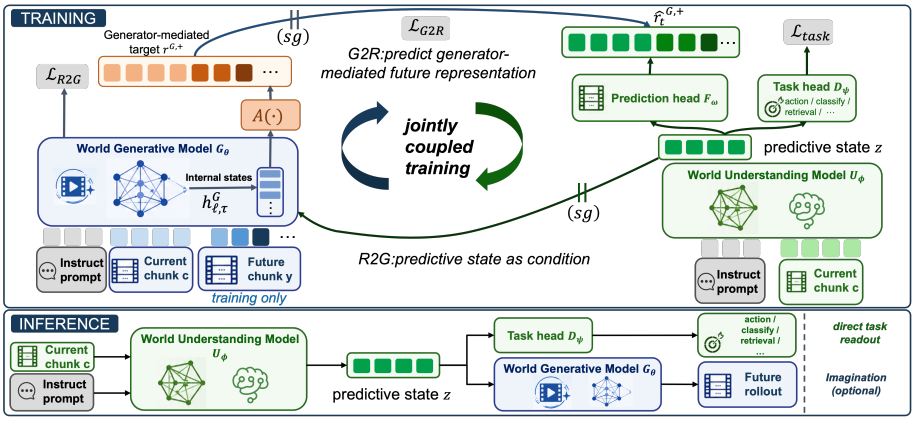


Figure 4: **Overview of Enfold.** The predictive encoder U_ϕ maps the observed context and instruction to z. In G2R, the generator G_θ processes the corrupted real future. Selected hidden states form a multi-level target predicted from z by the timestep-conditioned head F_ω. In R2G, a detached copy of z conditions future generation. Task heads also read a detached z, so task supervision does not directly shape the encoder. At control time, only U_ϕ and the action head are executed. G_θ is used only when an explicit video future is requested.

3.4 Detached Task Readouts

Given a task target q, a readout D_ψ is optimized from the detached representation:

$$\hat{q} = D_\psi\big(\mathrm{sg}(z)\big), \qquad \mathcal{L}_{\mathrm{task}} = \ell_{\mathrm{task}}(\hat{q}, q). \tag{6}$$

For embodied tasks, the action head is implemented as a conditional flow-matching model: it learns an action velocity field conditioned on $\mathrm{sg}(z)$ and generates action chunks by transporting Gaussian noise toward the demonstrated action distribution. Detachment makes downstream performance a readout of the information exposed by generator-mediated learning, rather than the result of action gradients adapting U_ϕ into a policy backbone.

3.5 Optimization and Inference

The complete objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{G2R}} + \lambda_{\mathrm{R2G}}\mathcal{L}_{\mathrm{R2G}} + \lambda_{\mathrm{task}}\mathcal{L}_{\mathrm{task}}, \tag{7}$$

with the following gradient routes:

$$\nabla_{\phi,\omega}\mathcal{L} = \nabla_{\phi,\omega}\mathcal{L}_{\mathrm{G2R}}, \quad \nabla_\theta\mathcal{L} = \lambda_{\mathrm{R2G}}\nabla_\theta\mathcal{L}_{\mathrm{R2G}}, \quad \nabla_\psi\mathcal{L} = \lambda_{\mathrm{task}}\nabla_\psi\mathcal{L}_{\mathrm{task}}. \tag{8}$$

The generator is updated only by future modeling, the predictive encoder and prediction head only by G2R, and each task head only by its readout loss.

At inference, U_ϕ is evaluated once to obtain z. Future imagination uses G_θ to sample from $p_\theta(y \mid c, e, z)$. Control instead generates an action chunk directly from z and never executes the video generator. This representation-level separation provides the algorithmic latency reduction. Enfold-Flash preserves the same model and inference path and adds TensorRT operator-level acceleration.

4 Experiments

The evaluation follows the information flow in Figure 4 and asks four questions. First, can the predictive representation support accurate control without executing the video generator, and what latency does this interface achieve? Second, does it adapt to an intervened scene, or merely replay a trajectory associated with the preceding context? Third, do the G2R and R2G paths contribute in their intended directions? Finally, what information does generator-mediated supervision add beyond a frozen visual representation? We address these questions in order, moving from task-level utility and behavior to mechanism and representation.

Table 1: Comparison on LIBERO. "P.T." denotes embodied pre-training, size is the total parameter count, and latency is measured per action prediction on the same NVIDIA A100 40GB GPU. Enfold-Flash uses TensorRT (NVIDIA, 2026). Best and second-best distinct values are **bold** and underlined.

Method	P.T.	Model Size	Latency↓ (ms)	Long	Goal	Object	Spatial	Average
MAINSTREAM VISION–LANGUAGE–ACTION MODELS								
OpenVLA–OFT (Kim et al., 2025)	✓	7B	—	94.5	97.9	98.4	97.6	97.1
π_0 (Black et al., 2024a)	✓	3.5B	184	88.4	94.4	96.8	98.0	94.4
$\pi_{0.5}$ (Intelligence et al., 2025)	✓	3.5B	184	92.4	98.0	98.2	**98.8**	96.9
GR00T–N1.6 (NVIDIA GEAR Team, 2025)	✓	3.3B	232	94.4	97.5	98.5	97.7	97.0
LATENT ACTION-BASED METHODS								
LAPA (Ye et al., 2025)	✓	7B	—	55.4	58.8	74.6	73.8	65.7
UniVLA (Bu et al., 2025)	✓	7B	—	92.0	95.6	96.8	96.5	95.2
Mantis (Yang et al., 2026b)	✓	5.8B	—	94.2	94.4	99.2	**98.8**	96.7
VLA–JEPA (Sun et al., 2026)	✗	3B	—	95.8	97.2	99.6	96.2	97.2
WORLD ACTION MODELS								
F1 (Lv et al., 2025)	✗	4B	352	91.3	95.4	97.8	98.2	95.7
Motus (Bi et al., 2026)	✗	8B	2759	97.6	96.6	99.8	96.8	97.7
Cosmos–Policy (Kim et al., 2026)	✗	2.1B	1133	97.6	**98.2**	**100.0**	98.1	**98.5**
LingBot–VA (Li et al., 2026)	✗	5.5B	3812	**98.5**	97.2	99.6	98.5	**98.5**
Fast–WAM (Yuan et al., 2026)	✗	6B	493	95.2	97.0	**100.0**	98.2	97.6
Enfold (Ours)	✗	3B	134	97.4	96.8	**100.0**	97.0	97.8
Enfold-Flash (Ours)	✗	3B	**49**	96.6	96.6	99.8	97.0	97.5

4.1 EXPERIMENTAL SETUP

We evaluate control on the 40 language-conditioned tasks drawn from LIBERO (Liu et al., 2023), the 50 dual-arm tasks of RoboTwin2.0 (Chen et al., 2025b), and four real-world bimanual tasks. RoboTwin2.0 is tested in both clean and randomized scenes. Enfold instantiates the generator with Cosmos-Predict 2.5 2B (Agarwal et al., 2025) and the predictive encoder with DINOv3 ViT-H+/16 (Siméoni et al., 2025). Simulation control is measured by task success rate, and real-robot evaluation uses a normalized completion score over ordered task stages. Future prediction uses PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). Representation quality is studied with frozen probes and direct feature comparisons. Training, data processing, action-head configurations, and all evaluation protocols are detailed in Appendix C.

4.2 EFFICIENT CONTROL WITHOUT VIDEO ROLLOUT

We first test the intended deployment mode: the action head reads z directly, and the video generator is absent from the control path. The central comparison is therefore not accuracy alone, but the trade-off between task success and the cost of exposing future-oriented information to the policy.

LIBERO. Table 1 reports the four LIBERO suites. Without embodied pre-training, Enfold averages 97.8%, including 100.0% on Object and 97.4% on Long. It exceeds Fast–WAM by 0.2 points on average and 2.2 on Long while using half as many parameters and predicting an action in 134 ms—a 3.7× speedup under the same A100 40GB protocol. The gains are modest on a saturated benchmark, but show that removing generative rollout preserves control quality.

Enfold-Flash reduces latency to 49 ms, 10.1× faster than Fast–WAM, with only a 0.3-point success reduction. The variants separate two efficiency sources: standard Enfold reaches 134 ms by omitting future-video rollout from control, while Enfold-Flash adds operator-level acceleration.

Table 2: RoboTwin2.0 success rates (%).

Method	P.T.	Clean	Rand.	Avg.
π_0 (Black et al., 2024a)	✓	65.92	58.40	62.16
$\pi_{0.5}$ (Intelligence et al., 2025)	✓	82.74	76.76	79.75
ABot–M0 (Yang et al., 2026a)	✗	81.20	80.40	80.80
Motus (Bi et al., 2026)	✓	88.66	87.02	87.80
LingBot–VA (Li et al., 2026)	✓	**92.90**	91.50	**92.20**
Fast–WAM (Yuan et al., 2026)	✗	91.88	91.78	91.83
Enfold (Ours)	✗	91.60	91.94	91.77
Enfold-Flash (Ours)	✗	91.96	**92.08**	92.02

RoboTwin2.0. RoboTwin2.0 tests dual-arm manipulation in clean and randomized settings. Enfold scores 91.60% clean and 91.94% randomized, averaging 91.77% (Table 2). Enfold-Flash scores 91.96% and 92.08%, averaging 92.02%: 0.19 points above Fast–WAM and 0.18 below LingBot–VA, despite no embodied pre-training. These near-ties do not support ranking without



Figure 5: **Real-robot tasks and distribution shifts.** Representative ID (top) and OOD (bottom) executions; each OOD row changes one factor while keeping the instruction and completion criterion fixed.

uncertainty estimates.The relevant result is that TensorRT acceleration preserves control quality: Enfold-Flash slightly improves over Enfold, and both retain randomized performance. Thus the interface transfers from single- to dual-arm manipulation without control-time video rollout.

Real-world bimanual manipulation. We evaluate the same representation-driven policy on four AgileX dual-arm tasks. The primary evaluation measures in-distribution task completion. A second split introduces controlled changes to object placement and scene appearance. This separates basic task execution from robustness to deviations not present in the demonstrations. Figure 5 shows the four tasks and their matched ID/OOD conditions.

Table 3: **Real-robot normalized completion score (%).** ID and OOD denote in- and out-of-distribution evaluation; the stage-based scoring protocol is detailed in Appendix C.

Method	Fold Towel	Desktop	Spoon	Store Plate	ID Avg.	OOD Avg.
Motus (Bi et al., 2026)	50.0	66.7	33.3	53.3	50.8	38.0
$\pi_{0.5}$ (Intelligence et al., 2025)	**93.3**	84.4	85.6	81.1	86.1	**83.3**
Fast–WAM (Yuan et al., 2026)	73.3	**88.9**	72.2	76.7	77.8	70.0
Enfold	92.2	**88.9**	**87.7**	**90.0**	**89.7**	76.6
Enfold-Flash	84.4	82.2	84.4	88.9	85.0	73.3

Enfold reaches an average normalized completion score of 89.7% in distribution, 11.9 points above Fast–WAM and 3.6 points above $\pi_{0.5}$. The improvement over Fast–WAM is not confined to one task: it is 18.9 points on *Fold Towel*, 15.5 on *Spoon Powder*, and 13.3 on *Store Plate*, while the two methods tie on *Organize Desktop*. This pattern is consistent with the predictive representation helping most when actions must track deformable geometry or an evolving object configuration.

Under distribution shift, $\pi_{0.5}$ leads at 83.3%. Enfold ranks second at 76.6%, 6.6 points above Fast–WAM. Enfold-Flash scores 85.0% ID and 73.3% OOD, gains of 7.2 and 3.3 points over Fast–WAM. Thus Enfold's advantage extends beyond the demonstrations, while the 6.7-point gap to $\pi_{0.5}$ identifies larger shifts as a remaining challenge.

4.3 Reimagining and Recovery after Intervention

Aggregate success does not reveal whether Enfold tracks the current scene or replays the preceding trajectory. We intervene on *Store Plate* and *Fold Towel* at t_2, changing the object configuration while keeping the instruction and prior interaction fixed. Enfold re-encodes the scene into z^+. Replay should retain the invalid continuation; context sensitivity should revise both imagination and action.

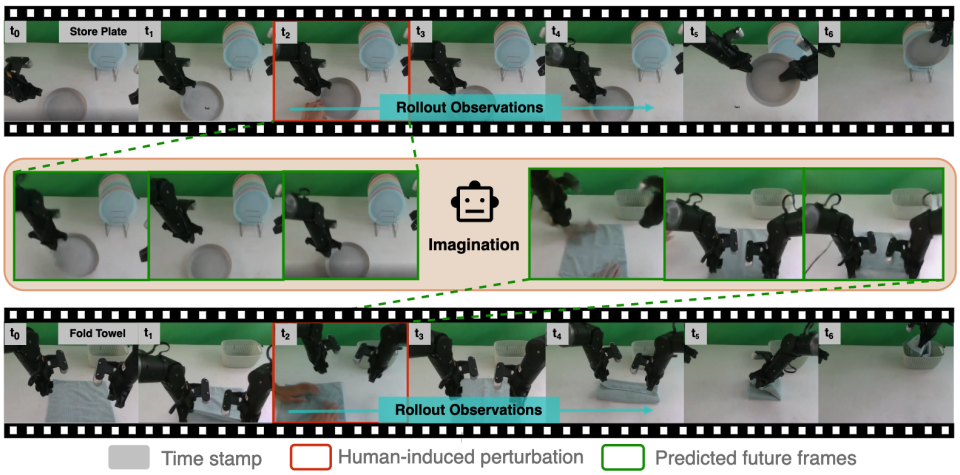


Figure 6: **Reimagining and recovery under human intervention.** Red marks the intervention at t_2. Green shows future frames decoded from the updated representation for *Store Plate* (left) and *Fold Towel* (right). The subsequent rollouts adapt to the modified scene and complete the original instruction. The generator is used only for visualization. Actions are predicted directly from the representation.

In *Store Plate*, imagination and action redirect toward the displaced plate; in *Fold Towel*, they adapt to changed cloth geometry and re-establish contact. Their coordinated departure from the pre-intervention trajectory argues against fixed replay and qualitatively supports consistency under an alternative present. Because the perturbation is directly observed, this establishes intervention-conditioned replanning, not full causal counterfactual inference.

4.4 Dissecting the Bidirectional Coupling

The main results establish that z supports control without video rollout, but not that the generator-mediated objective is responsible. We test each direction: G2R should make generator states better supervision than pixels or actions, while R2G should make the representation useful beyond the action head, including to its supervising generator.

G2R: generator states as supervision. We replace the generator target with future-pixel prediction or action-only training, keeping the downstream protocol fixed. Table 4 separates generator-state supervision from combining depths. A single-level target reaches 96.3%, versus 96.0% for future pixels and 94.9% for actions. Multi-level supervision reaches 97.8%, another 1.5 points. It beats a single level by 2.8 points on Goal and LIBERO-10 but only 0.2 on Spatial and Object, supporting complementary depths for goal-directed or multi-stage tasks. Because the downstream protocol is fixed, these differences isolate supervision; future pixels lead Spatial by 1.0 point, consistent with appearance and geometry sufficing for local rearrangement.

Table 4: Effect of representation supervision on LIBERO (%).

Supervision	Spa.	Obj.	Goal	L10	Avg.
Future pixels	**98.0**	99.8	91.8	94.2	96.0
Action only	97.6	99.6	92.8	89.6	94.9
Single-level states	96.8	99.8	94.0	94.6	96.3
Multi-level states	97.0	**100.0**	**96.8**	**97.4**	**97.8**

R2G: conditioning the generator on the learned representation. We compare Enfold with a video-only Cosmos baseline under a matched training budget and evaluation protocol. Both models predict the future from the same visual and language context.

Adding z raises PSNR from 25.92 to 27.27 dB and SSIM from 0.885 to 0.902, while reducing LPIPS by 12.2%, from 0.0550 to 0.0483 (Table 5). Agreement across distortion, structural, and perceptual metrics

Table 5: R2G conditioning improves LIBERO future prediction.

Metric	Video-only	Enfold
PSNR ↑	25.92	**27.27**
SSIM ↑	0.885	**0.902**
LPIPS ↓	0.0550	**0.0483**

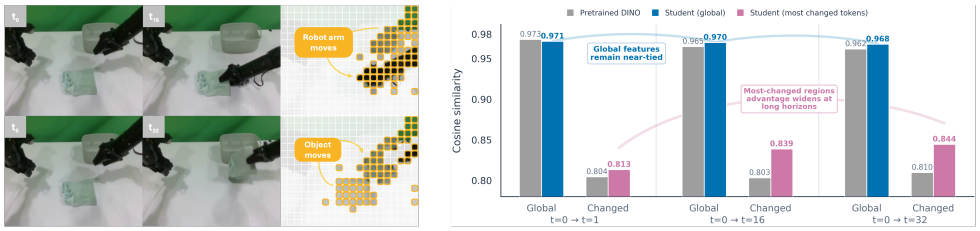


Figure 7: **Future-change probe.** Left: future DINO displacement localizes to moving arms and cloth. Right: Enfold's advantage over current-frame DINO grows with horizon.

argues against a metric-specific explanation. Because z uses the same context and adds no observation, the gain indicates that the encoder reorganizes context for the generator. Together, the ablations validate both directions: generator states bettersupervise z, while z remains useful to the generator rather than becoming a policy-specific summary. This reciprocity supports a shared predictive interface.

4.5 What Enfold Encodes

Task performance shows the utility of generator-state supervision but not what the representation encodes. Since generator states contain transition structure and stochastic variation, we ask whether Enfold reproduces this nuisance or learns future-oriented structure beyond static visual features.

From stochastic teacher features to a stable predictive representation. Raw generator states vary with scene content and generation noise (Section 2). We compare Enfold with representative Cosmos layers under controlled lighting perturbations. Nuisance sensitivity is feature displacement normalized by variation across input videos.

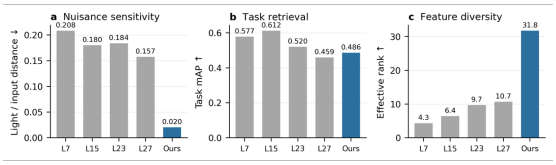


Figure 8: Cosmos-state and Enfold-representation probes at $t = 600$ on 100 clips. Lower lighting sensitivity is better; task mAP and effective rank measure task selectivity and feature diversity.

Enfold reduces this ratio from 0.157–0.208 for generator layers to 0.020, a 7.9–10.4× reduction (Figure 8). On this 100-clip set, its effective rank is 31.8, versus at most 10.7 for generator features, while retaining 0.486 task-retrieval mAP. The higher rank rules out feature collapse and shows the student is not a compressed copy of one teacher layer. Predictive learning instead filters variation unpredictable from the present while reorganizing shared structure into a diverse representation.

Predicting future change. Equal-capacity probes predict future DINO features from Enfold or current DINO, globally and on an analysis-only, future-selected patch subset. On moving patches (Figure 7), Enfold reduces residual cosine error over DINO by 4.6%, 18.1%, and 18.2% at $t = 1, 16, 32$. Globally, it trails slightly at $t = 1$ but reduces error by 14.0% and 15.2% at $t = 16, 32$. This short-horizon exception argues against a uniformly stronger image descriptor: Enfold's advantage grows with horizon and localizes to changing patches, indicating future-dependent rather than generic visual content. Appendix C.6 shows complementary token geometry.

5 Discussion and Conclusion

World generative models usually generate a future or remain in the control loop. Enfold instead predicts their future-conditioned states from current observations and instructions, transferring future-change structure into a present representation useful for generation and control without action gradients. Coordinated adaptation under intervention suggests recomputation rather than replay, although this evidence is qualitative and the generator remains required in training. Enfold internalizes generative computation for efficient control, invoking it only when needed.

6 AI USE STATEMENT.

Generative AI tools were used during manuscript preparation to draft and edit text for readability, refine the presentation of existing material, and assist with formatting tables, citations, and document layout. They were not used for generating or cleaning data, developing conceptual or theoretical models, translating research content, analyzing data, or interpreting results. The authors reviewed and verified all AI-assisted work and take full responsibility for the final text, claims, results, and artifacts in this paper.

7 ETHICS STATEMENT.

This work does not evaluate people or collect personally identifying or sensitive data. Teleoperated demonstrations contain only robot observations and actions from controlled tabletop scenes. Physical-robot evaluations follow workspace boundaries, termination limits, and safety-triggered stopping (Appendix C). Learned robot policies can fail unexpectedly; use beyond controlled research settings requires application-specific validation, safeguards, and human oversight.

8 REPRODUCIBILITY STATEMENT.

The objectives and gradient routes are specified in Equations 4–8; Section 4 defines the benchmarks, baselines, metrics, and evaluation setup. Appendix C reports dataset sizes, model configurations, hyperparameters, per-task results, and evaluation protocols, while the remaining appendices document theoretical and representation analyses. Together, these materials specify the implementation and evaluation needed to reproduce the results.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-Reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. 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, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024a.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 33431–33452, 2024b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

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

Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. *Advances in Neural Information Processing Systems*, 37:139002–139029, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, et al. Aha-wam:asynchronous horizon-adaptive world-action modeling with observation-guided context routing. *arXiv preprint arXiv:2606.09811*, 2026.

Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Feilong Tang, Donglin Wang, and Haoang Li. Unified diffusion vla: Vision-language-action model via joint discrete denoising diffusion process. *arXiv preprint arXiv:2511.01718*, 2025a.

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*, 2025b.

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. IGOR: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025c.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19752–19763, 2025d.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, Leslie Kaelbling, et al. Video language planning. In *International Conference on Learning Representations*, volume 2024, pp. 31138–31155, 2024.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.

Kyle B Hatch, Ashwin Balakrishna, Oier Mees, Suraj Nair, Seohong Park, Blake Wulfe, Masha Itkina, Benjamin Eysenbach, Sergey Levine, Thomas Kollar, et al. GHIL-Glue: Hierarchical control with filtered subgoal images. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9516–9524. IEEE, 2025.

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. *arXiv preprint arXiv:2412.14803*, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. \pi_ {0.5}: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

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

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuao Pei, Tian Nian, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025a.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025b.

Yudong Liu, Yuan Li, Zijia Tang, Yuxi Zheng, Yueqian Lin, Qinsi Wang, Yi Li, Shuangjun Liu, Shuai Zhang, Taotao Jing, et al. Latent bridge: Feature delta prediction for efficient dual-system vision-language-action model inference. *arXiv preprint arXiv:2605.02739*, 2026.

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. *arXiv preprint arXiv:2605.00078*, 2026.

Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*, 2025. URL https://arxiv.org/abs/2509.06951.

NVIDIA. NVIDIA TensorRT: An sdk for high-performance deep learning inference, 2026.

NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots. NVIDIA Research, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_6/.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations*, volume 2024, pp. 9379–9395, 2024.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

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 *International Conference on Learning Representations*, volume 2025, pp. 92033–92052, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. ABot-M0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026a.

Yi Yang, Xueqi Li, Yiyang Chen, Jin Song, Yihan Wang, Zipeng Xiao, Jiadi Su, You Qiaoben, Pengfei Liu, and Zhijie Deng. Mantis: A versatile vision-language-action model with disentangled visual foresight. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42505–42515, 2026b.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Hanyang Yu, Haitao Lin, Jingbo Zhang, Wenyao Zhang, Chenghao Gu, Heng Li, and Ping Tan. Maskwam: Unifying mask prompting and prediction for world-action models. *arXiv preprint arXiv:2606.13515*, 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.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, et al. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026a.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Yuhao Zhang, Wanxi Dong, Yue Shi, Yi Liang, Jingnan Gao, Qiaochu Yang, Yaxing Lyu, Zhixuan Liang, Yibin Liu, Congsheng Xu, et al. R3dp: Real-time 3d-aware policy for embodied manipulation. *arXiv preprint arXiv:2603.14498*, 2026b.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. Imagewam: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026c.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025a.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, volume 2025, pp. 54277–54296, 2025b.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

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

A Related Work

Figure 1 organizes prior work by the interface through which future information reaches action prediction. The following discussion reviews three corresponding lines of research: future-aware VLA models, generative world-action models, and predictive representation learning.

A.1 Vision–Language–Action Models

Vision–language–action (VLA) models transfer large-scale vision–language pre-training to robotic control through scalable adaptation and expressive action decoding (Brohan et al., 2022; Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024a; Intelligence et al., 2025; Kim et al., 2025; Liu et al., 2025b; Zheng et al., 2025a; Liu et al., 2025a; Chen et al., 2025a; Liang et al., 2025). Standard VLA objectives map observations and instructions directly to actions, leaving environmental dynamics implicit in action supervision. Recent methods introduce future queries, latent actions, visual subgoals, or auxiliary prediction to strengthen temporal reasoning (Zheng et al., 2025b; Wu et al., 2026; Bu et al., 2025; Hu et al., 2024; Tian et al., 2025; Yang et al., 2026b; Sun et al., 2026). Complementary efficiency-oriented work predicts feature or cache updates so that an expensive VLM need not run at every control step (Liu et al., 2026). VLA–JEPA is especially relevant to predictive supervision: a current-only student predicts target-encoder features computed from future frames, defining the future in the representation space of a discriminative encoder rather than through the intermediate computation of a generative model (Sun et al., 2026).

A.2 Generative World-Action Models

World-action models (WAMs) use predicted scene evolution as supervision or context for action generation. Imagine-then-act systems first predict a visual future and recover actions through inverse dynamics or a goal-conditioned policy (Du et al., 2023; 2024; Black et al., 2024b; Hatch et al., 2025; Bu et al., 2024; Tian et al., 2025; Hu et al., 2024). Unified architectures instead model videos and actions jointly, allowing dense visual dynamics to complement action labels (Lv et al., 2025; Bi et al., 2026; Kim et al., 2026; Li et al., 2026; Ye et al., 2026; Luo et al., 2026; Zhang et al., 2026a; Agarwal et al., 2026). Other work reduces explicit rollout through auxiliary future objectives, reusable latent context, visual subgoals, asynchronous prediction, or structured world representations (Yuan et al., 2026; Gao et al., 2025; Luo et al., 2026; Cai et al., 2026; Zhang et al., 2026c; Yu et al., 2026; Zhang et al., 2026b). Among these, VPP conditions a policy on predictive features extracted inside a video diffusion model (Hu et al., 2024). Fast–WAM retains video co-training but removes explicit future generation from its action-time path (Yuan et al., 2026). AHA–WAM amortizes a low-frequency video branch through reusable layer-wise context, whereas ImageWAM supplies its action expert with the caches produced by image-editing denoising (Cai et al., 2026; Zhang et al., 2026c). These methods establish that useful world-model context need not take the form of decoded video.

A.3 Predictive Representation Learning

Predictive representation learning models temporal structure in feature space rather than reconstructing complete observations. JEPA-style methods predict future target-encoder features, while latent-action methods infer transition variables through inverse and forward dynamics (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025; Schmidt & Jiang, 2024; Bruce et al., 2024; Ye et al., 2025). Pre-trained visual features have also been used as latent dynamics spaces for planning (Zhou et al., 2024). These representations support VLA pre-training, transfer, and alignment with executable robot actions (Chen et al., 2024; Bu et al., 2025; Chen et al., 2025d;c). Most such targets are produced by an encoder or an inferred transition variable and therefore describe future observations without exposing the intermediate temporal computation of a generative model.

Positioning of this work. Enfold differs from prior work in where predictive information is obtained and how it enters control. VLA–JEPA predicts features defined by a future encoder, whereas VPP, AHA–WAM, and ImageWAM obtain action-time context from an executed generative path or its caches. Fast–WAM avoids video rollout at inference but learns video and action prediction within a coupled architecture. Enfold instead uses multi-level states from a teacher-forced generator as training targets for a current-only encoder. The resulting representation is not shaped by task

gradients, can condition both action prediction and future generation, and requires no generator execution in the control path. Enfold therefore transfers future-conditioned generative computation into the present representation, rather than retaining the generator as part of action-time computation.

B A PREDICTIVE-PROJECTION INTERPRETATION OF G2R

G2R uses future-conditioned generator states as supervision even though the future itself is unavailable at deployment. What, then, can a current-only encoder learn from targets that also depend on the realized future and the stochastic generative process? We give an idealized predictive-projection interpretation. At a fixed stage of training, we hold the target-producing computation—the generator and the detached map from the current context to $\bar{z}$—fixed, and consider the population regression problem it induces. The analysis characterizes the direction of a G2R update. It is not a convergence claim for the jointly evolving training system.

Let $X = (c, e)$ collect the current visual context and language instruction, and let $Y \sim p(y \mid X)$ denote a realized future. Together with $\epsilon \sim \mathcal{N}(0, I)$ and corruption timestep t, the teacher-forced pass in Eq. 3 defines a random target $r_t^G = r_t^G(X, Y, \epsilon)$. Although this target is deterministic once (X, Y, ϵ, t) are fixed, it remains random from the perspective of the current context. Consider the squared-error counterpart of Eq. 4,

$$\widetilde{\mathcal{L}}_{\mathrm{G2R}}(f) = \mathbb{E}\left[\left\| f(X,t) - r_t^G \right\|_2^2\right]. \tag{9}$$

For an unrestricted predictor and a fixed induced target distribution, the population optimum is

$$f^\star(X,t) = \mathbb{E}\big[r_t^G \mid X, t\big]. \tag{10}$$

This conditional expectation is the L_2 projection of the generator state onto functions of the information available to the predictor. In particular, writing

$$r_t^G = \underbrace{\mathbb{E}[r_t^G \mid X,t]}_{r_t^{G,\mathrm{pred}}} + \underbrace{\big(r_t^G - \mathbb{E}[r_t^G \mid X,t]\big)}_{r_t^{G,\mathrm{res}}}, \qquad \mathbb{E}[r_t^{G,\mathrm{res}} \mid X,t] = 0, \tag{11}$$

gives, for any predictor $f(X,t)$,

$$\mathbb{E}\big[\|r_t^G - f(X,t)\|_2^2\big] = \mathbb{E}\Big[\|r_t^{G,\mathrm{res}}\|_2^2\Big] + \mathbb{E}\Big[\|r_t^{G,\mathrm{pred}} - f(X,t)\|_2^2\Big]. \tag{12}$$

The residual term includes variation due to unresolved future realizations, corruption noise, and any other target component that is not systematically predictable from (X, t). At the population level, reducing G2R cannot recover this term. It instead rewards the component of the generator computation that covaries predictably with the current scene and instruction. This does not rule out finite-sample memorization, but it explains why G2R should not be viewed as unconditional feature copying. It is also consistent with the substantially lower nuisance sensitivity of the learned representation reported in Section 4.5.

The implemented objective only approximates this idealized projection. $F_\omega \circ U_\phi$ is a finite-capacity function class, the target-producing branch changes over training, and Eq. 4 uses SmoothL1 rather than squared error. Under SmoothL1, the population optimizer is a robust conditional center instead of the conditional mean exactly. It approaches mean regression in the quadratic regime while limiting the influence of large target deviations. The projection view should therefore be read as an interpretation of what information the objective makes learnable, rather than as an exact closed-form description of the trained network.

This view also clarifies the timestep-conditioned and multi-level design. The encoder produces a shared state $z = U_\phi(X)$ that is independent of t, while $F_\omega(z, \gamma(t))$ predicts the target in the coordinate system induced at timestep t. Thus, z must support prediction of the family

$$\big\{\mathbb{E}[r_t^G \mid X,t]\big\}_{t\in\mathcal{T}}, \tag{13}$$

and the prediction head accounts for how that structure is expressed along the generative trajectory. Concatenating several generator depths extends this family across abstraction levels, allowing the predictor to draw on complementary structure without assuming that any single layer is universally optimal.

Finally, the interpretation distinguishes predictive organization from information gain. For a fixed trained encoder, $Z = U_\phi(X)$ is a deterministic function of the deployment-time input, and therefore

$$I(Y; Z \mid X) = 0. \tag{14}$$

Conditioning the generator on Z cannot reveal sample-specific future information absent from X. An R2G improvement instead indicates that Z reorganizes predictive structure already present in the context into coordinates that are more accessible to a finite-capacity generator. Detached task heads ask the complementary question of whether the same organization is actionable without allowing task gradients to define it.

Under these qualifications, Enfold can be viewed as learning an amortized, robust approximation to

$$\boxed{r_t^G = \mathbb{E}[r_t^G \mid c, e, t] + r_t^{G,\mathrm{res}}}. \tag{15}$$

The conditional expectation is the predictable generative structure, whereas the residual contains unresolved future and stochastic variation. G2R makes the first component accessible from the present, while R2G and the detached task readouts test whether the resulting representation remains useful for generation and control.

C Experimental Details

C.1 Benchmarks and Model Configuration

We use the four standard 10-task LIBERO suites (Liu et al., 2023). After removing no-operation transitions, we train on 1712 demonstrations and evaluate each task with 50 trials. RoboTwin2.0 provides 50 dual-arm tasks and structured domain randomization axes (Chen et al., 2025b). Our training set contains 27500 demonstrations, and evaluation uses the clean and randomized scene configurations. The real-world dataset contains 400 demonstrations from four tabletop tasks collected with an AgileX dual-arm platform.

Backbones and feature routing. All three embodiments use DINOv3 ViT-H+/16 as the predictive visual encoder (Siméoni et al., 2025). LIBERO supplies two views at 224×448, the real-robot policy uses one frontal 256×320 view, and RoboTwin2.0 uses front and bilateral wrist views at 384×320. Cosmos-Reason1-7B language features (Azzolini et al., 2025) are linearly projected and concatenated into the DINO stream from block 8. The action head reads projected patch tokens from the final DINO block. Features from blocks $\{7, 15, 23, 31\}$ are instead concatenated only for the predictive head and its multi-level supervision. Detached features from Cosmos blocks $\{7, 15, 23, 27\}$ provide the G2R targets (Agarwal et al., 2025), and the predicted representation is fed back to the same generator through R2G.

Prediction and action heads. The predictive head has 10 layers, maps $5120 \to 1024 \to 8192$, and combines its conditions by concatenation with 3D RoPE. The action head uses the same concatenative conditioning with 1D RoPE. It contains 10 layers for LIBERO and real-robot control and 16 layers for RoboTwin2.0.

Temporal processing, inference, and optimization. Each observation conditions a 32-step action chunk and a corresponding video sequence containing one context frame and 32 future frames. Temporal subsampling with stride 4 reduces the video sequence to $1 + 8$ frames. The causal VAE further compresses it to $1 + 2$ latent frames. Actions are trained with continuous flow matching from Gaussian noise and sampled with a 10-step Euler solver using a noise shift of 5.0. The policy replans after executing 10 actions on LIBERO and the real robot and 24 actions on RoboTwin2.0. All models use bf16, AdamW with a learning rate of 10^{-4}, a cosine schedule, and unit coefficients for the R2G and task losses. Enfold-Flash retains the same representation interface and adds TensorRT operator acceleration (NVIDIA, 2026). Internally trained baselines and ablations use the same observations, language conditions, task data, and evaluation protocols. Table 6 summarizes the dataset-specific settings.

Table 6: **Training and policy configuration.**

Configuration	LIBERO	Real Robot	RoboTwin2.0
Encoder and feature prediction			
Predictive encoder	DINOv3 ViT-H+/16	DINOv3 ViT-H+/16	DINOv3 ViT-H+/16
Camera views	2 (front, wrist)	1 (front)	3 (front, left wrist, right wrist)
Input resolution	224×448	256×320	384×320
Encoder positional encoding	2D RoPE	2D RoPE	2D RoPE
Text–DINO concat start layer	8	8	8
DINO feature concat layers	$\{7, 15, 23, 31\}$	$\{7, 15, 23, 31\}$	$\{7, 15, 23, 31\}$
Generative-model feature layers	$\{7, 15, 23, 27\}$	$\{7, 15, 23, 27\}$	$\{7, 15, 23, 27\}$
Generative condition	Concat, detached	Concat, detached	Concat, detached
Prediction and action heads			
Prediction-head depth	10	10	10
Prediction-head dimensions	$5120 \to 1024 \to 8192$	$5120 \to 1024 \to 8192$	$5120 \to 1024 \to 8192$
Prediction-head condition injection	Concat	Concat	Concat
Prediction-head positional encoding	3D RoPE	3D RoPE	3D RoPE
Action-head depth	10	10	16
Action-head hidden dimension	1024	1024	1024
Action-head condition injection	Concat	Concat	Concat
Action-head positional encoding	1D RoPE	1D RoPE	1D RoPE
Diffusion-time embedding dimension	256	256	256
Temporal horizons			
Action context / future horizon	$1/32$ steps	$1/32$ steps	$1/32$ steps
Raw-video context / future horizon	$1/32$ frames	$1/32$ frames	$1/32$ frames
Temporal video sampling stride	4	4	4
VAE-input context / future horizon	$1/8$ frames	$1/8$ frames	$1/8$ frames
VAE temporal compression ratio	4	4	4
Latent-video context / future horizon	$1/2$ latents	$1/2$ latents	$1/2$ latents
Action generation and execution			
Action generation objective	Flow matching	Flow matching	Flow matching
Initial action distribution	$\mathcal{N}(0, I)$	$\mathcal{N}(0, I)$	$\mathcal{N}(0, I)$
Training noise timesteps	1000	1000	1000
Train / inference noise shift	5.0/5.0	5.0/5.0	5.0/5.0
Action sampling solver	Euler	Euler	Euler
Action sampling steps	10	10	10
Executed actions per replanning	10	10	24
Guidance scale	1.0	1.0	1.0
Training			
Training hardware	1 node / 8× A100 40GB	1 node / 8× A100 40GB	4 nodes / 32× A100 40GB
Wall-clock training time	23 h	44 h	108 h
Global batch size	128	128	320
Optimizer and learning rate	AdamW / 1×10^{-4}	AdamW / 1×10^{-4}	AdamW / 1×10^{-4}
Learning-rate scheduler	Cosine	Cosine	Cosine
$\lambda_{\text{R2G}}, \lambda_{\text{task}}$	$1, 1$	$1, 1$	$1, 1$
Training epochs	10	10	5
Training steps	21,700	40,800	94,925
Numerical precision	bf16	bf16	bf16

C.2 Control Evaluation Protocols

Real robot. We deploy on a tabletop AgileX dual-arm platform observed by one fixed frontal RGB camera. Images are resized to 256×320. At each replanning step, Enfold encodes the latest observation and language instruction, predicts a 32-step action chunk, and executes the first 10 actions before acquiring a new observation. The dataset contains 400 teleoperated demonstrations, with 100 demonstrations for each of the four tasks.

Following a staged-completion evaluation protocol (Cai et al., 2026), we run $N = 30$ independent rollouts for each method, task, and evaluation setting. If rollout i completes $s_i \in \{0, 1, 2, 3\}$ ordered stages, its normalized completion score is

$$\text{Score} = \frac{100}{3N} \sum_{i=1}^{N} s_i. \tag{16}$$

A score of 3 is full task success. Lower scores retain partial progress. At a timeout or safety-triggered termination, the score is fixed at the last completed stage, and no rollout is discarded.

ID trials sample object poses and workspace configurations from the ranges used during data collection. OOD trials keep the instruction and completion criteria fixed while changing one factor when-

Table 7: **Per-task success rates on RoboTwin2.0 under clean and randomized evaluation settings.** The best value(s) in each row and setting are shown in **bold**. The second-best distinct value(s) are underlined.

	Enfold		Enfold-Flash		Fast-WAM		LingBot-VA		$\pi_{0.5}$		Motus	
Task	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	90%	94%	**100%**	99%	89%	93%
Beat Block Hammer	97%	**98%**	98%	94%	**99%**	97%	96%	**98%**	96%	93%	95%	88%
Blocks Ranking RGB	99%	95%	99%	94%	**100%**	**100%**	99%	98%	92%	85%	99%	97%
Blocks Ranking Size	91%	95%	91%	91%	**94%**	**98%**	**94%**	96%	49%	26%	75%	63%
Click Alarmclock	97%	**100%**	**100%**	**100%**	**100%**	**100%**	99%	**100%**	98%	89%	**100%**	**100%**
Click Bell	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	99%	66%	**100%**	**100%**
Dump Bin Bigbin	**99%**	93%	97%	95%	97%	96%	89%	96%	92%	**97%**	95%	91%
Grab Roller	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Handover Block	93%	73%	86%	78%	95%	**81%**	**99%**	78%	66%	57%	86%	73%
Handover Mic	99%	**100%**	**100%**	**100%**	99%	**100%**	94%	96%	98%	97%	78%	63%
Hanging Mug	48%	45%	51%	44%	**58%**	**62%**	40%	28%	18%	17%	38%	38%
Lift Pot	97%	**100%**	99%	**100%**	**100%**	**100%**	**100%**	99%	96%	85%	96%	99%
Move Can Pot	**95%**	93%	93%	96%	90%	88%	94%	**97%**	51%	55%	34%	74%
Move Pillbottle Pad	98%	98%	96%	97%	**100%**	**99%**	99%	**99%**	84%	61%	93%	96%
Move Playingcard Away	98%	**100%**	98%	99%	**100%**	**100%**	**100%**	99%	96%	84%	**100%**	96%
Move Stapler Pad	72%	71%	62%	74%	77%	64%	**91%**	79%	56%	42%	83%	**85%**
Open Laptop	97%	99%	**98%**	99%	**98%**	**100%**	92%	94%	90%	96%	95%	91%
Open Microwave	83%	94%	89%	**97%**	62%	45%	82%	86%	34%	77%	**95%**	91%
Pick Diverse Bottles	80%	81%	78%	85%	80%	85%	89%	82%	81%	71%	**90%**	**91%**
Pick Dual Bottles	88%	93%	89%	87%	**100%**	96%	**100%**	**99%**	93%	63%	96%	90%
Place A2B Left	95%	**95%**	94%	**95%**	95%	93%	**97%**	93%	87%	82%	88%	79%
Place A2B Right	93%	91%	93%	92%	93%	**99%**	**97%**	95%	87%	84%	91%	87%
Place Bread Basket	93%	**96%**	96%	95%	91%	93%	**97%**	95%	77%	64%	91%	94%
Place Bread Skillet	93%	87%	91%	91%	90%	**93%**	**95%**	90%	85%	66%	86%	83%
Place Burger Fries	97%	**100%**	**98%**	99%	96%	99%	97%	95%	94%	87%	**98%**	98%
Place Can Basket	75%	63%	75%	62%	71%	69%	**81%**	**84%**	62%	62%	**81%**	76%
Place Cans Plasticbox	99%	**99%**	99%	97%	99%	96%	**100%**	**99%**	94%	84%	98%	94%
Place Container Plate	**100%**	97%	**100%**	98%	96%	**100%**	99%	97%	99%	95%	98%	99%
Place Dual Shoes	86%	**90%**	92%	88%	**94%**	88%	**94%**	89%	75%	75%	93%	87%
Place Empty Cup	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	99%	99%	98%
Place Fan	93%	90%	93%	94%	96%	**96%**	**99%**	93%	87%	85%	91%	87%
Place Mouse Pad	90%	94%	**94%**	95%	83%	89%	93%	**96%**	60%	39%	66%	68%
Place Object Basket	81%	80%	84%	82%	89%	**88%**	**91%**	**88%**	80%	76%	81%	87%
Place Object Scale	92%	96%	94%	96%	90%	**97%**	**96%**	95%	86%	80%	88%	85%
Place Object Stand	95%	95%	95%	**97%**	90%	94%	**99%**	96%	91%	85%	98%	**97%**
Place Phone Stand	**98%**	98%	95%	**99%**	97%	**99%**	97%	97%	81%	81%	87%	86%
Place Shoe	95%	97%	96%	95%	96%	**99%**	98%	98%	92%	93%	**99%**	97%
Press Stapler	99%	**99%**	**100%**	**99%**	90%	97%	85%	82%	87%	83%	93%	98%
Put Bottles Dustbin	93%	89%	89%	**93%**	**95%**	90%	87%	91%	84%	79%	81%	79%
Put Object Cabinet	83%	89%	82%	**90%**	**94%**	89%	85%	87%	80%	79%	88%	71%
Rotate QRcode	83%	89%	88%	86%	93%	89%	**96%**	**91%**	89%	87%	89%	73%
Scan Object	86%	**93%**	84%	88%	89%	92%	**96%**	91%	72%	65%	67%	66%
Shake Bottle	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	97%	99%	97%	**100%**	97%
Shake Bottle Horizontally	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	99%	99%	99%	**100%**	98%
Stack Blocks Three	96%	96%	98%	97%	95%	97%	**99%**	**98%**	91%	76%	91%	95%
Stack Blocks Two	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	98%	97%	**100%**	**100%**	98%
Stack Bowls Three	85%	84%	82%	**92%**	80%	81%	**86%**	83%	77%	71%	79%	87%
Stack Bowls Two	97%	**99%**	96%	95%	92%	98%	94%	98%	95%	96%	**98%**	98%
Stamp Seal	81%	94%	86%	93%	90%	94%	**96%**	**97%**	79%	55%	93%	92%
Turn Switch	71%	69%	80%	66%	61%	59%	44%	45%	62%	54%	**84%**	**78%**
Average	91.60%	91.94%	91.96%	**92.08%**	91.88%	91.78%	**92.90%**	91.50%	82.74%	76.76%	88.66%	87.02%

Table 8: **Ordered task stages used for real-robot scoring.** Completing stages 1–3 yields scores 1–3, respectively. Score 0 denotes no completed stage.

Task	Stage 1	Stage 2	Stage 3 (full completion)
Fold Towel	Form the first bimanual fold	Reach the prescribed folded shape	Complete the final placement of the folded towel
Organize Desktop	Clear the first designated object group	Place the remaining intermediate group	Move all required objects to their target locations
Spoon Powder	Acquire and align the spoon	Scoop and transfer the powder	Complete the transfer and return the tool
Store Plate	Securely grasp the plate	Reorient it for storage	Place it stably in the target rack

ever possible. Configuration shifts alter the towel geometry, object appearance, or the workspace background. All methods use the same initial-state sampler, trial allocation, camera observations, and termination limits within each setting.

C.3 Future-Video Evaluation

The video-only Cosmos baseline and Enfold are trained for the same 21.7K optimization steps and evaluated on 100 fixed LIBERO demonstration clips. Given the first frame and language instruction, each model predicts the next 32 frames with 35 denoising steps. PSNR, SSIM, and LPIPS are computed only over the predicted frames.

C.4 Representation Analyses

Stability. We compare the final Enfold predictive encoder with Cosmos layers L7, L15, L23, and L27 at $t = 600$ on 100 clips and two noise seeds. Text conditioning is disabled. Lighting sensitivity is the feature displacement under the controlled illumination perturbation normalized by the mean displacement across different input clips. For diversity, each model contributes one pooled feature per clip. the two seed-specific Cosmos features are averaged before centering. Thus, for model M, effective rank is evaluated on $X^M_{100} \in \mathbb{R}^{100\times d_M}$ using Equation 25, and is bounded by 99. Task/scene retrieval uses the same 100 clip-level features. This evaluation is separate from the 20-video generator-state probe in Appendix D.

Future-feature probe. The probe uses 100 real-world episodes of the fold-towel task: 80 for training and 20 for validation. Equal-capacity MLPs predict the frozen-DINO feature z^D_t from either the Enfold representation z^S_0 or the current-frame DINO feature z^D_0. For $X \in \{S, D\}$, we report

$$s^X_t = \cos\big(f_t(z^X_0), z^D_t\big), \qquad t \in \{1, 16, 32\},$$

averaged over three probe seeds. The changed-region diagnostic uses the $k = 64$ patches with the largest future DINO displacement, $1 - \cos(z^D_{0,j}, z^D_{t,j})$, and both input representations are evaluated at the same patch locations.

Token-similarity visualization. For each observation, we manually select a query point on the gripper and map it to the corresponding patch token. We then compute its cosine similarity to every token in the frame, independently for frozen DINO and the instruction-conditioned Enfold predictive encoder.

C.5 Human-Intervention Visualization

The two episodes in Figure 6 begin from the standard task distribution. At t_2, a human relocates the plate or displaces the cloth without changing the language instruction. The next observation is processed at the normal control frequency to obtain an updated predictive representation. The middle panel is decoded from that representation only for visualization. The physical rollout continues using actions predicted directly from the representation. These examples are qualitative and are not used to estimate a recovery rate. No matched frozen-representation or no-intervention rollout is reported.

C.6 Additional Qualitative Results

Interaction structure in the token geometry. The fitted probe establishes future predictability but not how it is organized spatially. We therefore inspect the representation without a learned readout. For each observation, a query point on the gripper is mapped to its patch token, and we compute $\cos(q_t, k_{t,j})$ against all tokens in the same frame. Frozen DINO and Enfold are evaluated independently at the same query location.

Frozen DINO mainly associates the query with visually similar parts of the arm (Figure 9). In Enfold, high similarity extends from the gripper to the manipulated object and the regions involved in contact. The pattern appears before contact and persists while the object is moved, in both real and simulated examples. The cross-object association cannot be explained by local appearance similarity alone, and it is consistent with the token geometry organizing an interaction relation between the agent and the object. This visualization remains qualitative, but it complements the fitted probe: generator-mediated supervision shifts the representation toward the parts of the scene that participate in the future interaction.

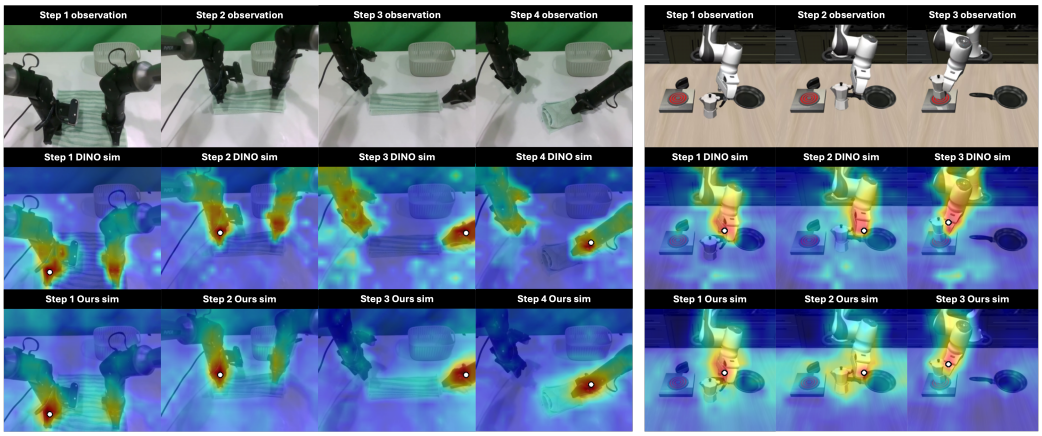


Figure 9: **Query-point similarity overlays.** The left four columns show a rollout of the real-world fold-towel task, while the right three columns show LIBERO trajectories. White dots mark manually selected query points on the robot gripper. The middle row shows cosine-similarity overlays from frozen DINO patch tokens, while the bottom row shows overlays from the Enfold predictive encoder. Whereas frozen DINO mainly responds to visually similar arm regions, Enfold also highlights the manipulated object and contact-relevant regions that subsequently change during manipulation.

PCA Visualization of Token Geometry Figure 10 provides a global view of the token organization underlying the query-point similarity analysis in Figure 9. For each observation, we project the spatial patch tokens to three principal components and map them to RGB.

Frozen DINO produces relatively sharp changes in the PCA map at visual object boundaries. The robot arm, manipulated object, tabletop, and background occupy distinct feature regions, consistent with a representation organized primarily by appearance and object identity. After generator-mediated training, Enfold changes this geometry near the manipulation. Across both the real-world fold-towel rollout and the LIBERO trajectory, the gripper and the object being operated on form more similar and spatially connected PCA regions, despite remaining visually distinct in the input image. Tokens outside the active interaction remain separated.

This complements Figure 9. The PCA maps show the broader reorganization of the token field, whereas the gripper-query cosine-similarity overlays directly measure the increased association between the gripper, manipulated object, and contact-relevant regions. Together, the two visualizations indicate that Enfold does not simply improve object discrimination. It reorganizes the representation around the interaction that determines the future manipulation.

Teacher-feature reconstruction and localization of future-prediction gains. The future-feature probe in Figure 7 uses fitted readouts to test whether the Enfold representation predicts a future frozen-DINO feature. Figure 11 instead visualizes the training-aligned G2R prediction directly. Given the current Enfold representation z, the prediction head produces $\hat{r}_\tau^G = F_\omega(z, \gamma(\tau))$ to match the Cosmos teacher target r_τ^G. At both displayed future horizons, the predicted features retain the coarse spatial organization of the teacher target, including the regions around the moving gripper, cloth, and their interaction.

The final column localizes the future-feature prediction result in Figure 7. It visualizes $\Delta s_h = s_h^S - s_h^D$ on the most-changed tokens, where s_h^S and s_h^D are the future-feature cosine scores obtained from Enfold and frozen DINO, respectively. Most selected tokens have positive values: a readout from the Enfold representation predicts their future DINO features more accurately than a readout from the current frozen-DINO representation. The positive regions concentrate on the moving arms, manipulated cloth, and contact-relevant areas. This spatial pattern is consistent with the trend in Figure 7: Enfold's advantage grows at longer horizons and is largest where the scene changes, rather than on static appearance-dominated regions.

Qualitative comparison of future-video artifacts. Figure 12 compares predicted future videos with the observations recorded during execution. The top two rows show FastWAM, and the bottom two rows show Enfold. For each method, the upper row is the executed rollout and the lower row is

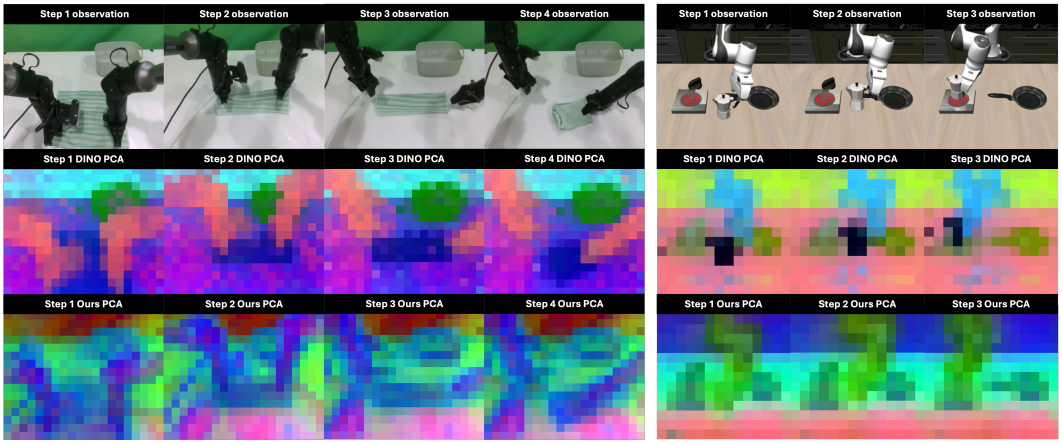


Figure 10: **PCA visualization of frozen DINO and Enfold token geometry.** The top row shows rollout observations. The middle and bottom rows visualize the corresponding spatial patch tokens from frozen DINO and the Enfold predictive encoder, respectively, by mapping their first three principal components to RGB. Frozen DINO separates the robot arm and task objects into visually distinct feature regions. Enfold makes the gripper and the object involved in the current manipulation more similar in the projection, forming a relational structure around the interaction.

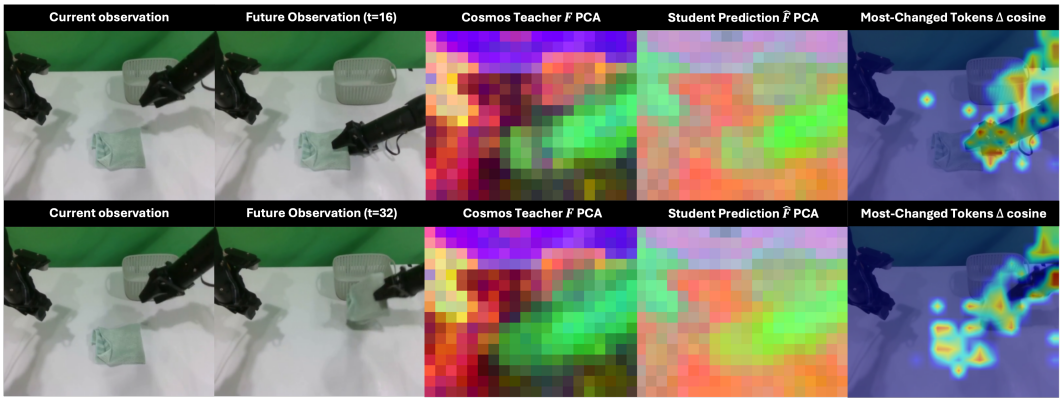


Figure 11: **Future alignment with teacher representations and prediction at most-changed tokens.** Rows show future horizons $t = 16$ and $t = 32$. From left to right: current and future observations, Cosmos teacher-target PCA, prediction-head output PCA, and the future-prediction gain $\Delta s_h = s_h^S - s_h^D$ on most-changed tokens.

the predicted future video. The comparison focuses on object presence and gripper-object contact, since an incorrect prediction of either can lead the policy to act on a state that has not actually occurred.

In the LIBERO example, FastWAM's wrist-view prediction places the object between the gripper fingers even though the object is still outside the gripper in the real rollout. The policy consequently continues to issue grasping motions, but cannot pick up the object. In *Fold Towel*, FastWAM predicts that the left arm has already grasped the towel before the grasp is established. It then proceeds as if the towel were secured, while the ungrasped towel cannot be folded. In *Store Plate*, FastWAM similarly predicts that the left arm has lifted the plate although it has not. Later predicted frames still show a plate in the hand and lead to a handover motion, but the gripper is empty. In all three cases, Enfold's predicted frames agree with the observed object and contact states, avoiding these false grasp and handover states during execution.

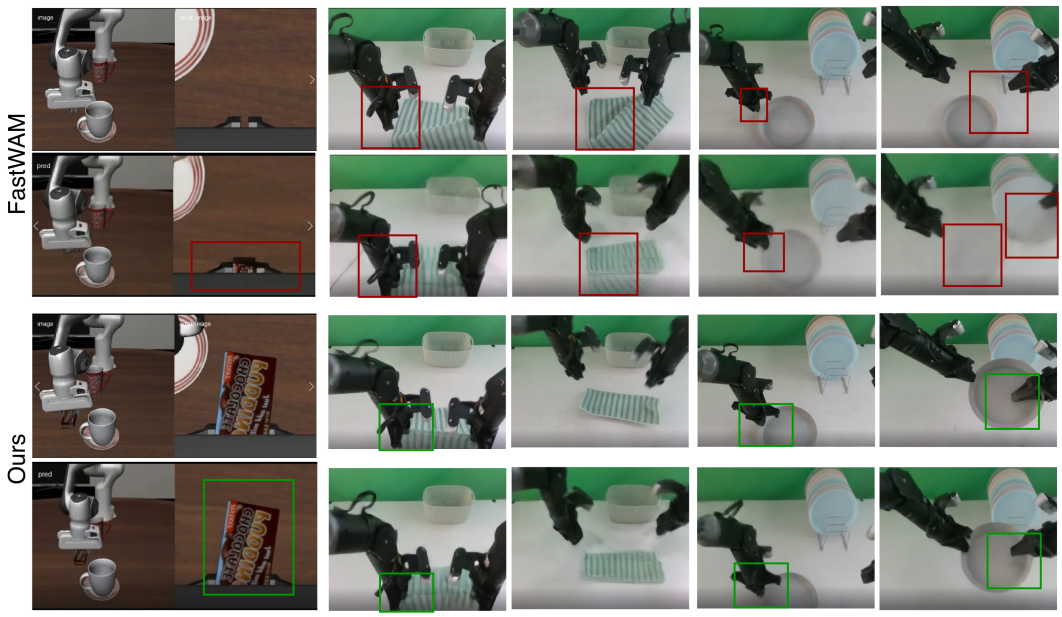


Figure 12: **Qualitative comparison of future-video predictions.** The left group shows a LIBERO task. The center and right groups show the real-robot *Fold Towel* and *Store Plate* tasks, respectively. The top two rows show FastWAM and the bottom two rows show Enfold. Within each method, the upper row contains rollout observations and the lower row contains predicted future frames. Red boxes mark mismatches in FastWAM predictions, including false object-in-gripper states in LIBERO and *Fold Towel*, and a false plate grasp that leads to an empty-hand handover in *Store Plate*. Green boxes mark the corresponding regions where Enfold predictions remain consistent with the observed future state.

D GENERATOR-STATE PROBE

This appendix specifies the probe used in Figures 2 and 3. The probe is designed to compare candidate supervision targets across generator depth and corruption level. It does not evaluate complete video rollouts.

D.1 SETUP

Data and controls. We probe a post-trained conditional video generator with 28 Transformer blocks (L0–L27). The dataset contains 20 videos from four tabletop manipulation tasks—`fold_towel`, `organize_desktop`, `spoon_powder`, and `store_plate`—with five episodes per task. Each sample contains 33 frames resized to 256×320. Short clips are padded by repeating the final frame, and no temporal resampling is applied.

Every video is evaluated with two noise seeds and three illumination variants: the original clip x, a dark variant $x^{1.45}$, and a bright variant $x^{0.70}$, with pixel values clipped to $[0, 1]$. All samples use the same neutral prompt, `A robot arm operating on a tabletop.`, so task retrieval cannot read the task identity directly from language. The complete probe contains 20 videos $\times$ 3 illumination variants $\times$ 2 seeds $\times$ 6 timesteps $=$ 720 teacher forward passes.

Corruption levels. Videos are encoded into VAE latents and evaluated at six corruption levels:

$$z \in \mathbb{R}^{1\times16\times9\times32\times40}, \qquad t \in \{0, 300, 600, 800, 900, 999\}.$$

At each level, we construct

$$z_{t,s} = (1 - \sigma_t)z + \sigma_t\epsilon_{t,s}, \qquad \sigma_t = t/1000, \qquad \epsilon_{t,s} \sim \mathcal{N}(0, I). \tag{17}$$

The first latent temporal slice is replaced by the clean conditioning frame. Thus, even at $t = 999$, the generator receives one clean frame. Each probe point uses a single teacher evaluation at a fixed timestep, rather than an iterative denoising trajectory.

D.2 Feature Extraction

A teacher pass returns the output of every generator block,

$$H^{t,\ell}_{i,v,s} \in \mathbb{R}^{2880\times 2048}, \tag{18}$$

where i, v, s, t, and ℓ index the video, illumination variant, noise seed, corruption timestep, and generator block. All blocks from the same pass share the same corrupted latent, timestep, and noise sample.

The probe uses the token-mean feature

$$f^{t,\ell}_{i,v,s} = \frac{1}{2880}\sum_{k=1}^{2880} H^{t,\ell}_{i,v,s,k} \in \mathbb{R}^{2048}. \tag{19}$$

This pooling is used only to characterize global feature geometry. Enfold's G2R target retains spatio-temporal tokens. Feature displacement is measured with cosine distance,

$$d(a,b) = 1 - \frac{a^\top b}{\|a\|_2\|b\|_2}. \tag{20}$$

D.3 Metrics

All quantities below are computed independently for each (t,ℓ) pair. Let $\bar{f}_i = \frac{1}{2}\sum_s f_{i,\mathrm{orig},s}$ denote the seed-averaged feature of the original clip.

Input dispersion. The reference scale for representation change is the mean distance between different videos,

$$D_{\mathrm{input}} = \frac{2}{N(N-1)}\sum_{i<j} d(\bar{f}_i, \bar{f}_j), \qquad N = 20. \tag{21}$$

Normalized sensitivities are interpreted only together with D_{input}, since a small ratio is not informative when the feature itself is nearly constant.

Noise sensitivity. We measure the displacement caused by changing only the diffusion noise,

$$D_{\mathrm{seed}} = \frac{1}{N}\sum_i d(f_{i,\mathrm{orig},0}, f_{i,\mathrm{orig},1}), \qquad \rho_{\mathrm{seed}} = \frac{D_{\mathrm{seed}}}{D_{\mathrm{input}}}. \tag{22}$$

Lower ρ_{seed} indicates that the representation is more stable to the sampled generation noise. Numerical distances at $t = 0$ are clipped to zero.

Illumination sensitivity. Using seed-matched original/dark and original/bright pairs, we compute

$$D_{\mathrm{light}} = \frac{1}{2NS}\sum_{i,s}\left[d(f_{i,\mathrm{orig},s}, f_{i,\mathrm{dark},s}) + d(f_{i,\mathrm{orig},s}, f_{i,\mathrm{bright},s})\right], \qquad \rho_{\mathrm{light}} = \frac{D_{\mathrm{light}}}{D_{\mathrm{input}}}, \tag{23}$$

with $S = 2$ seeds. Lower values indicate stronger invariance to the controlled brightness perturbation.

Layout sensitivity. For two `organize_desktop` episodes with different object arrangements, denoted a and b, we use

$$D_{\mathrm{layout}} = \frac{1}{S}\sum_s d(f_{a,\mathrm{orig},s}, f_{b,\mathrm{orig},s}), \qquad \rho_{\mathrm{layout}} = \frac{D_{\mathrm{layout}}}{D_{\mathrm{input}}}. \tag{24}$$

Unlike nuisance sensitivity, layout sensitivity is not minimized: object arrangement can be relevant to manipulation. Figure 3 therefore compares ρ_{layout} with ρ_{light} across depth.

Task retrieval. Each seed-averaged original feature retrieves the other 19 videos by cosine distance, and the four videos from the same task are positives. We report mean average precision over the 20 queries. Because task identity is correlated with objects and scene layout, this metric is referred to as task/scene retrieval rather than action recognition.

Effective rank in the generator-state probe. Let $X \in \mathbb{R}^{20\times 2048}$ contain the centered features $\{\bar{f}_i\}$. If $\{s_k\}$ are its singular values, we define

$$p_k = \frac{s_k^2}{\sum_j s_j^2}, \qquad \mathrm{erank}(X) = \exp\left(-\sum_k p_k \log p_k\right). \tag{25}$$

Effective rank measures how broadly variation is distributed across feature directions. Because X is centered, the value in this 20-video probe is at most 19. The 100-clip comparison in Figure 8 uses the same estimator but a separate matrix X^M_{100}, whose maximum rank is 99. Raw values from the two sample sets are therefore not compared. Within either set, effective rank is interpreted jointly with ρ_{seed}, since stochastic noise can also increase rank.

Aggregation in the figures. Figures 2(a,b) sweep task/scene retrieval and ρ_{seed} over depth and timestep. Panel (c) shows their trade-off at $t = 600$, with marker size indicating effective rank. Figure 3 reports median responses across $t \in \{300, 600, 800, 900\}$ and uses the interquartile range over these four timesteps. These statistics are descriptive and are not used for formal significance testing.

D.4 Scope and Limitations

The probe is intentionally small and controlled. Its 20 videos are sufficient to expose depth- and timestep-dependent trends, but not to establish a precise ranking of layers. Task retrieval is confounded with object and scene identity. The illumination intervention changes global intensity and contrast but does not model shadows, color temperature, or view-dependent reflectance, and the layout response is estimated from one episode pair.

Token averaging suppresses local spatial structure, so a weak pooled response does not imply that a layer lacks useful patch-level information. Effective rank depends on sample count and can be inflated by noise, so we compare it only within a shared evaluation set and interpret it together with seed sensitivity. Finally, the six fixed corruption levels are single forward passes, not a complete denoising trajectory. These probes motivate the multi-level target construction. The end-to-end value of that target is evaluated separately by the LIBERO supervision ablation in Table 4.