VIDEO OR ACTION? DEMONSTRATION MODALITY MATTERS IN EMBODIED IN-CONTEXT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Demonstrations can guide an agent's predict proper actions through In-Context Learning (ICL) without updating model parameters. Yet identical demonstration can be represented in different modalities (video or action) while providing different task information (*e.g.,*in videos, movements can accompany environmental responses). In this work, we study the effectiveness and complementarity of embodied demonstration modalities. Theoretically, we assume that demonstrator and learner responses depend on a shared, unobserved task condition through distinct feature maps. Then, we derive necessary and sufficient conditions for exact learner prediction in the infinite-context limit. These conditions distinguish *insufficient task information* in the demostration from *insufficient representation capacity*: increasing representation width cannot recover task-relevant information absent from the demonstration. Practically, we evaluate policy execution in various 2D and 3D environments as well as future video and action prediction on aligned human–robot recordings. Independently trained modality variants measure the benefit of each input configuration, while fixed-weight interventions test reliance on demonstration content. Our experiments yield three key findings. **(i)** Video alone supports effective execution, and action trajectories can achieve comparable performance when the demonstrator adapts its actions to environmental conditions. **(ii)** Video and measured action trajectories provide complementary cues for human–robot prediction: together they yield lower state and visual errors than either alone. Gains from combining modalities in policy execution vary across tasks. **(iii)** Incorrect demonstrations are more harmful than absent ones, but correct demonstrations can rescue drifting predictions even when supplied after prediction begins, without restarting the rollout or updating model weights. Code is available at anonymous.4open.science/r/anonymous-core-code-CBD2.

1 INTRODUCTION

In embodied environments, a policy maps observations to actions whose consequences depend on object properties and interaction conditions, such as friction and contact-surface orientation, even for the same task (Yu et al., 2017; Kumar et al., 2021). In embodied ICL, the *demonstrator* (source agent) provides an observed interaction as context, while the *learner* (target agent) uses it to guide its own actions or predict its future despite differences in their observation and action spaces (Yu et al., 2018; Dasari & Gupta, 2021; Zakka et al., 2022). In natural language processing, in-context learning (ICL) infers task behavior from inference-time examples without updating model parameters (Brown et al., 2020; Xie et al., 2022; Min et al., 2022), an approach also used in embodied tasks (Duan et al., 2017; Xu et al., 2022; Fu et al., 2024; Vosylius & Johns, 2025).

Human demonstrations provide such context without requiring a corresponding robot trajectory (Yu et al., 2018; Jain et al., 2024). Different demonstration representations retain different information (Figure 1). (i) *Action* trajectories encode measured movements or commands numerically; in embodied ICL, KAT (Di Palo & Johns, 2024) uses them alongside visual keypoints (tracked landmarks). MimicPlay (Wang et al., 2023) and Vid2Robot (Jain et al., 2024) use human video to guide manipulation, while Zero-WAM (Zhou et al., 2026) emphasizes the intermediate states and scene evolution that video conveys. However, imitation success need not imply reliance on task-relevant observations (Wen et al., 2020; Yang et al., 2026). (ii) *Video* demonstration captures both actor motion and environmental response: the same pushing trajectory can produce different object dis-

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.

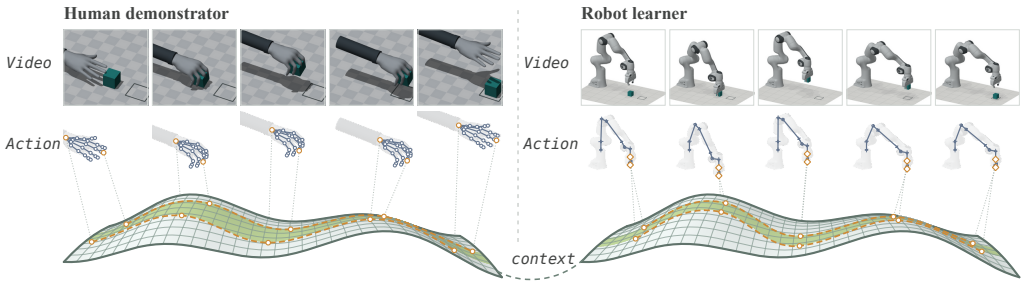


Figure 1: **Modalities difference between video and action under the same demonstration.** Schematic comparison of human (left) and robot (right) interactions through scene video, joint motion (Action), and keypoint trajectories. Each side depicts the same sequence across views; orange diamonds mark tracked points.

placements under different friction, revealing conditions absent from the trajectory alone. Conversely, a demonstrator that adjusts its actions to achieve a goal can reveal environmental conditions through those actions. The comparison therefore depends on how demonstrations are generated as well as how they are represented. It remains unclear when each modality provides information needed for the learner's own response, and when combining modalities offers additional benefit.

In this work, we study this question through a theoretical characterization of cross-embodiment in-context prediction. We model demonstrator and learner responses using distinct feature maps of their observations and a shared, unobserved task condition. Building on the population analysis of Kim & Suzuki (2024) and related linear-attention analyses (Zhang et al., 2024; Mahankali et al., 2024), we derive necessary and sufficient conditions for exact learner prediction in the infinite-context limit of a simplified linear-attention model. The result distinguishes *missing source information* from *insufficient representation capacity*: increasing representation width cannot recover learner-relevant information omitted by the source features.

To validate these findings in practice, we conduct policy execution experiments in two-dimensional (2D) tasks (throwing, planar rebound, and friction pushing) and three-dimensional (3D) tasks in the MuJoCo physics simulator (Todorov et al., 2012), using video- and trajectory-conditioned inputs (Jain et al., 2024; Fu et al., 2024). The 3D tasks involve passing through an opening (aerial gate), bouncing off a tilted surface (tilted bank), and launching from a ramp (ramp-to-air). We also study future video and state prediction on temporally aligned human–robot recordings. Independently trained modality variants test the benefit of each input configuration, while fixed-weight interventions test reliance on the supplied context. Our results highlight three findings. **(i) Actions can also reveal environmental conditions.** Fixed demonstrations use preset commands independent of hidden physical parameters; video records the resulting object responses and outperforms action inputs in these controls. Adaptive demonstrations use commands chosen to reach a goal under those parameters, allowing the commands themselves to convey task-relevant information. Action-only execution then improves substantially, reaching performance comparable to video on in-distribution (ID) tests across the three 2D tasks. **(ii) Video and action improve prediction individually and together.** On the evaluated human–robot recordings, video and measured hand trajectories each improve future prediction, and combining them yields lower state and visual prediction errors than either modality alone (Table 3). The paired 3D execution tests show that the additional benefit of combining modalities varies across tasks. **(iii) Demonstration interventions affect prediction, and late-arriving demonstrations can correct drift.** At fixed weights, removing demonstrations increases human–robot prediction error, while incorrect demonstrations cause larger errors. Introducing correct demonstrations during an ongoing rollout can recover already drifting predictions without restarting the rollout.

2 Related Work

Embodied In-Context Learning. One-shot imitation and visual meta-learning acquire tasks from demonstrations (Duan et al., 2017; Finn et al., 2017; Yu et al., 2018). Table 1 marks supplied demonstration inputs (✓) and absent inputs (✗); Action denotes demonstrator commands or measured

body trajectories. Other context includes block positions and gripper opening in One-shot IL, positions, velocities, and cumulative future rewards in Prompt-DT (Xu et al., 2022), 3D keypoints in KAT (Di Palo & Johns, 2024), and point clouds in Instant Policy (Vosylius & Johns, 2025). Video interfaces include goal frames in MimicPlay and full demonstrations (Dasari & Gupta, 2021; Wang et al., 2023; Jain et al., 2024; Zhou et al., 2026). ICRT combines images, gripper pose/opening, and actions (Fu et al., 2024); Instant Policy also uses gripper pose/opening. ICWM pairs images around exploratory actions (Wang et al., 2026), while GEN-1.5 combines video, sensors, and actions (Generalist Team, 2026). XIRL aligns task progress across agents with different bodies (Zakka et al., 2022). We compare demonstration modalities through separate retraining and fixed-weight interventions, informed by studies of demonstration labels (Min et al., 2022) and input removal (Hooker et al., 2019).

Table 1: **Demonstration modalities in representative embodied learning methods.**

Work	Main task	Demonstration		
		Source	Video (Images)	Action
One-shot IL (NIPS 2017)	Block stacking	Simulated	✗	✗
Prompt-DT (ICML 2022)	Locomotion + reaching	Simulated	✗	✓
MimicPlay (CoRL 2023)	Manipulation	Human / robot	✓	✗
Vid2Robot (RSS 2024)	Manipulation	Human / robot	✓	✗
KAT (RSS 2024)	Manipulation	Robot	✗	✓
ICRT (ICRA 2025)	Manipulation	Robot	✓	✓
Instant Policy (ICLR 2025)	Manipulation	Human / robot	✗	✓
ICWM (arXiv 2026)	Manipulation	Robot	✓	✓
GEN-1.5 (Blog 2026)	Manipulation	Human / robot	✓	✓
Zero-WAM (arXiv 2026)	Manipulation	Human	✓	✗

World-Action Models and Physical Adaptation. Diffusion Policy generates actions (Chi et al., 2025), whereas DreamZero jointly predicts visual futures and actions (Ye et al., 2026). Our predictor generates video and robot states (end-effector position, orientation, and gripper state), adapting Faster-WAM (Zhao et al., 2026) and shared attention across modality-specific experts (separate processing branches) (Liang et al., 2025) to study demonstration use. Online system identification and Rapid Motor Adaptation infer dynamics from the robot's experience (Yu et al., 2017; Kumar et al., 2021); ICWM uses self-generated interactions without parameter updates (Wang et al., 2026). We examine another demonstrator's context, with execution evaluated separately through simulated policies.

Task Inference from Demonstrations. Bayesian accounts interpret demonstrations as evidence about a shared, unobserved task (Xie et al., 2022). Other analyses study function prediction (Garg et al., 2022), regression (Akyürek et al., 2023; von Oswald et al., 2023), and autoregressive processes (Sander et al., 2024). Extending the population model of Kim & Suzuki (2024), we characterize when demonstrator features contain sufficient information for a learner with a different response, and how representation width constrains prediction.

3 Demonstration Modalities in Embodied ICL

3.1 From Language ICL to Embodied ICL

In the formulation of Garg et al. (2022), language ICL examples and queries follow the same task function. In embodied ICL, the demonstrator and learner share task conditions but may respond differently because their bodies differ (Jain et al., 2024). For example, a human push reveals how an object moves under friction; predicting a robot's response also requires the robot's own state. We ask which demonstration modalities help the learner predict its own future at fixed model weights.

We use h and r for the demonstrator and learner. A demonstration $H = (X^h, Y^h)$ pairs an initial observation X^h with its observed response Y^h. Given H and its current observation Q (the query), the learner predicts its future response Y:

$$P = (H, Q) \quad \longrightarrow \quad \widehat{Y},$$

where P is the input and $\widehat{Y}$ is the prediction. In human–robot recordings, Y^h contains human video and measured hand trajectories; Y contains future robot video and states (end-effector pose and gripper state).

A shared, unobserved task or scene condition $\tau \in \mathcal{T}$ determines a pair of response laws:

$$\mathcal{F}_{hr} = \{(f_\tau^h, f_\tau^r) : \tau \in \mathcal{T}\}, \qquad f_\tau^e : \mathcal{X}_e \to \mathcal{P}(\mathcal{Y}_e). \tag{1}$$

For $e \in \{h, r\}$, f_τ^e maps observations in $\mathcal{X}_e$ to distributions over responses in $\mathcal{Y}_e$; $\mathcal{P}(\mathcal{Y}_e)$ denotes those distributions. The learner observes H and Q, not τ.

3.2 Demonstration Modalities and Task Information

Video V^h records the demonstrator's movement and the environmental response. Action trajectories M^h (*Action*) describe actions or measured movement; in human recordings, they contain measured hand trajectories. We consider four contexts, paired with the same learner query Q:

$$\begin{gathered} H_{\mathrm{V}} = V^h, \quad H_{\mathrm{A}} = M^h, \quad H_{\mathrm{VA}} = (V^h, M^h), \quad H_\varnothing = \varnothing, \\ P_\star = (H_\star, Q) \longmapsto \widehat{Y}_\star, \qquad \star \in \{\mathrm{V}, \mathrm{A}, \mathrm{VA}, \varnothing\}. \end{gathered} \tag{2}$$

A modality is informative when it helps distinguish task conditions τ that require different learner responses Y. Preset actions may be identical across such conditions, while video shows different environmental responses. Actions adapted to those conditions may themselves reveal task information. We ask when these contexts are effective or complementary; the following analysis characterizes information and representation requirements for exact learner prediction.

3.3 Theoretical Analysis

We study when demonstrations allow exact prediction of a learner with different responses, and how the required representation width depends on those differences.

Assumptions for the population analysis

Tasks and responses. Inputs $x \in \mathcal{X} \subset \mathbb{R}^d$ follow $\mathbb{P}$, with input dimension d and task-feature dimension k. Fixed feature maps $f, g : \mathcal{X} \to \mathbb{R}^k$ describe the demonstrator and learner, with $\mathbb{E}[\|x\|^2 + \|f(x)\|^2 + \|g(x)\|^2] < \infty$. For task $\tau = \beta \in \mathbb{R}^k$ in Equation (1), their responses are

$$y_i^h = \beta^\top f(x_i), \qquad y_q^r = \beta^\top g(x_q), \qquad \mathbb{E}[\beta\beta^\top] = I_k. \tag{3}$$

Here I_k is the $k \times k$ identity matrix. **Sampling.** The n demonstration inputs $x_1, \ldots, x_n$ and query x_q are drawn independently from $\mathbb{P}$, independently of β. Thus $H = ((x_i, y_i^h))_{i=1}^n$, $Q = x_q$, and the prediction target is $Y = y_q^r$.

Learner. We use an m-dimensional bias-free two-layer representation $h_\mu(x) = \int a\sigma(w^\top x)\, d\mu(a, w)$, followed by linear attention $W \in \mathbb{R}^{m \times m}$ (Kim & Suzuki, 2024). Here μ is a probability measure on $(a, w) \in \mathbb{R}^m \times \mathbb{R}^d$ and $\sigma(t) = \max(t, 0)$ is ReLU (rectified linear unit). We analyze the infinite-context limit $n \to \infty$.

Write $h = h_\mu$ and $A_h = \mathbb{E}_{x \sim \mathbb{P}}[f h^\top]$, the cross-moment of demonstrator features and the representation. Linear attention forms a weighted sum of demonstration responses. Its prediction and limiting squared-error risk, averaged over tasks, are

$$\widehat{y}_{q,n}^r = \frac{1}{n} \sum_{i=1}^n y_i^h h_\mu(x_i)^\top W h_\mu(x_q), \qquad \mathcal{L}_{\mathrm{TF}}(\mu, W) = \frac{1}{2} \mathbb{E}_{x \sim \mathbb{P}} \|g - A_{h_\mu} W h_\mu\|^2. \tag{4}$$

Let $\mathcal{B}$ be the scalar bias-free ReLU Barron space: functions expressible as ReLU mixtures with finite $\int |a| \|w\|\, d\nu$, where ν is a probability measure on $\mathbb{R} \times \mathbb{R}^d$. Membership in $\mathcal{B}^k$ means that each coordinate belongs to $\mathcal{B}$. We extend the matched case $f = g$ of Kim & Suzuki (2024) to distinct demonstrator and learner features.

Proposition 1 (Heterogeneous realizability). *With $\mathbb{E}_x$ denoting expectation over $x \sim \mathbb{P}$, let $K_g = \mathbb{E}_x[gg^\top]$ be the learner second moment, $C_{fg} = \mathbb{E}_x[fg^\top]$ the source–learner cross-moment, and $\mathcal{D}_f = \{\mathbb{E}_x[f\phi] : \phi \in \mathcal{B}\}$ the source-accessible directions. Then $\mathcal{L}_{\mathrm{TF}}(\mu, W) = 0$ admits a representation with $\operatorname{ess\,sup}_\mu \|a\|\|w\| < \infty$ and some W if and only if*

$$g \in \mathcal{B}^k, \qquad \mathcal{R}(K_g) \subseteq \mathcal{D}_f, \\ \dim\big(\mathcal{R}(C_{fg}) \cap \mathcal{R}(K_g)\big) \geq 2 \dim \mathcal{R}(K_g) - m, \tag{5}$$

where $\mathcal{R}(M)$ is the column space of M, and $\operatorname{ess\,sup}_\mu$ is the supremum after ignoring sets of zero μ-measure.

Exact prediction requires learner features that the model can represent and task directions that the demonstration makes accessible. When these conditions hold, less overlap between the moment spaces requires greater width at fixed learner rank. Increasing width cannot recover an inaccessible task direction. For demonstration modalities, the relevant question is whether the input distinguishes conditions requiring different learner responses: identical action trajectories may hide differences visible in object responses. Definitions, the proof, and explicit width examples appear in Appendix C.

4 Experiments

We study three questions about demonstration use in embodied ICL:

- **Q1: How does demonstration affect the relative effectiveness of video and action trajectories?**
- **Q2: How do video and action trajectories help, individually and together?**
- **Q3: How do demonstration interventions affect the learner, and can late-arriving demonstrations restore prediction accuracy?**

We evaluate demonstration modalities across diverse settings: policy execution in 2D and 3D simulations, and joint prediction of future robot states and video from human–robot recordings.

4.1 Experimental Settings

Controllable experiments. We evaluate throwing, planar rebound, and friction pushing in 2D, and three custom MuJoCo (Todorov et al., 2012) tasks in 3D: aerial gate, tilted bank, and ramp-to-air. Fixed-probe demonstrations separate issued commands or tracked trajectories (Action) from observed object responses (Video). Fixed probes use preset commands independent of hidden physics; adaptive demonstrations use commands chosen to reach a goal under that physics. We compare these demonstrations with matched learner tasks and training budgets, testing in-distribution (ID) and out-of-distribution (OOD) conditions with learner geometry outside training ranges (Table 2). Additional curling interventions test context relevance across scene styles and movement directions.

Human–robot prediction. Given the robot's current image and state, a Transformer with approximately 0.6 billion trainable parameters (0.6B) predicts 32 future states and video, conditioned on human video, measured hand trajectories, or both from the same aligned episode. We evaluate translation average displacement error (ADE) and visual quality on 50 matched test windows, each a contiguous 32-step segment from an episode (one recorded interaction). Hand–arm video retains hand, forearm, and sleeve regions while masking the surrounding scene.

Training and comparisons. We independently train None (no demonstration), Action (action trajectories), Video, and Full (video + action trajectories), retaining the learner query. Separate interventions remove, replace, or delay demonstrations at fixed weights, reusing the same queries and random generation initialization across conditions. Correct supplies task-matched demonstrations, Absent removes them, and Incorrect supplies task-mismatched demonstrations; *No demo* in figures means Absent. Training budgets, supervision, and evaluation protocols are detailed in Appendices A and B.

4.2 Controllable Experiments: Execution and Response Information

We compare Action and Video under fixed and adaptive demonstrations (Q1), assess their combination (Q2), and test context relevance (Q3). Table 2(a,b) reports all six tasks shown in Figure 2; parts (c,d)

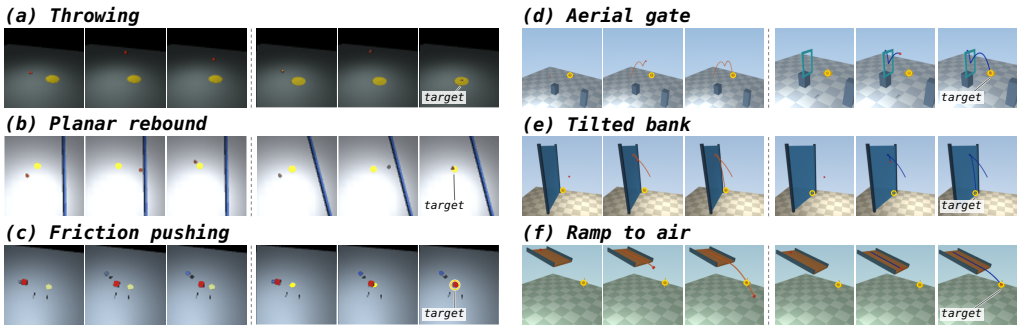


Figure 2: **Demonstrations and executions across six controllable tasks.** Panels (a)–(c) match the three 2D tasks in Table 2(a), and panels (d)–(f) match its three 3D tasks in part (b). Each panel shows three selected fixed-probe demonstration frames and three Video-policy execution frames from the first test record; target labels mark the learner's goal. Adaptive-policy results for these same six learner tasks appear in Table 2(c,d); the visualized demonstrations use fixed probes.

Table 2: **Execution success with fixed and adaptive demonstrations (%, ↑).** Fixed actions are independent of hidden physics; adaptive actions target a goal under that physics. Results average three training seeds in (a,b) and throwing in (c); other results use one training seed. Matched fixed/adaptive 3D comparisons with shared inputs and supervision are in Appendix B.3.1. OOD shifts learner geometry. Checks/crosses denote input presence/absence; bold/underline mark the best/second-best result per column.

(a) Fixed: 2D tasks

Video	Action	Throwing		Planar rebound		Friction pushing	
		ID	OOD	ID	OOD	ID	OOD
✗	✗	29.85	21.53	49.56	15.40	33.66	18.10
✗	✓	29.65	21.84	48.96	17.07	33.92	18.46
✓	✗	**96.55**	**95.65**	**93.28**	26.92	**97.14**	**83.68**
✓	✓	91.83	89.58	92.72	**27.20**	96.65	81.58

(b) Fixed: 3D tasks

Video	Action	Aerial gate		Tilted bank		Ramp to air	
		ID	OOD	ID	OOD	ID	OOD
✗	✗	49.35	48.83	45.25	11.46	61.88	64.88
✗	✓	49.35	48.76	45.70	11.56	62.30	64.68
✓	✗	68.10	67.12	65.53	15.10	99.51	67.84
✓	✓	**68.26**	**68.13**	**74.84**	**19.63**	**99.54**	**67.90**

(c) Adaptive: 2D tasks

Video	Action	Throwing		Planar rebound		Friction pushing	
		ID	OOD	ID	OOD	ID	OOD
✗	✗	29.85	21.53	49.07	14.94	32.96	17.68
✗	✓	99.89	99.66	**92.92**	**33.94**	**98.63**	88.57
✓	✗	99.69	99.07	92.48	25.78	98.24	85.30
✓	✓	**100.00**	**100.00**	91.99	28.32	98.19	**88.67**

(d) Adaptive: 3D tasks

Video	Action	Aerial gate		Tilted bank		Ramp to air	
		ID	OOD	ID	OOD	ID	OOD
✗	✗	49.32	48.34	43.75	10.35	62.11	63.96
✗	✓	73.83	72.27	**83.01**	**21.48**	99.32	68.95
✓	✗	45.70	45.51	81.05	21.09	97.56	68.36
✓	✓	**78.91**	**75.88**	82.91	19.73	**99.51**	**69.53**

report adaptive demonstrations for the same six tasks. Protocols and matched 3D comparisons appear in Appendix B.3.1.

Q1: Actions can also reveal environmental conditions. In the fixed-probe controls, Video outperforms Action across all six tasks, whose Action inputs omit object responses (Table 2(a,b)). Adaptive demonstrations substantially improve Action: throwing ID success rises from 29.65% to 99.89%, and Action attains 92.92% and 98.63% ID success in rebound and pushing, respectively (part c).

Adaptive demonstrations also improve Action ID success against matched fixed controls in 3D (Appendix B.3.1); Table 2(d) reports the adaptive results. In aerial gate, adaptive demonstrations

Table 3: **Prediction accuracy of independently trained models.** Each row is trained with the indicated inputs. Means on 50 matched test windows with paired generation noise. Video columns indicate retained hand–arm and surrounding scene regions.

Video		Action	ADE (mm)↓	FDE (mm)↓	LPIPS↓
Hand–arm	Scene				
✗	✗	✗	28.85	46.53	0.0865
✗	✗	✓	14.93	20.80	0.0676
✓	✗	✗	13.05	17.20	0.0613
✓	✓	✗	11.91	14.98	0.0604
✓	✓	✓	**10.61**	**14.24**	**0.0582**

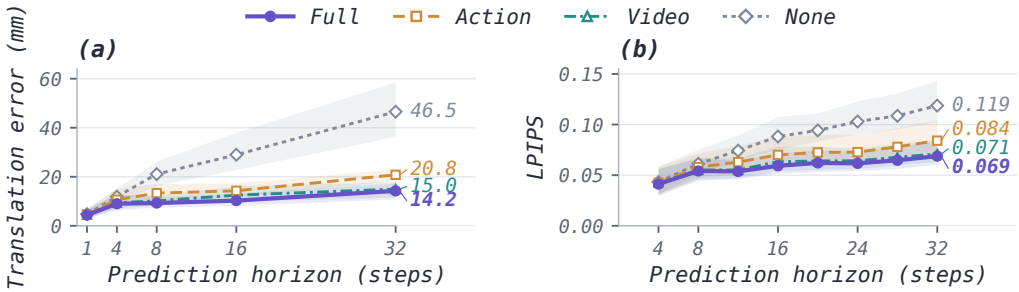


Figure 3: **The context advantage grows with horizon.** (a) Translation error and (b) LPIPS for four independently trained models on 50 matched test windows. Lines show means; bands show pointwise 95% confidence intervals from resampling instruction groups (windows sharing a task instruction).

improve Action while reducing Video performance. Every model in (d) receives the source initial state, goal, geometry, and mass; the demonstrator uses hidden dynamics to choose its actions.

Q2: Combining modalities helps selectively. In the fixed-probe controls (Table 2(a,b)), adding actions did not consistently improve on video in 2D. In 3D, tilted bank showed the clearest gain under both ID and OOD conditions; gains in aerial gate and ramp to air were small.

Q3: Execution depends on demonstration relevance. At fixed Video weights, matching demonstrations improved curling execution, whereas incorrect-physics video eliminated success (Appendix B.2).

4.3 Effect of Demonstration Modality on Prediction

We compare the predictive value and complementarity of demonstration modalities in aligned human–robot interactions (Q2), using separately trained predictors. Final displacement error (FDE) is the position error at the final predicted step. Learned perceptual image patch similarity (LPIPS) measures image difference using deep features (Zhang et al., 2018); lower values are better.

Modality effectiveness and complementarity (Q2). Video outperformed Action, and combining both gave the lowest trajectory errors (Table 3). Hand–arm video improved prediction over Action alone; retaining the surrounding scene further reduced error. Numerical action trajectories provided an additional benefit with complete video. Appendix B.1.1 describes the video masking procedure.

Context matters more as prediction extends beyond the current observation. As the prediction horizon (number of future steps) increased, Full and None had similar early translation errors but diverged later (Figure 3): the gap grew from 0.34 mm at step 1 to 32.29 mm at step 32. Demonstration context thus provided a larger benefit farther into the predicted continuation.

Evaluation. The five independently trained predictors share architecture and training budget, retain the current robot image and state, and receive no language. ADE averages position error over 32 steps. Additional visual metrics are the structural similarity index (SSIM) (Wang et al., 2004) and peak signal-to-noise ratio (PSNR, in decibels); higher values indicate closer image agreement. Supplementary diagnostics report mean squared error (MSE) for states and video latents, the compressed video features reconstructed into frames by the frozen variational autoencoder (VAE) (Wan-Video, 2025). Masking is applied during both training and evaluation. Visual metrics, masking details, and evaluation settings are given in Appendix B.1.1.

4.4 Effects of Demonstration Changes

We test demonstration changes (Q3) at fixed predictor weights, query inputs, and generation noise. Content replacement and video-prefix removal use the Full predictor; delayed arrival also tests the independently trained Action and Video predictors.

Context relevance. Replacing both demonstration modalities with a different-instruction recording increased state and visual prediction errors more than removing both (Table 9). A task-mismatched demonstration can thus misdirect the learner. The value of demonstration context depends on its relevance as well as its modality.

Interactions between modalities. Coupled future attention lets future-video and future-state features read each other; decoupled attention blocks both directions while retaining access to demonstrations. For the variants trained with 50% complete context, correct demonstrations reduced ADE with both attention configurations on the same 50 test windows. The ADE reduction from correct context and increase from incorrect context, each relative to absent context, were larger with coupling (Table 6). At fixed Full weights, retaining only an initial portion of the demonstration video (its prefix) increased prediction error with Action retained (Table 10). Video predictions decoded from latent features showed corresponding changes in object placement (Figure 4).

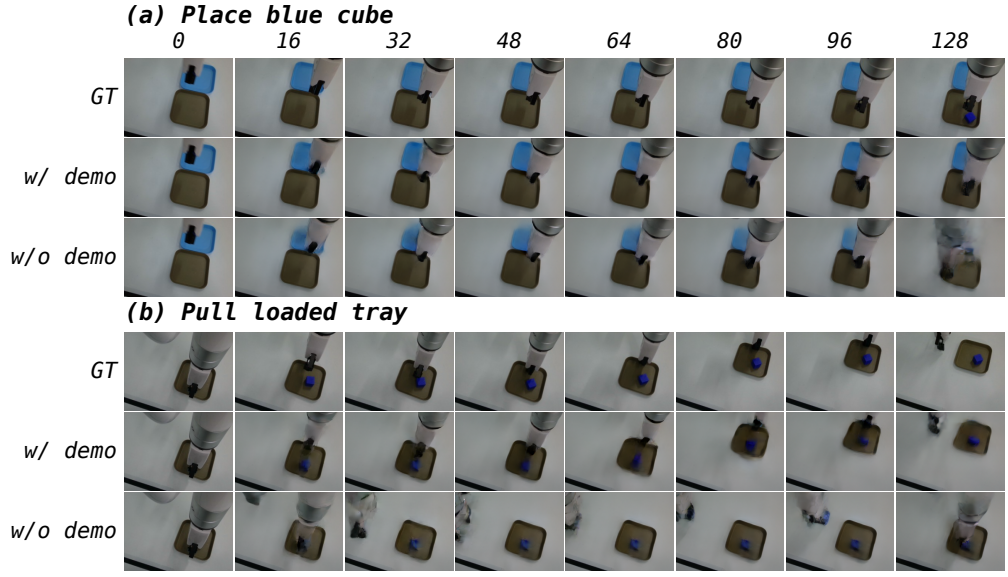


Figure 4: **Decoded robot futures with and without the demonstration.** Selected cube-placement and tray-pulling predictions. Columns span steps 0–128; rows show ground truth (GT) and predictions from the same fixed-weight model with both demonstration video and action inputs (w/ demo), or with both demonstration inputs masked (w/o demo).

Demonstration intervention and prediction recovery. After 8 or 32 predicted steps without demonstrations, we supply each predictor's trained input: Action, Video, or both. Each independently trained predictor has a no-arrival control with fixed weights and paired noise. Prediction continues

from the predicted frame and state. Delayed access reduced position error and improved recovery across all predictors and both horizons (Figure 5). Recovery requires eight consecutive steps within 20 mm for windows exceeding this error at arrival; stability uses the latest eight steps of all windows. Decoded futures show joint arrival (Figure 6); protocols are in Appendix B.1.1.

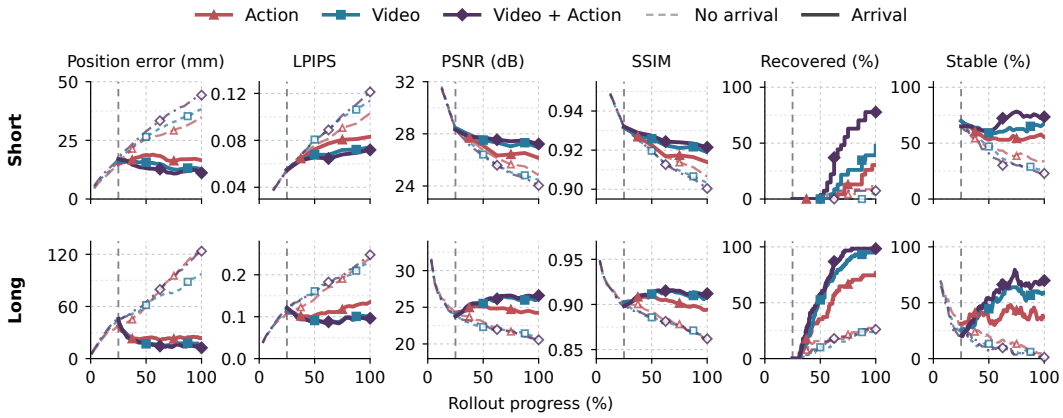


Figure 5: **Prediction with late-arriving demonstrations.** Colors identify independently trained predictors; solid/dashed curves show arrival/no arrival at the same weights. Arrival follows 8 of 32 steps (short) or 32 of 128 steps (long). Means over the same 83 windows from 42 episodes.

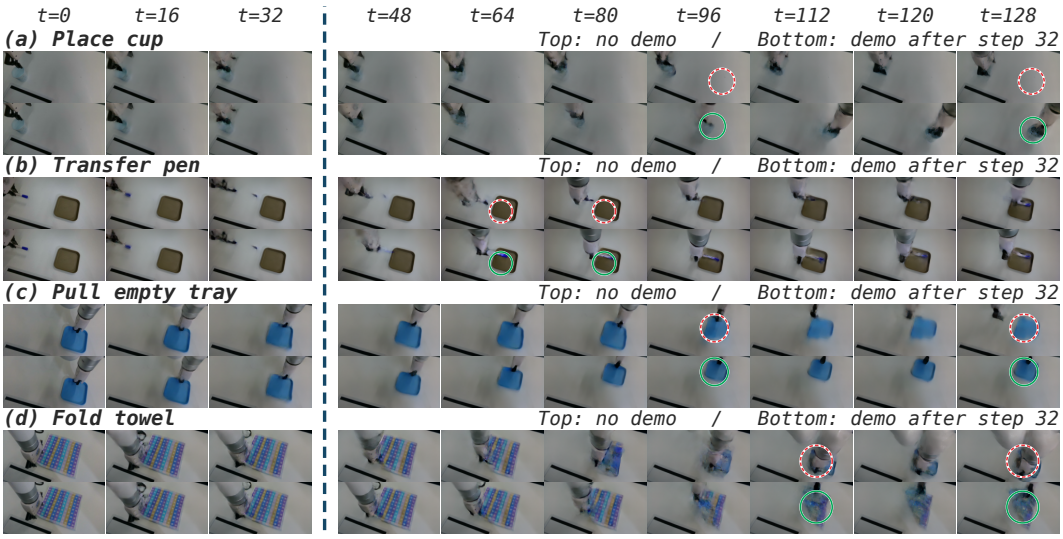


Figure 6: **Demonstration arrival changes selected continuations.** The divider marks step 32; paired rows share the same prefix and noise.

5 Conclusion

We studied the effectiveness and complementarity of video and action trajectories in embodied in-context learning. Our theory distinguishes missing demonstration information from insufficient representation capacity. Experiments show that video supports effective execution, while action trajectories become more informative when demonstrators adapt their actions to environmental conditions. On human–robot recordings, both modalities improve future prediction, and combining them further reduces state and visual errors. Their combined benefit in execution varies across tasks. At fixed weights, incorrect demonstrations can be more harmful than missing ones, whereas correct demonstrations introduced during a rollout can recover drifting predictions without restarting it.

REPRODUCIBILITY STATEMENT

Appendices A and B.1.1 describe the models, data, training, and evaluation protocols. Raw code is available to ensure reproducibility.

AI USE STATEMENT

AI assistance was used for writing, reference checks, and plotting code. Experimental figures use recorded results or simulated trajectories.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What Learning Algorithm Is In-Context Learning? Investigations with Linear Models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.15661.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *The International Journal of Robotics Research*, 2025. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Sudeep Dasari and Abhinav Gupta. Transformers for One-Shot Visual Imitation. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155, pp. 2071–2084, 2021. URL https://proceedings.mlr.press/v155/dasari21a.html.

Norman Di Palo and Edward Johns. Keypoint Action Tokens Enable In-Context Imitation Learning in Robotics. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.096. URL https://www.roboticsproceedings.org/rss20/p096.html.

Yan Duan, Marcin Andrychowicz, Bradly C. Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-Shot Imitation Learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.07326.

Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-Shot Visual Imitation Learning via Meta-Learning. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78, pp. 357–368, 2017. URL https://proceedings.mlr.press/v78/finn17a.html.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-Context Imitation Learning via Next-Token Prediction. *arXiv preprint arXiv:2408.15980*, 2024. URL https://arxiv.org/abs/2408.15980.

Shivam Garg, Dimitris Tsipras, Percy S. Liang, and Gregory Valiant. What Can Transformers Learn In-Context? A Case Study of Simple Function Classes. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30583–30598, 2022. doi: 10.52202/068431-2217. URL https://papers.nips.cc/paper_files/paper/2022/hash/c529dba08a146ea8d6cf715ae8930cbe-Abstract-Conference.html.

Generalist Team. GEN-1.5: Embodied Foundation Models are One-Shot Learners. Generalist AI technical blog, August 2026. URL https://generalistai.com/blog/gen-1.5.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A Benchmark for Interpretability Methods in Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1806.10758.

Vidhi Jain, Maria Attarian, Nikhil J. Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R. Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2Robot: End-to-end Video-conditioned Policy Learning with Cross-Attention Transformers. *arXiv preprint arXiv:2403.12943*, 2024. URL https://arxiv.org/abs/2403.12943.

Juno Kim and Taiji Suzuki. Transformers learn nonlinear features in context: Nonconvex mean-field dynamics on the attention landscape. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 24527–24561, 2024. URL https://proceedings.mlr.press/v235/kim24af.html.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid Motor Adaptation for Legged Robots. In *Proceedings of Robotics: Science and Systems*, 2021. URL https://www.roboticsproceedings.org/rss17/p011.html.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-Transformers: A Sparse and Scalable Architecture for Multi-Modal Foundation Models. *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2411.04996.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Arvind Mahankali, Tatsunori B. Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.03576.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, 2022. URL https://aclanthology.org/2022.emnlp-main.759/.

Michael Eli Sander, Raja Giryes, Taiji Suzuki, Mathieu Blondel, and Gabriel Peyré. How Do Transformers Perform In-Context Autoregressive Learning? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 43235–43254, 2024. URL https://proceedings.mlr.press/v235/sander24a.html.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A Physics Engine for Model-Based Control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109. URL https://doi.org/10.1109/IROS.2012.6386109.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers Learn In-Context by Gradient Descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 35151–35174, 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Vitalis Vosylius and Edward Johns. Instant Policy: In-Context Imitation Learning via Graph Diffusion. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=je3GZissZc.

Wan-Video. Wan2.2. Software repository, 2025. URL https://github.com/Wan-Video/Wan2.2.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. MimicPlay: Long-Horizon Imitation Learning by Watching Human Play. *arXiv preprint arXiv:2302.12422*, 2023. URL https://arxiv.org/abs/2302.12422.

Siyin Wang, Junhao Shi, Senyu Fei, Zhaoyang Fu, Li Ji, Jingjing Gong, and Xipeng Qiu. In-Context World Modeling for Robotic Control. *arXiv preprint arXiv:2606.26025*, 2026. URL https://arxiv.org/abs/2606.26025.

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. doi: 10.1109/TIP.2003.819861. URL https://www.cns.nyu.edu/pub/lcv/wang03-reprint.pdf.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting Copycat Agents in Behavioral Cloning from Observation Histories. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1b113258af3968aaf3969ca67e744ff8-Abstract.html.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.02080.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua B. Tenenbaum, and Chuang Gan. Prompting Decision Transformer for Few-Shot Policy Generalization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24631–24645, 2022. URL https://proceedings.mlr.press/v162/xu22g.html.

Songhua Yang, Ziyu Liu, Xuetao Li, Ruqi Xiao, Kangxin Zhu, and Miao Li. ICI-VLA: In-Context Imitation with Spatiotemporally Aligned Demonstrations for Vision-Language-Action Models. *arXiv preprint arXiv:2609.07581*, 2026. URL https://arxiv.org/abs/2609.07581.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World Action Models are Zero-shot Policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Tianhe Yu, Chelsea Finn, Sudeep Dasari, Annie Xie, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.002. URL https://www.roboticsproceedings.org/rss14/p02.html.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the Unknown: Learning a Universal Policy with Online System Identification. In *Proceedings of Robotics: Science and Systems*, 2017. URL https://arxiv.org/abs/1702.02453.

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. XIRL: Cross-embodiment Inverse Reinforcement Learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, pp. 537–546, 2022. URL https://proceedings.mlr.press/v164/zakka22a.html.

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*, 2018. URL https://arxiv.org/abs/1801.03924.

Ruiqi Zhang, Spencer Frei, and Peter L. Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024. URL https://www.jmlr.org/papers/v25/23-1042.html.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-WAM: Efficient Inference-Time Future Conditioning for Robust World Action Models. *arXiv preprint arXiv:2608.04404*, 2026. URL https://arxiv.org/abs/2608.04404.

Jiaming Zhou, Qihang Zhang, Gangwei Xu, Cunxin Fan, Yujie Zhao, Ruilin Wang, Yiming Luo, Shuai Yang, Xing Zhu, Yujun Shen, Junwei Liang, and Yinghao Xu. Zero-WAM: In-Context World-Action Modeling from Human Videos for Open-Ended Task Generalization. *arXiv preprint arXiv:2608.26103*, 2026. URL https://arxiv.org/abs/2608.26103.

A Implementation and dataset details

A.1 Inputs and temporal alignment

Human and robot streams share a time axis. Video contains frames at steps $0, 4, \ldots, 32$, at 224×384 resolution (height $\times$ width). Action trajectories and robot states cover steps $1, \ldots, 32$. Each robot state contains a 6D end-effector pose and one gripper value. The 84D human input concatenates four 3D hand-frame vectors and 24 3D hand coordinates in stored order. We normalize human and robot inputs separately, using per-coordinate training means and standard deviations with a 10^{-4} floor on the standard deviations.

Noise conditioning and attention. Let $\sigma \in [0, 1]$ denote the noise level, distinct from the ReLU function in the theory. Each expert encodes $t = 1000\sigma$ with a 256D sinusoidal embedding and a two-layer multilayer perceptron (MLP) with a sigmoid linear unit (SiLU) activation. A SiLU–Linear projection produces shift, scale, and residual-gate vectors for both the attention and feed-forward branches. These six vectors combine with learned modulation parameters in each block. Clean context uses $t = 0$; future video and state tokens use their respective noise levels.

Video queries and keys use temporal–height–width rotary positional embeddings (RoPE); action-trajectory/state tokens use one-dimensional temporal RoPE. Aligned source and target times share temporal indices. Learned role embeddings distinguish token roles, and masking leaves positions unchanged. Each expert projects its queries, keys, and values into the common $128 \times$ number of heads attention width, then projects the output back to its own residual width.

Inference solver. We use 20 explicit Euler steps for both streams. Let x_j concatenate the noisy future-video and future-state tokens at solver step j, and v_θ their joint predicted flow velocity. With $u_j = 1 - j/20$ and $\sigma_j = 5u_j/(1 + 4u_j)$, $j = 0, \ldots, 20$, the update is $x_{j+1} = x_j + (\sigma_{j+1} - \sigma_j)v_\theta(x_j, 1000\sigma_j; P)$ for $j = 0, \ldots, 19$. Both future streams start from Gaussian noise and are updated jointly; the observed context is held fixed throughout integration.

A.2 Recorded-model representation and objective

A frozen Wan2.2 variational autoencoder (VAE) (Wan-Video, 2025) encodes frames at offsets $0, 4, \ldots, 32$ as three latents $z_j^e \in \mathbb{R}^{48\times14\times24}$, $j \in \{0, 1, 2\}$. The current observation is z_0^e and the future continuation is $Z_{1:2}^e = (z_1^e, z_2^e)$. The temporal blocks cover offsets 0, 4–16, and 20–32; later latents also retain earlier information. The demonstrator's measured hand trajectories $M^h \in \mathbb{R}^{32\times84}$ and future robot states $S^r \in \mathbb{R}^{32\times7}$ use frame-step indices; a colon selects an inclusive index range. The current robot state is $s_0^r \in \mathbb{R}^7$. The recorded context, query, and prediction target are

$$\begin{aligned} &H = (X^h, Y^h) = (z_0^h, (Z_{1:2}^h, M_{1:32}^h)), \\ &Q = (z_0^r, s_0^r), \qquad Y = (Z_{1:2}^r, S_{1:32}^r), \qquad P = (H, Q). \end{aligned} \tag{6}$$

An intervention ι selects, removes, or replaces demonstration content, giving $P_\iota = (\iota(H), Q)$ and $\widehat{Y} = G_\theta(P_\iota; \xi)$. Its expected prediction error is $R(\iota; \theta) = \mathbb{E}[\ell(\widehat{Y}, Y)]$, averaged over tasks, paired data, noise, and replacement sampling. Interventions fix weights θ, query Q, and initial noise ξ.

Demonstration video V^h is encoded as $(z_0^h, Z_{1:2}^h)$. Video and state experts use separate projections and feed-forward layers with shared attention; 2×2 video patches give 504 tokens, alongside 65 state/action-trajectory tokens. Let C_ι index all query Q tokens and visible demonstration tokens, and let T index the future tokens. The hidden-token diagonal is $\Delta_\iota = \{(k, k) : k \notin C_\iota \cup T\}$. For query-token index i and key-token index j, the binary attention permission is

$$A_{ij}(\iota) = \mathbf{1}\{(i, j) \in (C_\iota \times C_\iota) \cup [T \times (C_\iota \cup T)] \cup \Delta_\iota\}. \tag{7}$$

Here $\mathbf{1}\{\cdot\}$ equals one when its condition holds and zero otherwise. Future tokens attend to visible context and one another; visible context tokens attend within context. Hidden tokens retain their positions and attend only to themselves. Section 4.4 ablates only the cross-stream future attention. During Full-model training, complete context has probability 0.9; the other seven joint visibility patterns share probability 0.1. Binary visibility flags b_0, b_1, b_2 (one for visible) control z_0^h, $(z_1^h, M_{1:16}^h)$, and $(z_2^h, M_{17:32}^h)$, respectively. The robot query remains visible.

We train both output streams by flow matching (Lipman et al., 2023). For stream $c \in \{v, s\}$ (video or state), let d_c count its target scalars and $o \in \mathbb{R}^{d_c}$ be the flattened clean target and $\epsilon \in \mathbb{R}^{d_c}$ standard Gaussian noise. With stream indices suppressed, the noisy target and flow velocity are

$$o_\sigma = (1-\sigma)o + \sigma\epsilon, \quad v^* = \epsilon - o, \quad \sigma = 5u/(1+4u), \quad u \sim \mathrm{Unif}(0,1). \tag{8}$$

The streams draw independent noise and levels σ_v, σ_s. The velocity predictor v_θ^c reads both noisy streams, their noise levels, and P_ι, with target $v^{c*} = \epsilon^c - o^c$. With the fixed scheduler weight w specified in Appendix A, the flow-matching loss averages over training windows, masks, and noise:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\sum\nolimits_{c\in\{v,s\}} \frac{w(\sigma_c)}{d_c} \left\| v_\theta^c - v^{c*} \right\|_2^2\right]. \tag{9}$$

Both experts start from random initialization.

A.3 Architecture and optimization

FP32 denotes 32-bit floating-point precision; BF16 denotes the bfloat16 format. LR is the learning rate and FFN a feed-forward network. The main variants share the configuration in Table 4. Optimization changes for the language and future-attention experiments are specified with those protocols.

Table 4: **Configuration of the main 0.6B human–robot predictor.** Runtime and peak memory refer to one 10,000-update run.

Architecture		Optimization		Precision and compute	
Trainable params.	592,926,279	Optimizer	AdamW	Model parameters	FP32
Layers	22	Total updates	10,000	Optimizer states	FP32
Attention heads	10	Global batch	128	Model autocast	BF16
Head width	128	LR (peak / final)	$10^{-4}/10^{-5}$	Frozen VAE	BF16
Video width	1280	Warmup updates	500	RGB encoding	Online per batch
State width	640	Weight decay	0.1	GPUs	4× A100-80GB
FFN / expert width	4×	Clip norm	1	Runtime	11.8 h
Sequence tokens	569	Training seed	0	Peak memory / GPU	36.59 GiB

Windows are sampled with replacement, using 256 draws per episode per nominal epoch.

For the timestep weighting in Equation 9, let $\chi(u) = 5u/(1+4u)$ and $\sigma_j = \chi(1 - j/1000)$ for $j = 0, \ldots, 999$. Define $a_w = \min_j \exp[-2(\sigma_j - 1/2)^2]$ and $c_w = 1000^{-1} \sum_j \{\exp[-2(\sigma_j - 1/2)^2] - a_w\}$. The loss weight is

$$w(\sigma) = \frac{\exp[-2(\sigma - 1/2)^2] - a_w}{c_w + 10^{-10}}. \tag{10}$$

Video and state errors are averaged separately before weighting. Validation uses unweighted velocity MSE at $\sigma \in \{0.2, 0.5, 0.8\}$; output MSE compares the integrated prediction with the clean target.

A.4 Dataset and evaluation windows

The dataset contains 1,316 paired episodes, with 1,252 assigned to training. Short segments use repeat-last padding; the main comparison selects complete 32-step target windows. Test-window sets and generation seeds are specified with each experiment in Appendix B.

B Supplementary experiments

B.1 Human–robot prediction: supplementary results

B.1.1 Prediction metrics and modality controls

Evaluation. We split paired human–robot recordings by episode record. Main comparisons and static interventions use the same 50 complete 32-step windows from 14 instruction groups. Each

model is trained with one seed; evaluation pairs windows and generation noise across conditions. The main comparison uses generation seed 7000; supplementary experiments specify their seeds below. SD denotes the standard deviation across generation seeds.

Evaluation metrics. Let $p_{i,t}$ be the ground-truth end-effector position at step t of window i, and $\hat{p}_{i,t,j}$ its prediction with generation seed j, both denormalized to millimeters. Translation ADE is

$$\text{ADE} = \frac{1}{32NK} \sum_{j=1}^{K} \sum_{i=1}^{N} \sum_{t=1}^{32} \|\hat{p}_{i,t,j} - p_{i,t}\|_2, \tag{11}$$

Here N counts windows and K generation seeds ($N = 50$, $K = 1$ for the main comparison). FDE uses the final step only. Main comparisons use the final step-10,000 checkpoints with their training modalities; the horizon analysis uses the same windows. The confidence bands in Figure 3 use 10,000 bootstrap resamples of the 14 instruction groups, retaining all windows within each sampled group.

Normalized state MSE averages squared error across seven coordinates. Image metrics exclude the observed initial frame: LPIPS (Zhang et al., 2018) uses version 0.1 with an AlexNet backbone (Krizhevsky et al., 2012) on RGB in $[-1, 1]$, SSIM uses RGB in $[0, 1]$, and PSNR averages per-frame dB values with data range 2.

Hand–arm-focused demonstration video. Segment Anything Model (SAM) 2.1 masks with manual and geometric corrections retain hands, forearms, sleeves, and any contacting-object fragments inside the mask boundaries. In training and evaluation, we set the remaining pixels to RGB $(128, 128, 128)$ before resizing and online VAE encoding. The hand–arm predictor uses video alone, with the same robot inputs and targets as the main comparison.

Language descriptions of human demonstrations. A 0.6B predictor receives automatically generated captions in place of human video and action trajectories. Each caption describes up to nine frames from an approximately four-second segment that ends by the robot query's start. Integer timestamps define the segments; original frame indices align the queries. This experiment uses 14,634 training windows. Reported test metrics use 559 windows from 49 episodes selected by the main comparison's recording-overlap filter.

The caption-conditioned predictor and no-demonstration baseline share initialization, data order, architecture, and loss. Both train for 10,000 updates with batch 32 and AdamW (learning rate 10^{-4}, 500 warmup updates, weight decay 0.01). A learned encoder maps word and punctuation tokens to 128 reserved positions. The caption-conditioned predictor drops the whole caption on half the training examples; the baseline always masks it. Both retain robot inputs and encode RGB online with the frozen VAE.

Table 5: **Language-only demonstration context.** Mean ± SD over generation seeds 0, 1, and 2 on 559 paired windows. The last two rows share weights; training captions use 50% dropout.

Caption		ADE (mm)↓	FDE (mm)↓
Training	Evaluation		
✗	✗	25.16 ± 0.15	38.16 ± 0.42
✓	✓	24.58 ± 0.15	37.08 ± 0.60
✓	✗	25.44 ± 0.03	39.06 ± 0.49

Caption input lowers mean ADE by 0.86 mm relative to masking it at fixed weights, and by 0.58 mm relative to the separately trained no-demonstration model.

Ablation of future video–state coupling. The 2×2 experiment varies future-attention coupling and the proportion of training examples with complete context (0% or 50%). The remaining examples have no demonstration. All variants use the 0.6B architecture and aligned inputs of Appendix A.2. For future video/state tokens T_v, T_s, decoupling removes $(T_v \times T_s) \cup (T_s \times T_v)$ from Equation 7. Within-stream attention, clean-context access, masked token positions, and both unit-weight flow losses remain unchanged.

All models train for 10,000 further updates on 1,252 episodes from the same no-demonstration checkpoint, with matched initialization, data order, and mask/noise draws. We reset the optimizer and scheduler and use the settings in Table 4, except for a learning rate that decreases from 3×10^{-5} to 3×10^{-6}.

We evaluate the final checkpoints on the same 50 windows with 20 Euler integration steps and seeds 7000, 17000, and 27000. Correct supplies aligned human context, Absent masks it, and Incorrect uses a different-instruction episode. P_{20} is the fraction of trajectories whose translation error stays within 20 mm at all 32 steps.

Table 6: **Future video–state attention.** Means over three generation seeds on 50 test windows; ADE in mm and P_{20} in %.

Coupled	Train context	Absent		Correct		Incorrect	
		ADE↓	P_{20}↑	ADE↓	P_{20}↑	ADE↓	P_{20}↑
✗	0%	26.50	21.33	43.07	5.33	39.41	13.33
✗	50%	26.52	19.33	23.38	26.67	27.67	15.33
✓	0%	26.58	18.00	42.28	8.00	38.46	15.33
✓	50%	27.03	18.67	20.48	30.67	29.08	12.00

Table 7: **Paired ADE differences by generation seed (mm).** The same 50 windows are evaluated with each seed. Percentages denote training context; gain is absent minus correct. Training × coupling is the coupled-minus-decoupled ADE difference at 50% context minus that at 0%.

	Generation seed			
Comparison	7000	17000	27000	Mean
No context at evaluation				
Coupled − decoupled (0%)	0.40	0.44	−0.60	0.08
Coupled − decoupled (50%)	0.73	0.50	0.30	0.51
50% − 0% (decoupled)	0.11	−0.08	0.03	0.02
50% − 0% (coupled)	0.44	−0.02	0.93	0.45
Training × coupling	0.34	0.06	0.90	0.43
50% context in training				
Correct − absent (decoupled)	−3.33	−2.75	−3.34	−3.14
Correct − absent (coupled)	−7.26	−6.57	−5.81	−6.55
Correct − incorrect (decoupled)	−4.12	−3.96	−4.79	−4.29
Gain: coupled − decoupled	3.93	3.82	2.48	3.41

Long-horizon visual comparison. Figure 4 compares complete and masked context using the main video-plus-action predictor at update 10,000. Four 32-step calls, each with 20 Euler steps, produce a 128-step continuation; each call passes its predicted frame and state to the next. The figure shows two selected windows, cropped for legibility. Metrics use the full 224×384 frames.

Context arrival during autoregressive prediction. Each predictor uses its main-comparison checkpoint at update 10,000. Both schedules use one fixed generation-noise schedule and the same 83 complete 128-step windows from 42 episodes and 14 instructions. For each predictor, demonstrations remain absent throughout or its trained modality/modalities arrive at the specified step. Each arrival condition is paired with that predictor's own absent control, matching weights, initial inputs, window order, and per-call noise. The $8 + 24$ schedule keeps eight predicted steps and predicts 24 more after context arrives. The $32 + 96$ schedule uses four 32-step calls, with arrival after the first. Prediction starts from the observed robot frame and state; subsequent calls use their preceding predictions, with 20 Euler steps and paired noise per call. The short schedule uses human context through step 40 and scores robot steps through 32.

Position is scored every step and images every four steps; PSNR is averaged in dB. Let e_t be translation error in mm and c the arrival step. Recovery requires eight consecutive post-arrival steps

with $e_t \leq 20$ among windows with $e_c > 20$. For Action, Video, and joint predictors, these subsets contain 23, 23, 27 windows in the short schedule and 51, 59, 61 in the long schedule, respectively. Stability requires the final eight scored steps to meet this threshold, using all 83 windows.

Table 8: **Joint-predictor results after demonstration arrival.** Both modalities arrive together; the same 83 windows per row. ADE/LPIPS begin at arrival; FDE is final translation error. SSIM/PSNR cover all predicted frames. Correct supplies context from the start.

Schedule	Context	ADE (mm)↓	FDE (mm)↓	LPIPS↓	SSIM↑	PSNR (dB)↑
8+24	Absent	33.09	44.26	0.095	0.9191	26.55
8+24	Arrival	12.87	11.15	0.067	0.9288	28.13
8+24	Correct	10.86	10.99	0.063	0.9328	28.97
32+96	Absent	81.44	123.60	0.184	0.8895	23.09
32+96	Arrival	17.73	12.45	0.093	0.9131	26.06
32+96	Correct	13.17	12.46	0.083	0.9213	27.32

B.1.2 Content and temporal interventions

Both tests retain the Full checkpoint and use the main comparison's 50 windows and generation-noise schedule. Full training jointly masks corresponding video and Action time blocks (Appendix A.2); evaluation varies the visible modalities separately. Table 9 removes either modality or replaces both with a different-instruction recording, replacing its video and action trajectories together. Table 10 retains all 32 Action steps while varying video availability. The causal VAE prefixes z_0^h, $z_{0:1}^h$, and $z_{0:2}^h$ encode human frames through steps 0, 16, and 32, respectively.

Table 9: **Input interventions on the trained Full model.** Every row uses the same Full checkpoint; Video and Action indicate inputs visible at evaluation. Generation seed 7000 on 50 paired windows. Wrong task marks replacement of both inputs.

Video	Action	Wrong task	ADE (mm)↓	State MSE (10^{-3})↓	LPIPS↓
✓	✓	✗	10.61	147.7	0.058
✗	✓	✗	35.17	223.9	0.093
✓	✗	✗	17.64	208.6	0.066
✗	✗	✗	36.27	328.7	0.092
✓	✓	✓	128.31	994.4	0.194

Table 10: **Video availability at fixed Full weights.** Action is retained; the 50 windows and generation seed 7000 follow Table 9.

Video prefix	Action	ADE (mm)↓	FDE (mm)↓	LPIPS↓
✗	✓	35.17	60.32	0.093
z_0^h	✓	19.74	35.87	0.073
$z_{0:1}^h$	✓	11.87	19.10	0.059
$z_{0:2}^h$	✓	10.61	14.24	0.058

B.2 Policy execution: protocols and mechanism controls

B.2.1 Curling: scene transfer and context interventions

Task and training. The learner maps an RGB image of a puck and goal to a launch velocity. Demonstrations approach from a different direction under the same physics, providing 16 video frames, launch trajectories, or both. Paired tasks share demonstrator trajectories and learner images but differ in object response. Success requires stopping within 7.5 cm of the goal after 2.4 s.

Policies train for 3,000 updates on 24,000 records, using three seeds and eight passes. Paired whole-demonstration dropout retains context on exactly four passes per record. An online RGB encoder

trained from scratch feeds two base-action branches, a response gate, and an additive correction. Physical-mode, oracle-action, and start/goal localization labels are used only for training.

Evaluation. Training excludes glacier/wood scenes and target headings $105°$–$155°$; evaluation uses headings $115°$–$145°$ with different source/target styles and directions. Evaluation uses 256 independent families with two physical siblings (paired physical conditions) and four targets each. Scene conditions share these 2,048 queries and physical settings. At final weights, context removal masks all source inputs; physical siblings then produce identical actions. Table 11 reports the interventions; column names list demonstrator and learner scene styles, in that order.

Table 11: **Curling execution and context interventions.** Mean success (%) over three training seeds on independent confirmation families. Training columns identify the policy; evaluation columns show supplied inputs. Interventions retain each policy's weights.

Training		Evaluation		Intervention	Rink–slate	Glacier–wood	Wood–glacier
Video	Action	Video	Action				
✗	✗	✗	✗	—	24.92	24.92	24.87
✗	✓	✗	✓	—	24.69	24.69	24.67
✓	✗	✗	✗	Removed	25.08	25.08	25.08
✓	✗	✓	✗	—	71.11	71.29	70.49
✓	✗	✓	✗	Wrong physics	0.00	0.00	0.00
✓	✗	✓	✗	Query swapped	8.06	7.91	7.83
✓	✓	✗	✗	Removed	25.26	25.28	25.28
✓	✓	✓	✓	—	66.86	66.80	66.36
✓	✓	✓	✓	Wrong physics	0.00	0.00	0.00
✓	✓	✓	✓	Query swapped	7.86	7.98	7.89

B.3 Paired 3D geometry: ID and OOD

In all three tasks, the learner receives its goal, numeric state, and geometry. A convolutional neural network (CNN) trained from scratch encodes source RGB online, alongside fixed spatial moments of RGB changes. Each modality variant has a base branch that reads only learner inputs and a residual branch. Oracle actions are commands computed by a simulator-informed expert. Commands for a nominal reference world supervise the base; commands for each task's actual physical conditions supervise the residual. Training uses squared error and detaches the base when fitting the residual. Physical siblings share nominal labels.

Training uses 16,384 records (2,048 families, two siblings, four queries), batch 64, and 16 passes. AdamW has weight decay 10^{-4} and learning rate 3×10^{-4}, reduced fivefold after 3,277 updates.

Actions and success criteria. All action components below are velocities in m/s.

- **Aerial gate:** a 3D launch velocity and a 3D velocity increment at 0.8 s, each component in $[-12, 12]$. Success requires gate-center error at most 22 cm at 0.8 s, a clear aperture crossing within 1.6 s without rim contact, and target error at most 20 cm at 1.6 s.
- **Tilted bank:** a 3D launch velocity in $[1.5, 5.5] \times [-2, 2] \times [2, 8]$. Success requires panel contact, no other collision, and target error at most 14 cm at 1.2 s.
- **Ramp-to-air:** two initial ramp-tangent velocity components in $[1, 5.5] \times [-1.2, 1.2]$. Success requires ramp contact, downhill-edge departure, no other collision, and horizontal target error at most 13 cm at the descending target-plane crossing within 1.7 s.

Final 4,096-update checkpoints are evaluated on 1,024 records per condition: 128 families, two physics siblings, and four target queries. Three training seeds share the evaluation families. Following earlier shift evaluations, we fixed the OOD ranges just beyond training support for this paired test.

Geometric shifts leave source demonstrations, physics, and other query fields unchanged. The two factors are gate and target lateral positions for aerial gate, yaw and pitch for tilted bank, and tilt and yaw for ramp-to-air. Each shift preserves the original sign. Bank and ramp goals remain fixed. The ramp's local starting position also stays fixed, so its world position moves with the surface.

Table 12: **Paired 3D execution success (%) under geometric shifts.** Means over three training seeds at the native timestep. Video/Action identify training inputs; Removed masks both at fixed weights. F1/F2 shift gate/goal position, bank yaw/pitch, or ramp tilt/yaw; Both shifts both factors and corresponds to OOD in Table 2(b).

Training			Gate				Bank				Ramp			
Video	Action	Evaluation	ID	F1	F2	Both	ID	F1	F2	Both	ID	F1	F2	Both
✗	✗	As trained	49.35	49.32	49.32	48.83	45.25	33.01	22.62	11.46	61.88	57.75	58.07	64.88
✗	✓	As trained	49.35	49.32	49.32	48.76	45.70	33.11	22.82	11.56	62.30	57.71	57.94	64.68
✓	✗	As trained	68.10	67.97	67.42	67.12	65.53	40.72	30.24	15.10	99.51	67.64	88.51	67.84
✓	✓	As trained	68.26	67.94	68.29	68.13	74.84	46.35	35.61	19.63	99.54	68.20	87.63	67.90
✓	✓	Removed	49.74	49.54	49.77	49.41	46.29	33.98	20.28	10.32	61.43	56.77	58.01	63.57

Training and OOD ranges are as follows (angles in radians). Gate and target lateral magnitudes $|y|$ change from ≤ 0.35 and ≤ 0.60 m to $[0.36, 0.40]$ and $[0.61, 0.66]$ m, respectively. Bank $|\mathrm{yaw}|$ and $|\mathrm{pitch}|$ change from ≤ 0.30 and ≤ 0.16 to $[0.305, 0.325]$ and $[0.165, 0.180]$. Ramp tilt changes from $[0.20, 0.36]$ to $[0.365, 0.380]$; its $|\mathrm{yaw}|$ changes from ≤ 0.30 to $[0.305, 0.325]$.

We replay saved actions at native/finer timesteps of 0.004/0.002 s for gate, 0.001/0.0005 s for bank, and 0.001/0.00025 s for ramp.

B.3.1 Mechanism controls with policy execution

All four modality variants share learner inputs and future-coordinate supervision and train for 4,096 updates. Throwing and rebound use 16,384 records for 16 passes; pushing uses 8,192 records for 32 passes. Policies execute predicted commands in simulation and receive no physical-parameter supervision.

Tasks. In throwing, success requires passing within 20 cm of the target while descending through its height. In rebound, a planar slider moves without gravity or rotation, contacts a wall, and finishes within 10 cm of its goal at 1.2 s; contact damping and friction vary across tasks. In pushing, a straight push moves a rigid body under friction, contact, and gravity. The learner receives the goal, initial state, mass, incline, and deadline. Action describes source commands or pusher tracking, excluding the object response; Video records the interaction.

Fixed versus adaptive demonstrations. Fixed probes use commands independent of hidden physics. Adaptive experts use the physics to choose commands that reach a source goal. The throwing expert aims at two calibration targets. The rebound expert reaches within 1.5 cm of $(0.3, 0.3)$ m at 1.2 s after wall contact; its Action input contains the source initial state and selected launch velocity. The pushing expert moves a 0.2 kg object to $(0.3, 0)$ m on a horizontal surface, choosing a speed in $[0.35, 2.2]$ m/s.

Learner tasks and labels, architecture, initialization, sampling order, and update budget are matched between fixed and adaptive demonstrations. None is reused because its inputs and labels coincide. Throwing uses three training seeds; rebound and pushing use seed 0. Table 2(c) reports adaptive results.

Geometry extrapolation. Final policies are evaluated on paired ID/OOD queries that differ only in geometry; demonstrations, physics, and success criteria are retained. Each condition contains 2,048 records from 256 families, with two physical siblings and four queries per family. Earlier evaluations informed the OOD ranges below, which were fixed before this paired test. Rebound retains the wall-angle sign and the world-coordinate target.

The ranges below are given as ID $\rightarrow$ OOD. For throwing, target x is $[0.70, 1.50] \rightarrow [1.54, 1.58]$ m. For rebound, wall distance is $[0.80, 1.20] \rightarrow [1.22, 1.24]$ m and absolute wall angle is $[0, 0.50] \rightarrow [0.55, 0.60]$ rad. For pushing, target distance is $[0.180, 0.340] \rightarrow [0.348, 0.356]$ m. Table 2(a) averages three training seeds evaluated on the same families.

Table 13: **Matched fixed/adaptive demonstrations across 3D geometry conditions.** Execution success (%) at the native timestep, training seed 0. F1/F2 shift one factor; Both shifts both and matches the OOD condition in Table 2(d). All rows receive the same source specification. The no-demonstration baseline is reused for fixed and adaptive demonstrations.

			Aerial gate				Tilted bank				Ramp to air			
Demo.	Video	Action	ID	F1	F2	Both	ID	F1	F2	Both	ID	F1	F2	Both
Fixed	✗	✗	49.32	48.93	49.41	48.34	43.75	33.50	22.17	10.35	62.11	57.42	58.50	63.96
Fixed	✗	✓	49.32	48.93	49.41	48.24	43.65	32.23	21.88	10.06	61.23	56.64	57.52	64.36
Fixed	✓	✗	69.34	68.95	68.36	68.46	67.68	34.96	32.91	16.89	99.51	66.11	87.50	67.38
Fixed	✓	✓	66.80	67.09	66.60	66.89	71.78	40.14	35.06	18.26	99.90	68.85	87.89	67.29
Adaptive	✗	✗	49.32	48.93	49.41	48.34	43.75	33.50	22.17	10.35	62.11	57.42	58.50	63.96
Adaptive	✗	✓	73.83	72.85	72.56	72.27	83.01	57.23	38.48	21.48	99.32	66.11	86.13	68.95
Adaptive	✓	✗	45.70	45.80	45.21	45.51	81.05	52.83	38.18	21.09	97.56	68.85	84.96	68.36
Adaptive	✓	✓	78.91	77.05	77.05	75.88	82.91	56.25	37.50	19.73	99.51	69.24	87.40	69.53

Table 14: **Fixed-weight context interventions with fixed and adaptive demonstrations.** Execution success (%) at the native timestep, training seed 0. Each ID/Both cell has 1,024 records. Policies are trained with both modalities; marks show evaluation inputs. The common numerical source specification remains available.

				Aerial gate		Tilted bank		Ramp to air	
Demo.	Video	Action	Intervention	ID	Both	ID	Both	ID	Both
Fixed	✗	✗	Removed	49.41	48.24	45.12	10.25	61.43	63.87
Fixed	✓	✓	Wrong physics	0.88	0.88	35.35	8.69	49.51	52.15
Adaptive	✗	✗	Removed	49.02	48.44	44.43	10.25	62.99	64.06
Adaptive	✓	✓	Wrong physics	2.05	1.95	40.72	9.47	48.83	52.44

Adaptive demonstrations in 3D. These controls use the tasks and evaluation conditions of Appendix B.3. Every model receives the same source initial state, goal/waypoint, known geometry, and mass across paired physical conditions and fixed/adaptive demonstrations. Source commands form a separate 12-dimensional Action input.

Source experts use true dynamics to choose commands; learners receive the listed source fields and their selected demonstration modalities. All source demonstrations meet the success criteria at both native and finer timesteps. Aerial-gate videos omit the gate while preserving the selected source trajectory.

Training and evaluation. For fixed and adaptive demonstrations, we train variants with video, action trajectories, or both. The shared no-demonstration baseline retains the common source fields. All models start from the same seed-0 initialization and train for 4,096 updates, with 16 passes over 16,384 records and online RGB encoding. Only the correction branch reads source fields; the nominal branch reads the learner specification.

Whole-demonstration dropout masks video and action inputs on half the exposures and retains the source fields. Actual-action supervision covers all 16 exposures (eight in the original suite). Table 2(d) reports adaptive results; Table 13 compares fixed and adaptive demonstrations under this shared protocol.

We evaluate final policies on the same 1,024 records from 128 families per geometry condition (Table 13). Context interventions either mask video and action inputs or exchange them between physical siblings; both retain the common source fields (Table 14). We re-execute saved actions at the native and finer timesteps specified in Appendix B.3.

C THEORETICAL DEFINITIONS AND PROOFS

C.1 POPULATION MODEL AND CONVENTIONS

Demonstrations are independent pairs in the population model and temporal blocks from one interaction in the recorded-data model (Equation (6)).

Notation. Here f, g denote feature maps and f_τ^e denotes conditional response laws. The dimensions d, k, and m refer to inputs, task features, and the output of the learned representation, respectively. We write $\mathbb{E}$ for expectation, $\|\cdot\|$ for Euclidean norm, $\top$ for transpose, and I_k for the $k \times k$ identity matrix. The representation is $h = h_\mu$, with attention parameter W. In this analysis, σ denotes ReLU and μ the representation measure; in the flow-matching formulation, σ instead denotes the noise level.

Representation. Following Kim & Suzuki (2024), we consider a two-layer ReLU network followed by linear self-attention. For an atom $\vartheta = (a, w)$ with $a \in \mathbb{R}^m$, $w \in \mathbb{R}^d$, and $\sigma(t) = \max(t, 0)$, write $h_\vartheta(x) = a\sigma(w^\top x)$. A probability measure μ over atoms defines the mean-field representation $h_\mu(x) = \int h_\vartheta(x)\, \mu(\mathrm{d}\vartheta)$. The attention parameter is $W \in \mathbb{R}^{m\times m}$ (Zhang et al., 2024; Mahankali et al., 2024).

Task coordinates. The isotropic second-moment assumption is a coordinate choice: for positive definite $\Sigma_\beta = \mathbb{E}[\beta\beta^\top]$, replace (β, f, g) by $(\Sigma_\beta^{-1/2}\beta, \Sigma_\beta^{1/2}f, \Sigma_\beta^{1/2}g)$; for singular $\Sigma_\beta = U\Lambda U^\top$, let Λ contain its positive eigenvalues and U the corresponding orthonormal eigenvectors, and replace (β, f, g) by $(\Lambda^{-1/2}U^\top\beta, \Lambda^{1/2}U^\top f, \Lambda^{1/2}U^\top g)$. If $\Sigma_\beta = 0$, all task responses vanish almost surely. Covariates are independently and identically distributed (i.i.d.) from $\mathbb{P}$, independently of β.

C.2 POPULATION REALIZABILITY: DEFINITIONS AND PROOFS

C.2.1 BARRON SPACE

Functions are identified up to $\mathbb{P}$-almost-sure equality. For scalar output and $p \in [1, \infty]$, let ν range over probability measures on $\mathbb{R} \times \mathbb{R}^d$ and define

$$\|\phi\|_{\mathcal{B}_p} := \inf_{\nu:\, \phi(x)=\int a\sigma(w^\top x)\, d\nu} \| |a|\|w\| \|_{L^p(\mathbb{R}\times\mathbb{R}^d,\nu)} \tag{12}$$

Here $L^p(\nu)$ uses the pth-moment norm (essential supremum for $p = \infty$). The bias-free ReLU Barron space $\mathcal{B}_p$ of order $p \in [1, \infty]$ consists of functions with finite Barron norm $\|\cdot\|_{\mathcal{B}_p}$. ReLU homogeneity makes this norm independent of p (Kim & Suzuki, 2024), so we write $\mathcal{B}$. $\mathcal{B}^k$ is the coordinatewise extension of scalar Barron space.

Bounded-product representation and population limit. For $h \in \mathcal{B}^m$, choose coordinate representations ν_ℓ and let $B = \sum_{\ell=1}^m \int |a|\|w\|\, d\nu_\ell < \infty$. For $B > 0$, mix the measures $|a|\|w\|\, d\nu_\ell/B$ and map each nonzero atom to $(B\,\mathrm{sign}(a)e_\ell, w/\|w\|)$, where e_ℓ is a coordinate vector. ReLU homogeneity preserves h and gives a probability representation with atom product B. For $B = 0$, use a zero atom. Conversely, bounded atom product places each coordinate in $\mathcal{B}$.

The bound $\|h(x)\| \le B\|x\|$ gives $h \in L^2(\mathbb{P})$. Thus $fh^\top$ is integrable and the strong law yields $n^{-1}\sum_i f(x_i)h(x_i)^\top \to A_h$ almost surely. The limiting prediction is $\beta^\top A_h W h(x_q)$; independence and $\mathbb{E}[\beta\beta^\top] = I_k$ give its squared-error risk in Equation (4).

C.2.2 PROOF OF PROPOSITION 1

Proof. Write $\mathrm{range}(M) = \mathcal{R}(M)$ for a matrix column space and $\mathcal{L}_{\mathrm{TF}}(h, W)$ for the risk induced by a representation $h = h_\mu$. Denote $A_h = \mathbb{E}[fh^\top], K_g := \mathbb{E}_x[gg^\top], C_{fg} := \mathbb{E}_x[fg^\top], \mathcal{D}_f := \{\mathbb{E}_x[f\phi] : \phi \in \mathcal{B}\}$, and

$$\mathcal{U} := \mathrm{span}\{g_1, \ldots, g_k\}, \qquad \mathcal{V} := \mathrm{range}(K_g), \qquad \mathcal{M} := \mathrm{range}(C_{fg}),$$

and let $r := \mathrm{rank}(K_g)$. Since K_g is the Gram matrix of the coordinate functions of g, we have $\dim\mathcal{U} = \dim\mathcal{V} = r$.

Introduce the linear map

$$\Gamma_f : L^2(\mathbb{P}) \longrightarrow \mathbb{R}^k, \qquad \Gamma_f(\phi) := \mathbb{E}[f\phi].$$

By definition,

$$\Gamma_f(\mathcal{B}) = \mathcal{D}_f, \qquad \Gamma_f(\mathcal{U}) = \mathcal{M}.$$

Also, writing

$$\delta := \dim(\mathcal{V} + \mathcal{M}) - \dim \mathcal{M} = r - \dim(\mathcal{V} \cap \mathcal{M}),$$

we obtain

$$\delta = \operatorname{rank}[K_g \quad C_{fg}] - \operatorname{rank}(C_{fg}).$$

Thus the dimension condition in (5) is equivalent to $m \geq r + \delta$.

Necessity. Suppose $\mathcal{L}_{\mathrm{TF}}(h, W) = 0$. Then

$$g = A_h W h \qquad \mathbb{P}\text{-almost surely.}$$

Let

$$\mathcal{S} := \operatorname{span}\{h_1, \ldots, h_m\}.$$

Each coordinate of g is a linear combination of the coordinates of h. Consequently,

$$g \in \mathcal{B}^k, \qquad \mathcal{U} \subseteq \mathcal{S}, \qquad \dim \mathcal{S} \leq m.$$

This proves the Barron condition.

Set $C_h := \mathbb{E}[hh^\top]$. The zero-loss identity gives

$$K_g = A_h W C_h W^\top A_h^\top.$$

Therefore,

$$\mathcal{V} \subseteq \operatorname{range}(A_h) = \Gamma_f(\mathcal{S}) \subseteq \Gamma_f(\mathcal{B}) = \mathcal{D}_f,$$

which proves accessibility.

Since $\mathcal{U} \subseteq \mathcal{S}$, we have $\mathcal{M} \subseteq \Gamma_f(\mathcal{S})$. Together with $\mathcal{V} \subseteq \Gamma_f(\mathcal{S})$, this yields

$$\delta \leq \dim \Gamma_f(\mathcal{S}) - \dim \mathcal{M}.$$

The map Γ_f induces a surjective linear map

$$\mathcal{S}/\mathcal{U} \longrightarrow \Gamma_f(\mathcal{S})/\mathcal{M}, \qquad \phi + \mathcal{U} \longmapsto \Gamma_f(\phi) + \mathcal{M}.$$

Hence

$$\delta \leq \dim \Gamma_f(\mathcal{S}) - \dim \mathcal{M} \leq \dim \mathcal{S} - \dim \mathcal{U} \leq m - r.$$

Thus $m \geq r + \delta$, proving the dimension condition.

Sufficiency. Assume (5). If $g = 0$, taking $h = 0$ and $W = 0$ proves the claim. Otherwise, choose an orthonormal basis $u_1, \ldots, u_r$ of $\mathcal{U}$. Because $g \in \mathcal{B}^k$ and $\mathcal{B}$ is a linear space, every u_i belongs to $\mathcal{B}$.

Let $u = (u_1, \ldots, u_r)^\top$. There exists $D_0 \in \mathbb{R}^{k \times r}$ such that

$$g = D_0 u.$$

Orthonormality gives

$$\mathbb{E}[uu^\top] = I_r, \qquad K_g = D_0 D_0^\top, \qquad \operatorname{range}(D_0) = \mathcal{V}.$$

Choose $y_1, \ldots, y_\delta \in \mathcal{V}$ whose cosets form a basis of $(\mathcal{V} + \mathcal{M})/\mathcal{M}$. Since $\mathcal{V} \subseteq \mathcal{D}_f$, there exist $\phi_1, \ldots, \phi_\delta \in \mathcal{B}$ with

$$\Gamma_f(\phi_j) = y_j, \qquad j = 1, \ldots, \delta.$$

When $\delta = 0$, these lists are empty.

Since $m \geq r + \delta$, define

$$h := (u_1, \ldots, u_r, \phi_1, \ldots, \phi_\delta, \underbrace{0, \ldots, 0}_{m-r-\delta})^\top \in \mathcal{B}^m.$$

Its source cross-moment matrix is

$$A_h = [\Gamma_f(u_1) \quad \cdots \quad \Gamma_f(u_r) \quad y_1 \quad \cdots \quad y_\delta \quad 0],$$

where the final zero block has $m - r - \delta$ columns. Because $\Gamma_f(\mathcal{U}) = \mathcal{M}$ and the cosets of the y_j span $(\mathcal{V} + \mathcal{M})/\mathcal{M}$,

$$\operatorname{range}(A_h) = \mathcal{M} + \operatorname{span}\{y_1, \ldots, y_\delta\} = \mathcal{M} + \mathcal{V}.$$

Set

$$D := \begin{bmatrix} D_0 & 0_{k\times(m-r)} \end{bmatrix}.$$

Then $g = Dh$ and

$$\operatorname{range}(D) = \mathcal{V} \subseteq \operatorname{range}(A_h).$$

Let $A_h^\dagger$ denote the Moore–Penrose pseudoinverse and choose

$$W := A_h^\dagger D.$$

Since $A_h A_h^\dagger$ is the orthogonal projector onto $\operatorname{range}(A_h)$,

$$A_h W = A_h A_h^\dagger D = D.$$

Consequently,

$$A_h W h = Dh = g \qquad \mathbb{P}\text{-almost surely},$$

so $\mathcal{L}_{\mathrm{TF}}(h, W) = 0$. The normalization above realizes this $h \in \mathcal{B}^m$ with a probability measure of bounded atom product. □

C.3 Examples of the width threshold

Let $\mathbb{P}$ be uniform on the four coordinate vectors in $\mathbb{R}^4$ and define $\phi_j(x) = 2\,\mathrm{ReLU}(x_j)$. Then $\mathbb{E}[\phi_i \phi_j] = \mathbf{1}\{i = j\}$. Fix $g = (\phi_1, \phi_2)^\top$, so $K_g = I_2$, and vary only the demonstrator map f:

$f^\top$	C_{fg}	$\dim(\mathcal{R}(C_{fg}) \cap \mathcal{R}(K_g))$	Minimum m
(ϕ_1, ϕ_2)	I_2	2	2
(ϕ_1, ϕ_3)	$\mathrm{diag}(1, 0)$	1	3
(ϕ_3, ϕ_4)	$0_{2\times 2}$	0	4

In every row, $\mathcal{D}_f = \mathbb{R}^2$. Equation (5) therefore gives $m_{\min} = 4 - \dim(\mathcal{R}(C_{fg}) \cap \mathcal{R}(K_g))$. Each bound is attained by $h = (\phi_1, \ldots, \phi_m)^\top$ and $W = A_h^\dagger [I_2\ 0]$, which give $A_h W h = g$.