TAM-WM: TRANSITION-ALIGNED MULTI-VIEW WORLD MODEL FOR VISUAL ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-view observations provide complementary information for visual robot manipulation, but they also require a world model to represent the consequences of an action consistently across viewpoints. Existing approaches often encourage synchronized observations to have similar representations, which primarily enforces agreement about the current state. For action selection, however, the model must also assign consistent finite-horizon changes to the same physical transition; strong state agreement alone does not guarantee such transition agreement.

To address the issue of inconsistent transition representations across viewpoints, we propose TAM-WM, which learns a shared visual predictive coordinate in two stages. Stage A aligns the displacements induced by the same observed physical transition across synchronized views. Stage B freezes this coordinate and learns state- and action-conditioned residual dynamics, enabling candidate actions from a shared current state to be compared through their predicted changes. A downstream online correction teacher uses these predictions to score local proposals and selectively correct a nominal policy action. On Can PH, a fixed-scorer ablation shows that removing transition alignment reduces demonstrated-action ranking, while the learned scores correlate with simulator-executed candidate outcomes. Across eight simulated manipulation tasks, the complete system achieves the highest equal-weight average success-rate point estimate among the evaluated methods: 75.85%, 5.28 percentage points above VILA and 8.99 points above the behavior-cloning anchor.

INTRODUCTION

Visual robot manipulation requires anticipating how actions will change a scene, yet a single camera provides only a partial observation. Objects can be occluded, and contact visible from one viewpoint may be hidden from another. A scene camera can reveal global object motion, while a wrist camera shows local interactions between the gripper and an object. These complementary observations are different projections of the same physical interaction. Visual world models offer a way to predict action consequences for planning and action selection Finn & Levine (2017); Ebert et al. (2018); Hafner et al. (2019); Hansen et al. (2022). *How can a multi-view world model turn complementary visual observations into a coherent prediction of how the physical world changes under robot actions?*

Existing multi-view methods encourage synchronized observations to share information through correspondence, reconstruction, masked prediction, or view-invariant representations Sermanet et al. (2018); Seo et al. (2023); Pang et al. (2025). More recent work also incorporates action information and physical dynamics into cross-view representations or learns action-conditioned future prediction Lee et al. (2026); Jeong et al. (2026); Guo et al. (2026).

Yet three challenges remain when these representations are used for action selection. **State agreement does not guarantee transition agreement.** On Can PH, DINO-WM has a cross-view State Match of 0.9318 but a Delta Match of only 0.0012 (Figure 1). The two cameras' representations largely agree on the current scene but disagree on how it has changed. In a matched controlled intervention, adding transition alignment increases Delta Match and demonstrated-action MRR while State Match remains essentially unchanged. This motivates directly aligning the displacement assigned to matched physical transitions rather than relying on state agreement alone. **Aligned ob-**

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.

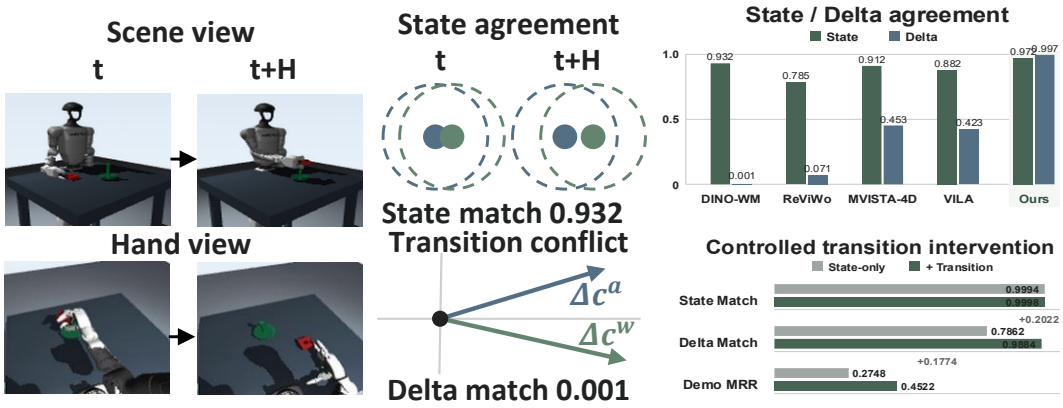

Figure 1: Motivation and controlled intervention. Top: high cross-view State Match can coexist with substantially lower Delta Match, so state-level agreement does not guarantee agreement on the same physical transition. Bottom: in a matched controlled intervention, adding transition alignment leaves State Match nearly unchanged while increasing Delta Match and demonstrated-action MRR. These controlled results do not imply a monotonic relationship between Delta Match and ranking across different methods.

served transitions do not tell us what alternative actions will do. Training data show what happened after the executed action, but action selection requires predicting what several candidate actions would do from the same current state before executing any of them. **A good predictor does not automatically make a good controller.** Its scores must correspond to actual physical outcomes and be reliable enough to guide action selection.

To address these three problems, we introduce TAM-WM. Stage A learns a shared visual predictive coordinate by aligning finite-horizon displacements for the same observed physical transition across synchronized views. Stage B freezes the Stage-A coordinate and learns state- and action-conditioned residual dynamics to predict the change caused by a candidate action from the current state. For downstream control and validation, selective correction scores local candidates and replaces the nominal action only when a candidate is predicted to be sufficiently better and reliable.

We evaluate TAM-WM through cross-view diagnostics, controlled candidate ranking, simulator-executed physical outcomes, and selective closed-loop correction.

Our contributions are:

- **Transition-level diagnosis for multi-view world models.** We show that strong approximate cross-view agreement of synchronized states can coexist with severe disagreement in the finite-horizon displacement assigned to the same physical transition.
- **Transition-aligned action-conditioned dynamics.** TAM-WM learns a visual predictive coordinate by aligning matched cross-view transitions, then freezes this coordinate and learns state- and action-conditioned residual dynamics for comparing alternative actions.
- **Prediction-to-control validation.** We evaluate candidate ranking, simulator-executed physical outcomes, and selective closed-loop correction, testing whether predictive improvements translate into action selection and control.

Related Work

State-Level and Multi-View Representation Learning

Visual pretraining supplies transferable features for robot manipulation Caron et al. (2021); Oquab et al. (2023); Nair et al. (2022); Xiao et al. (2022); Majumdar et al. (2023), and world models use pixel or latent dynamics for planning Finn & Levine (2017); Hafner et al. (2019; 2020); Kipf et al. (2020); Zhou et al. (2025). Within multi-view learning, temporal correspondence methods learn

aligned representations from video, masked prediction uses synchronized views as context, and factorization methods separate shared task information from view-dependent content Sermanet et al. (2018); Dwibedi et al. (2019); Zakka et al. (2022); Seo et al. (2023); Pang et al. (2025). These objectives provide complementary ways to represent shared states, observations, and temporal correspondence. TAM-WM studies the finite-horizon change assigned to a matched observed transition.

Action-Centric and Dynamics-Aware Cross-View Representations

MVP-LAM learns action-centric latent tokens through cross-viewpoint future reconstruction Lee et al. (2026), while VILA learns view-invariant latent actions using action and physical-dynamics supervision Jeong et al. (2026). Their latent actions represent observed behavior across views and support downstream policy learning. TAM-WM targets a different predictive object: it learns a state coordinate c_t, aligns its displacement for matched observed transitions, and then trains $T_\theta(c_t, A)$ to predict the displacement under each candidate action. This forward interface compares proposals that share a current state but imply different predicted futures.

Action-Conditioned Multi-View World Models and Control

Action-conditioned world models support controllable future prediction and action inference. Ctrl-World models controllable futures for robot manipulation Guo et al. (2026), and MVISTA-4D combines view-consistent 4D prediction with test-time action inference Wang et al. (2026). Their emphasis on future generation, controllability, and action inference complements TAM-WM's shared state-coordinate dynamics for comparing alternative action consequences. Downstream, learned dynamics can support candidate scoring and residual control Chua et al. (2018); Hansen et al. (2022); Johannink et al. (2019); policy distillation can transfer selected behavior to a policy-only model Rusu et al. (2016); Liu et al. (2022). In TAM-WM these are downstream uses of the Stage-A/Stage-B predictor.

Method

Problem Setup and Design Overview

TAM-WM learns a visual predictive state coordinate and a forward model of action consequences in that coordinate. Stage A aligns the finite-horizon state displacements assigned to the same observed transition by synchronized views. With this coordinate frozen, Stage B predicts the change induced by each candidate action. Candidate ranking tests these predictions, and a downstream controller uses them for selective correction.

A training tuple contains synchronized multi-view observations at two endpoints of an executed action chunk:

$$\mathcal{T}_t = (o_t^a, o_t^w, o_{t+H}^a, o_{t+H}^w, A_{t:t+H-1}),$$

where $v \in \{a, w\}$ denotes the agent and wrist model views, and $A_{t:t+H-1}$ is the executed H-step action chunk. A frozen pretrained visual encoder ϕ produces $z_t^v = \phi(o_t^v)$. Stage A uses the transition-relevant encoder E_c to produce $c_t^v = E_c(z_t^v)$ such that latent displacements $\Delta c_t^v = c_{t+H}^v - c_t^v$ induced by the same executed action are consistent across views; the action chunk identifies which physical transition both views observe but is not used as input to Stage A encoders. Stage B learns state- and action-conditioned dynamics $T_\theta(c_t, A_{t:t+H-1})$ in the frozen Stage A coordinate. Future observations o_{t+H}^v supervise training but are unavailable at inference. At deployment, the system uses cross-view aggregation and evaluates candidates by predicted progress toward a goal prototype.

Transition-Aligned Predictive Coordinate

Stage-A design. Stage A treats the finite-horizon displacement in a predictive state coordinate as the unit of cross-view supervision. State alignment anchors the coordinates at synchronized times; transition alignment directly constrains their finite-horizon changes. Auxiliary reconstruction, cross-composition, and decorrelation discourage collapse and redundancy. The auxiliary context provides a reconstruction path that Stage B does not consume; decorrelation discourages overlap but does not impose complete separation of information.

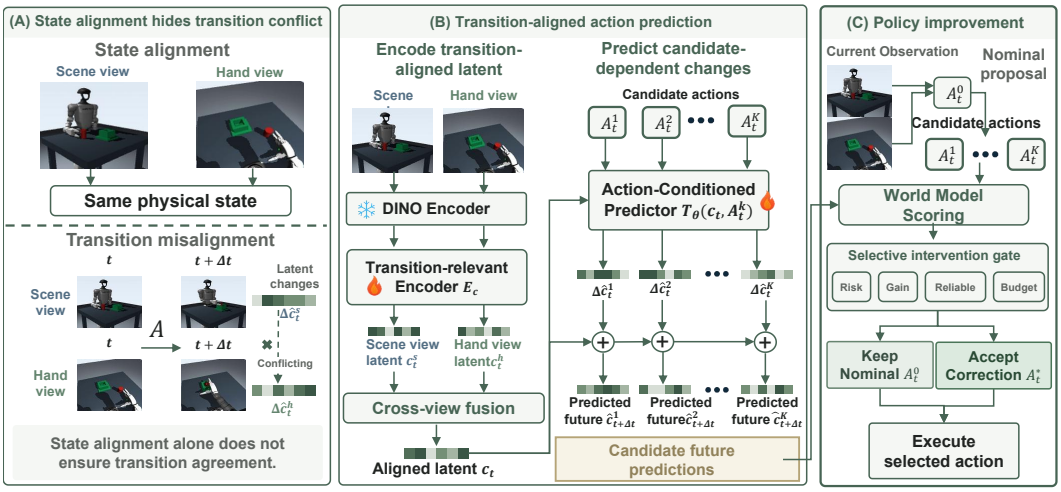

Figure 2: TAM-WM from observed transitions to action selection. (A) Synchronized views can assign different displacements to the same observed transition. (B) Stage A aligns these changes in a predictive state coordinate; Stage B maps that shared current coordinate and each candidate action to a predicted change. (C) A downstream controller scores candidate futures and selectively corrects a nominal action. Scene/hand labels denote the agent/wrist model views.

For each view v, we encode $c_t^v = E_c(z_t^v)$ and compute latent displacements $\Delta c_t^v = c_{t+H}^v - c_t^v$. Aligning the two endpoints $c_t^a \approx c_t^w$ and $c_{t+H}^a \approx c_{t+H}^w$ need not yield close displacements when the remaining endpoint errors vary with time. Exact endpoint agreement, by contrast, implies exact displacement agreement. Let $e_t = c_t^a - c_t^w$ denote the residual alignment error. The displacement mismatch decomposes as:

$$\Delta c_t^a - \Delta c_t^w = (c_{t+H}^a - c_{t+H}^w) - (c_t^a - c_t^w) = e_{t+H} - e_t.$$

The difference between endpoint errors matters most relative to a small encoded displacement; the corresponding bound appears in the appendix.

We use state alignment as a coordinate anchor and directly constrain the displacement later predicted under candidate actions:

$$\begin{aligned} \mathcal{L}_{\text{state}} &= d_\epsilon(c_t^a, c_t^w), \\ \mathcal{L}_{\text{trans}} &= d_\epsilon(\Delta c_t^a, \Delta c_t^w), \end{aligned}$$

where $d_\epsilon(x, y) = \|x-y\|_2^2 + 1 - x^\top y / ((\|x\|_2 + \epsilon)(\|y\|_2 + \epsilon))$ combines MSE and cosine similarity. Future coordinates enter the displacement construction, while the explicit per-tuple state-alignment term applies to the current synchronized coordinates.

Discouraging coordinate collapse. The alignment objectives alone admit a trivial solution where all inputs map to a constant. We discourage this collapsed solution with an auxiliary-context encoder E_u, which produces $u_m^v = E_u(z_m^v)$ at a different time index $m \neq t$ in the same demonstration. The view-specific decoder reconstructs the frozen feature as $\hat{z}_t^{v_s \to v_t} = R_{v_t}(c_t^{v_s}, u_m^{v_t})$. The supporting reconstruction objective includes agent-to-wrist and wrist-to-agent cross-view terms and agent-to-agent and wrist-to-wrist same-view terms (see the auxiliary reconstruction details in the appendix). Sampling another frame reduces direct same-frame copying and encourages c_t to retain time-specific content, although u can still contain current-state information. The auxiliary branch can carry observation details that the dynamics model does not consume. We penalize cross-correlation $\mathcal{L}_{\text{decor}}$ over a shared subset of c and u coordinates. The complete Stage A objective is:

$$\mathcal{L}_A = \lambda_s \mathcal{L}_{\text{state}} + \lambda_\Delta \mathcal{L}_{\text{trans}} + \mathcal{L}_{\text{recon}} + \lambda_d \mathcal{L}_{\text{decor}}.$$

The appendix gives sampling and decoder details and tests the coordinate's time-specific contribution to reconstruction.

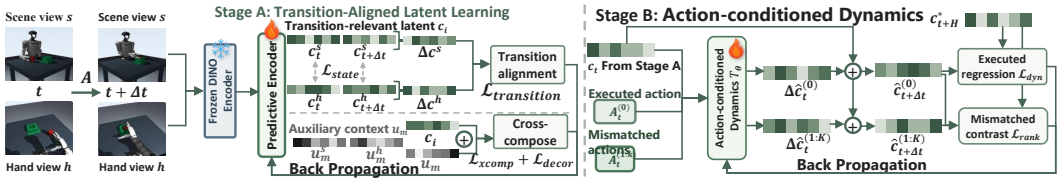

Figure 3: Two-stage training pipeline. Scene/hand labels in the illustration denote the agent/wrist model views. Stage A aligns matched cross-view changes and uses auxiliary context for feature reconstruction; after its predictive coordinate is frozen, Stage B predicts candidate-dependent transitions. The predictive coordinate is visual-only; Stage B additionally receives the action sequence. Stage A includes same-view and cross-view reconstruction. Future observations are training targets, not inference inputs.

Action-Conditioned Residual Dynamics

Transition alignment alone does not ensure candidate-sensitive prediction: aligned observed changes must also support a forward map from the current state and alternative action chunks to their predicted consequences. We freeze Stage A before training Stage B, so Stage-B gradients cannot redefine its cross-view coordinate. The view-specific coordinates c_t^a and c_t^w form a consensus predictive coordinate:

$$c_t = \mathrm{Agg}(c_t^a, c_t^w).$$

The same aggregation operator defines the observed future target,

$$c^\star_{t+H} = \mathrm{Agg}(c^a_{t+H}, c^w_{t+H}), \qquad \Delta c_t^\star = c^\star_{t+H} - c_t.$$

We then train

$$\Delta\hat{c}_t = T_\theta(c_t, A_{t:t+H-1}), \qquad \hat{c}_{t+H} = c_t + \Delta\hat{c}_t,$$

with

$$\mathcal{L}_{\mathrm{dyn}} = d_\epsilon(\hat{c}_{t+H}, c^\star_{t+H}) + \lambda_r d_\epsilon(\Delta\hat{c}_t, \Delta c_t^\star).$$

The aggregation operator is detailed in the appendix. Candidate actions share c_t but differ in $A_{t:t+H-1}$, so the model must produce candidate-dependent predicted displacements $\Delta\hat{c}_t$ in the same state coordinate. The endpoint term limits long-horizon drift. The dynamics model receives a visual predictive-coordinate token and H action-step tokens. Proprioception may remain available to the separate policy and robot controller, but does not enter either TAM-WM training stage.

Regression on demonstrated transitions alone can produce a model that matches observed outcomes while remaining insensitive to action variations. We therefore contrast the executed action $A_t^{(0)}$ with in-batch alternatives $A_t^{(k)}$ drawn from other transitions:

$$\begin{aligned} \ell_k &= -\|\hat{c}_{t+H}(A_t^{(k)}) - c^\star_{t+H}\|_2^2/\tau, \\ \mathcal{L}_{\mathrm{rank}} &= \mathrm{CE}([\ell_0, \ldots, \ell_K], 0), \end{aligned}$$

yielding $\mathcal{L}_B = \mathcal{L}_{\mathrm{dyn}} + \lambda_{\mathrm{DINO}}\mathcal{L}_{\mathrm{DINO}} + \lambda_a\mathcal{L}_{\mathrm{rank}}$. Here $\mathcal{L}_{\mathrm{DINO}}$ reconstructs frozen future DINO features as an auxiliary feature-space regularizer. This objective encourages the dynamics to distinguish the executed action from sampled alternatives. In-batch alternatives are training surrogates, not simulator-verified counterfactual outcomes; physical correspondence is tested by executing candidates from restored states in the physical-ranking experiment (Q3). Network and sampling details appear in the appendix.

From Prediction to Control: Selective Correction

The action-conditioned predictor provides a control interface: local proposals around a nominal policy action can be compared through their predicted future coordinates in the shared state space. In the downstream online policy distillation (OPD) protocol, the Stage-B model serves as a selective online correction teacher. It considers local candidates around A_t^{nom}. For candidate $A_t^{(k)}$, an ensemble predicts a future consensus coordinate and scores its distance to a task-specific goal prototype c_g formed from successful terminal training states:

$$J_t^{(q,k)} = \left\| c_t + T_{\theta_q}(c_t, A_t^{(k)}) - c_g \right\|_2^2.$$

The teacher accepts a correction when the nominal action is judged at risk, the candidate passes uncertainty, damage, and trust-region filters, predicted progress exceeds a gain margin, and intervention budget remains. Otherwise the nominal action executes. This selectivity matters because prediction can be unreliable away from demonstrated states and candidate search adds online cost. A separate distillation stage collects accepted teacher corrections and fits a policy-only student, anchored to the nominal policy, for execution without online search. The appendix gives the full gate, risk-classifier, runtime, and student-training details. Closed-loop evaluation measures the model and selective controller together.

EXPERIMENTS

The evaluation first tests complete-system closed-loop performance, then examines the transition-level mechanism, candidate-sensitive prediction, physical grounding, and selective correction. Fixed-state and fixed-candidate diagnostics remain separate from the paired closed-loop evaluation.

EXPERIMENTAL SETUP

Tasks. We evaluate eight RoboMimic PH, MetaWorld-style, and PandaGym manipulation tasks. All methods share task definitions, test initial states, episode counts, and success criteria, with each row reporting its method-specific configuration. The eight-task success rates describe the reported system configurations; the fixed-state, fixed-candidate protocols separately probe representation and scorer behavior. Our G1 Door Open v3 differs from ReViWo's Sawyer task; the ReViWo entry is a re-evaluation on this Door task, not its published value Pang et al. (2025).

Baselines. The comparisons are BC, DINO-WM, ReViWo, MV-MWM, MVP-LAM, MVISTA-4D, Ctrl-World, and VILA.

Evaluation protocols. The controlled scorer protocol fixes held-out states and $K = 50$ candidates at $H = 8$, evaluating demonstrated-action ranking and physical ordering over 3,200 rollouts from 64 restored Can states. Closed-loop tests use paired resets, with 180 episodes per task and 450 for Door. Tables 2, 3(a), and 4 share the canonical Can checkpoint; separately trained diagnostics, full specifications, and the checkpoint map appear in Appendix Tables 6 and 11.

Metrics. Let g_i^v be the current predictive-coordinate vector for view v and transition i; for raw DINO features, g_i^v is the mean over visual tokens. State Match is the sample mean of cosine similarity between synchronized current vectors, and Delta Match is the corresponding mean between their finite-horizon changes:

$$\mathrm{SM} = \frac{1}{N}\sum_i \cos_{10^{-8}}(g_i^a, g_i^w), \qquad \mathrm{DM} = \frac{1}{N}\sum_i \cos_{10^{-8}}(g_{i,H}^a - g_i^a,\, g_{i,H}^w - g_i^w),$$

where $\cos_{10^{-8}}(x, y) = x^\top y / [\max(\|x\|_2, 10^{-8}) \max(\|y\|_2, 10^{-8})]$. For demonstrated-action ranking, each query has $K = 50$ candidates including the demonstrated action. If s_{ij} is the evaluator's distance score (lower is better), its rank is $r_i = 1 + \sum_{j \neq 0} \mathbf{1}[s_{ij} < s_{i0}]$ and MRR is $N^{-1} \sum_i r_i^{-1}$. Demo MRR is the equal-weight mean over the matched-magnitude, nearest-action, state-matched, and mixed-hard candidate families. For TAM-WM, the score is the Euclidean distance between the predicted and observed future coordinates. Physical Spearman ρ, Top-1 physical rank, and normalized regret use simulator-executed candidate outcomes; the reported confidence intervals resample restored states.

MAIN CLOSED-LOOP RESULTS

We report complete-system closed-loop success over eight tasks under paired resets, with each row showing its corresponding system configuration.

Cross-task closed-loop performance. The complete online system has a 75.85% equal-weight average success-rate point estimate, 5.28 percentage points above the strongest external baseline,

Table 1: Eight-task closed-loop success rates. Avg. equally weights the tasks; Ours denotes the complete online selective-correction system.

Method	Can	Sq.	Tool	Door	Peg	Shelf	Stack	Push	Avg.
BC	74.44%	78.89%	60.56%	78.22%	45.56%	47.78%	75.00%	74.44%	66.86%
DINO-WM	64.44%	77.22%	49.44%	75.33%	27.22%	76.11%	72.78%	76.11%	64.83%
ReViWo	75.00%	73.33%	51.67%	75.56%	28.33%	72.78%	72.22%	77.78%	65.83%
MV-MWM	75.56%	73.89%	48.89%	78.89%	25.56%	76.67%	77.22%	78.89%	66.95%
MVP-LAM	76.67%	76.11%	55.56%	78.22%	28.89%	68.89%	75.56%	80.56%	67.56%
MVISTA-4D	76.11%	77.22%	53.89%	77.56%	30.56%	77.22%	75.56%	79.44%	68.45%
Ctrl-World	77.78%	**79.44%**	56.67%	75.78%	31.67%	78.33%	78.89%	82.22%	70.10%
VILA	77.22%	76.67%	58.89%	76.22%	39.44%	77.22%	77.78%	81.11%	70.57%
Ours w/o transition	76.67%	76.11%	64.44%	79.33%	49.44%	77.22%	76.11%	79.44%	72.35%
Ours w/o OPD	78.89%	78.33%	56.67%	78.44%	47.78%	73.89%	77.78%	78.89%	71.33%
Ours	**81.11%**	78.89%	**67.22%**	**84.00%**	**52.22%**	**80.56%**	**79.44%**	**83.33%**	**75.85%**

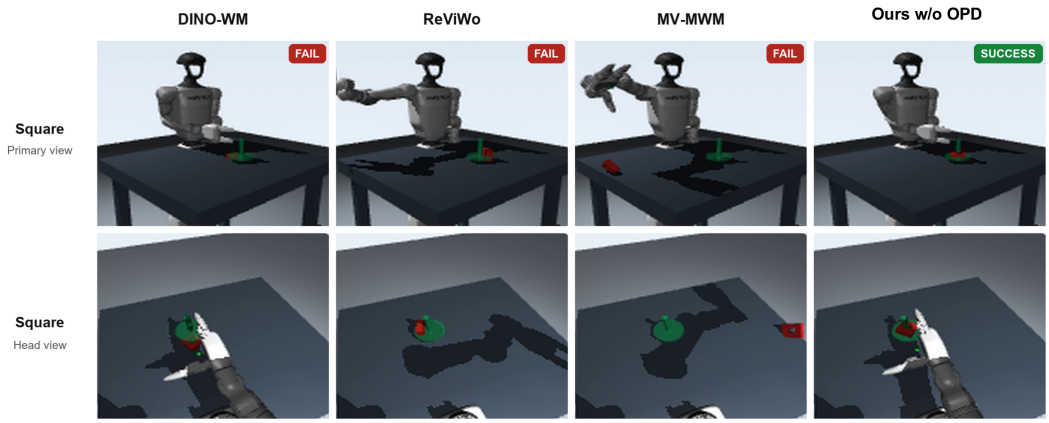

Figure 4: Unitree G1 qualitative Square comparison. Final primary/head views compare four methods under one task protocol; badges reproduce controlled terminal decisions. The Ours panel denotes the phase-conditioned hybrid controller without OPD. The other seven dual-view tasks appear in the appendix. Primary/head display views are distinct from the agent/wrist model inputs.

VILA (70.57%), and 8.99 points above BC (66.86%) across the eight tasks. The difference includes the predictive coordinate, action-conditioned scorer, candidate procedure, and online correction. The within-system no-transition row probes the predictive coordinate, while the no-OPD row retains the learned model without online selective correction. The fixed-candidate experiments that follow separately test predictive quality without intervention. Ours outperforms VILA on all eight tasks. Ctrl-World is higher on Square (79.44% versus 78.89%), while Ours has the highest overall average. Each method and task has one finalized run, so these cross-task point estimates do not measure independent training-seed variation.

The separate matched Door policy-only study and its 1.8-point student–teacher success-rate difference are reported in Appendix Table 9; this point estimate is not claimed significant.

Real-robot demonstration. The multimedia supplement provides anonymized hardware demonstrations of dual-view deployment, selective online correction, and policy-only execution as qualitative feasibility evidence; all reported success rates are simulated.

State Agreement versus Transition Agreement

Approximate state agreement can leave transition residuals. Table 2 distinguishes the observed cross-view properties: State Match summarizes approximate state agreement, whereas Delta Match summarizes transition agreement. DINO-WM has high State Match and very low Delta Match on Can PH. Its wrist-view scorer nevertheless achieves useful demonstrated-action ranking and positive

physical correlation. This counterexample motivates testing the transition objective in the shared TAM-WM coordinate without equating Delta Match with scorer quality.

Table 2: Can PH representation and candidate-ranking results. State Match and Delta Match are mean cross-view cosine similarities for current features and finite-horizon changes; Demo MRR ranks the demonstrated action among $K = 50$ candidates. DINO-WM uses mean-pooled raw features for Match and its wrist-view scorer for ranking. Physical ρ repeats each method's result in Table 4, based on simulator-executed candidates.

Method	State Match ↑	Delta Match ↑	Demo MRR ↑	Physical ρ ↑
Raw DINO-WM	0.9318	0.0012	0.5863	0.3746
ReViWo	0.7852	0.0709	0.1847	-0.0136
MV-MWM	0.2021	0.0357	0.4380	-0.0479
MVP-LAM	0.8260	0.3170	0.2428	0.2584
MVISTA-4D	0.9120	0.4530	0.3104	0.2360
Ctrl-World	0.6680	0.1120	0.3293	0.1030
VILA	0.8820	0.4230	0.3627	0.1240
Ours	**0.9724**	**0.9971**	**0.7057**	**0.4333**

Matched transition intervention. In a matched fixed-capacity control, adding transition alignment leaves State Match nearly unchanged (0.9994 to 0.9998), while increasing Delta Match (0.7862 to 0.9884) and Demo MRR (0.2748 to 0.4522), as summarized in Figure 1. This is a separate matched fixed-capacity Stage-B/scorer experiment, so its absolute values are not directly comparable to the canonical full-model checkpoint. Full controlled diagnostics are reported in Appendix Table 17.

Candidate-Sensitive Prediction and Ablations

Table 3 tests Stage-A supervision under a fixed scorer and Stage-B losses in a fixed coordinate. Panel (b) is a separate control with decorrelation fixed.

Table 3: Can PH ablations. (a) Stage A with a fixed scorer. (b) Stage B with a fixed coordinate.

(a) Stage A objectives

Variant	MRR ↑	ρ ↑
Ours w/o transition	0.3536	0.238
No cross-composition	0.5914	0.236
Capacity-matched adapter	0.3421	0.042
Ours	**0.7057**	**0.4333**

(b) Stage B objectives

Objective	MRR ↑	ρ ↑	Regret ↓
Future only	0.5778	0.3562	0.0592
Future + delta	0.5884	0.3755	0.0576
Future + rank	0.6051	0.3912	0.0514
All losses	**0.6275**	**0.4176**	**0.0483**

Ours has higher Can PH Demo MRR and Physical ρ than the no-transition and capacity-matched variants. All Stage-B losses yield the highest MRR and ρ and the lowest regret among tested variants. In the separate input control (Appendix Table 8), Action MRR is 0.6847 with the full (c_t, A) input, 0.2145 with the action chunk alone, and 0.2934 with shuffled c_t, supporting state-conditioned candidate prediction. The paired MRR gain over DINO-WM has a state-level 95% interval of $[0.1042, 0.1347]$. The corresponding controlled physical diagnostics appear in Appendix Table 17.

DINO-WM remains stronger on generic future-state retrieval, indicating that future-feature retrieval and action utility measure distinct predictive properties; full results are reported in Appendix Table 12.

Physical Grounding

The canonical Can protocol evaluates $K = 50$ candidates from each of 64 restored states at $H = 8$, executing every candidate in the simulator.

Table 4: Can PH physical ranking across $K = 50$ simulator-executed candidates from 64 restored states. CI denotes the reported state-level 95% interval.

Method	$\rho \uparrow$	CI	Top-1 $\uparrow$	Regret $\downarrow$
DINO-WM	0.3746	[0.3269, 0.4218]	**0.3862**	0.0944
ReViWo	-0.0136	[-0.0564, 0.0296]	0.1071	0.3375
MV-MWM	-0.0479	[-0.0923, -0.0035]	0.1346	0.3378
MVP-LAM	0.2584	[0.2205, 0.2961]	0.2908	0.0886
MVISTA-4D	0.2360	[0.163, 0.279]	0.2573	0.1150
Ctrl-World	0.1030	[0.052, 0.154]	0.1427	0.1560
VILA	0.1240	[0.068, 0.172]	0.2053	0.1210
Ours	**0.4333**	**[0.3692, 0.4914]**	0.3807	**0.0606**

On Can PH, Ours has higher full-bank Spearman ρ and lower regret than the listed baselines in this protocol. On Square PH, DINO-WM has higher physical ρ than Ours (0.2638 versus 0.2101), while Ours has the lowest regret (0.3801; Appendix Table 13).

Selective Correction and OPD Analysis

The main closed-loop comparison separates Ours w/o OPD from the complete controller. The Door gate ablation further tests whether selective intervention, rather than correction alone, accounts for the deployment behavior.

Table 5: Door Open v3 OPD gate ablation on the shared 450-episode paired split.

Variant	SR $\uparrow$	Intervene $\downarrow$
Random correction	348/450 (77.3%)	13.6%
Always intervene	350/450 (77.8%)	100.0%
No nominal-risk gate	363/450 (80.7%)	13.5%
No candidate-safety gate	374/450 (83.1%)	13.9%
Ours	378/450 (84.00%)	13.5%

Blind or unconditional correction is weaker than the complete gated controller. Removing either gate also reduces success on the paired Door split, showing that a good predictor alone is insufficient for the final controller. The matched policy-only student is reported as a deployment extension in Appendix Table 9.

Future Work

Future work includes extending TAM-WM to more diverse camera configurations and less tightly synchronized multi-view observations, improving policy-only distillation so that more of the online teacher's gains can be retained without candidate search, and extending the shared predictive coordinate to multimodal goals and longer-horizon action prediction.

Conclusion

TAM-WM learns a shared predictive state coordinate by aligning the observed finite-horizon displacement of the same physical transition across views. With this coordinate frozen, state- and action-conditioned dynamics predict the consequences of alternative candidate actions. Controlled ranking and simulator-executed outcomes test these predictions, while selective correction evaluates their use in closed-loop control. Together, these evaluations connect cross-view transition supervision to action comparison through predicted state changes.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and language polishing, retrieve relevant literature and discover related work. The research assistance included code development, and program execution. The authors take responsibility for the final content of the paper, including its mathematical claims, implementation, experimental results, and conclusions.

REFERENCES

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1801–1810, 2019.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. In *arXiv preprint arXiv:1812.00568*, 2018.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *International Conference on Robotics and Automation*, pp. 2786–2793, 2017.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, 2026. URL https://ctrl-world.github.io/.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8385–8406. PMLR, 2022.

Youngjoon Jeong, Junha Chun, and Taesup Kim. Learning to act robustly with view-invariant latent actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6781–6790, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Jeong_Learning_to_Act_Robustly_with_View-Invariant_Latent_Actions_CVPR_2026_paper.html.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation*, pp. 6023–6029. IEEE, 2019. doi: 10.1109/ICRA.2019.8794127.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020.

Jung Min Lee, Dohyeok Lee, Seokhun Ju, Taehyun Cho, Jin Woo Koo, Li Zhao, Sangwoo Hong, and Jungwoo Lee. MVP-LAM: Learning action-centric latent action via cross-viewpoint reconstruction. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=r0SYyjDhxY.

I-Chun Arthur Liu, Shagun Uppal, Gaurav S. Sukhatme, Joseph J. Lim, Peter Englert, and Youngwoon Lee. Distilling motion planner augmented policies into visual control policies for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 641–650. PMLR, 2022.

Arjun Majumdar, Karmesh Yadav, Soroush Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Jimmy Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. VC-1: A universal visual representation for robot manipulation. In *arXiv preprint arXiv:2306.14846*, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *International Conference on Learning Representations*, 2025.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. In *International Conference on Learning Representations*, 2016.

Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023.

Pierre Sermanet, Corey Lynch, Jasmine Hsu, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *International Conference on Robotics and Automation*, pp. 1134–1141, 2018.

Jiaxu Wang, Yicheng Jiang, Tianlun He, Jingkai Sun, Qiang Zhang, Junhao He, Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao, and Xiangyu Yue. MVISTA-4D: View-consistent 4d world model with test-time action inference for robotic manipulation. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://mercerai.github.io/MVISTA-4D/`.

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

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 of *Proceedings of Machine Learning Research*, pp. 537–546. PMLR, 2022.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

Supplementary Material

Complete G1 Qualitative Outcomes

Figures 5–7 provide the primary and head views for the seven G1 tasks omitted from the main-paper qualitative panel. All panels use the final frame of the same 96-frame rollout protocol and the success decisions in the controlled scorer record. A separate DINO-WM-scored correction rollout appears in Figure 8.

Method Details

Training Tuples and Predictive Data Flow

For each time index t, the training data provide synchronized agent and wrist observations before and after an executed action chunk:

$$\mathcal{T}_t = \left(o_t^a, o_t^w, o_{t+H}^a, o_{t+H}^w, A_{t:t+H-1}\right).$$

The two cameras observe the same physical transition, and the action chunk identifies that transition. A frozen pretrained visual encoder ϕ produces patch-token features $z_t^v = \phi(o_t^v)$ for view $v \in \{a, w\}$. The visual encoder is never updated by either training stage. Stage A maps these features to a transition-relevant coordinate $c_t^v = E_c(z_t^v)$ and an auxiliary context $u_t^v = E_u(z_t^v)$. Stage A sees synchronized visual endpoints but does not receive the action as encoder input. Stage B receives the frozen consensus coordinate and an action chunk, and predicts the corresponding coordinate change. Thus, future observations supervise the target during training but are not available to candidate scoring at inference.

This separation is important for interpreting the experiments. Stage A asks whether two views encode the same observed physical change consistently. Stage B asks whether alternative actions induce distinguishable predicted changes in that fixed coordinate. OPD then asks whether those predictions are sufficiently reliable to replace a nominal policy action. These are separate learning and evaluation questions; the closed-loop results measure their interaction, while the controlled ranking protocols isolate the predictive model.

Stage A: Transition-Aligned Coordinate Learning

State and transition objectives. Let

$$c_t^v = E_c(z_t^v), \qquad \Delta c_t^v = c_{t+H}^v - c_t^v.$$

We use the distance

$$d_\epsilon(x, y) = \|x - y\|_2^2 + 1 - \frac{x^\top y}{(\|x\|_2 + \epsilon)(\|y\|_2 + \epsilon)}$$

to retain both magnitude and directional information. State alignment anchors the synchronized coordinates,

$$\mathcal{L}_{\text{state}} = d_\epsilon(c_t^a, c_t^w),$$

with the explicit state term evaluated at the current endpoint. Future coordinates still enter the displacement construction, whereas transition alignment directly constrains the action-induced displacements,

$$\mathcal{L}_{\text{trans}} = d_\epsilon(\Delta c_t^a, \Delta c_t^w).$$

The second objective is not implied in practice by a small approximate state-alignment loss: it supervises the difference between endpoints, which is the quantity later consumed by the residual dynamics model.

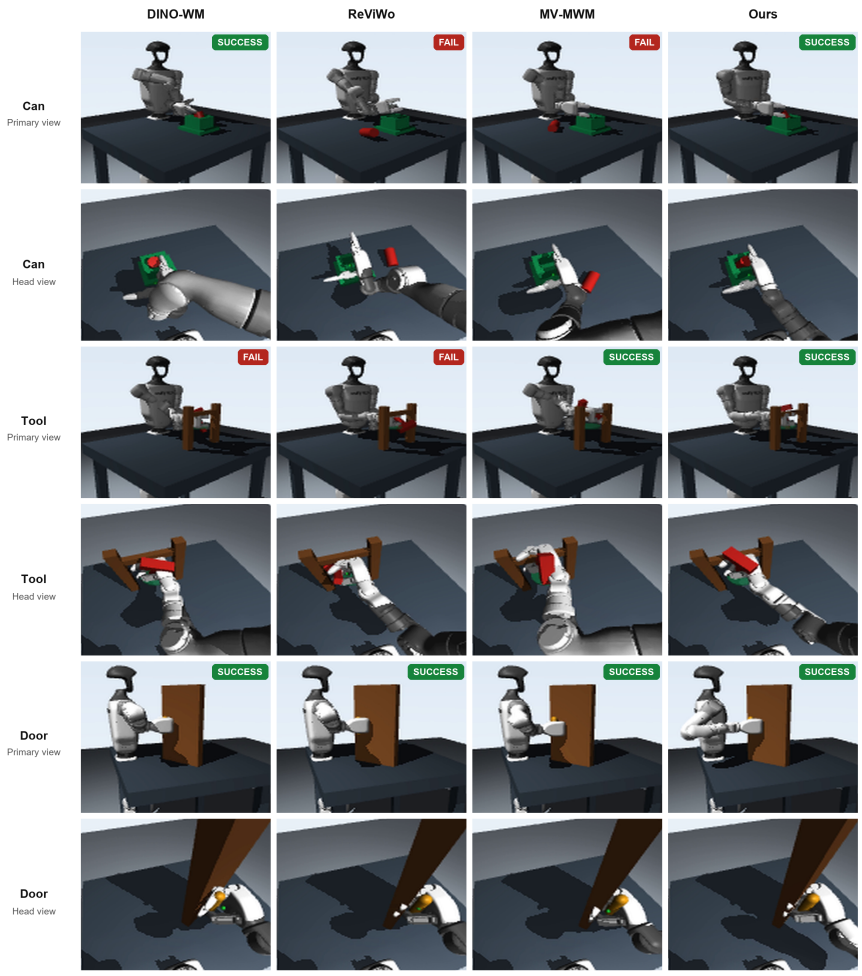

Figure 5: Four-method G1 outcomes for Can, Tool, and Door, part of the seven tasks omitted from the main-paper qualitative panel. Each task is shown from the final primary and head views; terminal success badges are shared across the paired views.

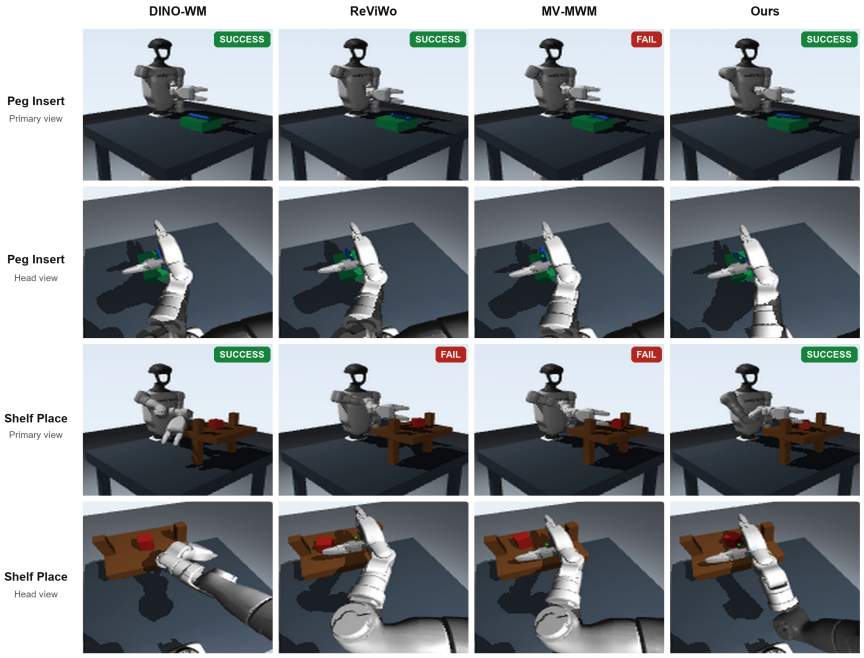

Figure 6: Four-method G1 outcomes for Peg Insert and Shelf Place, part of the seven tasks omitted from the main-paper qualitative panel, using the same dual-view layout and controlled terminal-success labels as Figure 5.

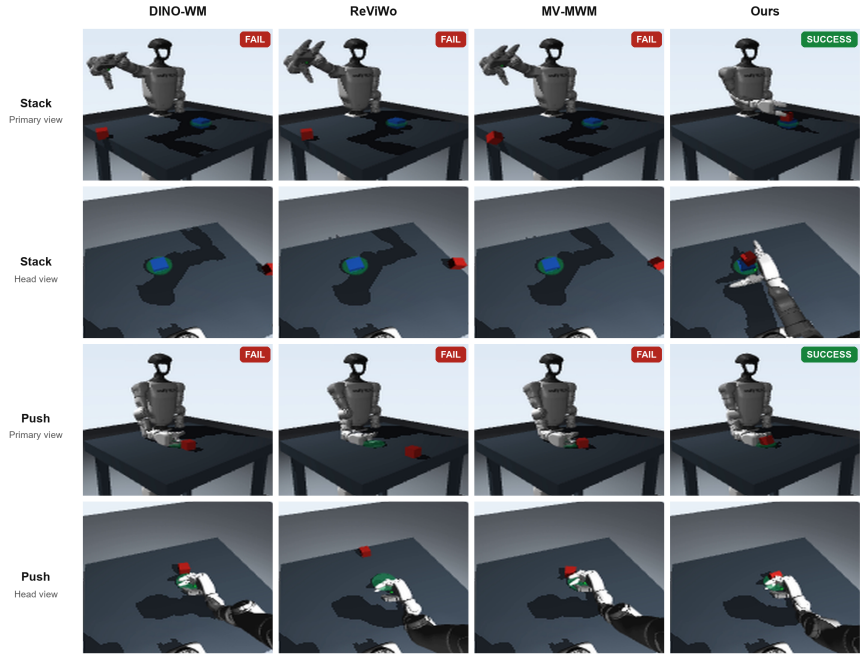

Figure 7: Four-method G1 outcomes for Stack and Push, completing the seven tasks omitted from the main-paper qualitative panel under the same dual-view protocol.

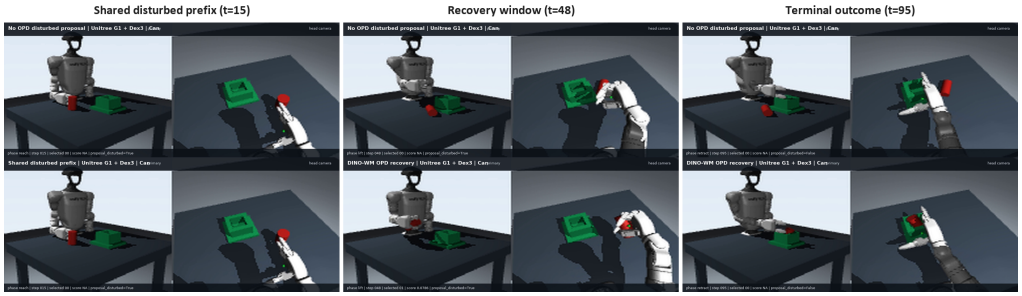

Figure 8: Generic selective-correction illustration on G1 Can using a DINO-WM scorer. From a shared disturbed prefix through $t = 15$, the nominal rollout ends at 0.153 m error; the OPD-style selector intervenes 36 times and ends at 0.003 m. The panels show one rollout, with primary/head display views distinct from the agent/wrist model inputs.

Residual-error derivation. Let $e_t = c_t^a - c_t^w$. Since $\Delta c_t^v = c_{t+H}^v - c_t^v$,

$$\Delta c_t^a - \Delta c_t^w = e_{t+H} - e_t,$$
$$\|\Delta c_t^a - \Delta c_t^w\|_2 \le \|e_{t+H}\|_2 + \|e_t\|_2.$$

Exact endpoint agreement implies transition agreement, but approximate state alignment gives only this loose bound. Its decision relevance is largest when the true action-induced displacement is small relative to the changing endpoint residual.

Cross-composition and anti-collapse routing. State and transition alignment alone admit a collapsed solution. We therefore pair the predictive coordinate with an auxiliary context $u_m^v = E_u(z_m^v)$, where $m \neq t$ is sampled within the same demonstration. For source view v_s and target view v_t, a view-specific decoder reconstructs the frozen target feature:

$$\hat{z}_t^{v_s \to v_t} = R_{v_t}(c_t^{v_s}, u_m^{v_t}),$$

$$\mathcal{L}_{\text{recon}} = \tfrac{1}{2}\left(\lambda_{\text{cross}}\mathcal{L}_{\text{cross}} + \lambda_{\text{self}}\mathcal{L}_{\text{self}}\right),$$
$$\mathcal{L}_{\text{cross}} = \tfrac{1}{2}\left(\ell_{a\to w} + \ell_{w\to a}\right), \qquad \mathcal{L}_{\text{self}} = \tfrac{1}{2}\left(\ell_{a\to a} + \ell_{w\to w}\right),$$

where $\ell_{v_s \to v_t}$ combines MSE and cosine distance between $\hat{z}_t^{v_s \to v_t}$ and $z_t^{v_t}$, as in the implementation. The different time index reduces direct same-frame copying and encourages use of c_t for the current target. It does not guarantee that $u_m^{v_t}$ contains no current-state cues. View-dependent or reconstruction-only information can also be routed through u.

To reduce duplicated information, the implementation uses a partial-coordinate decorrelation regularizer. Let $d = \min(\dim(c), \dim(u))$, and let C_d and U_d contain the first d columns of the respective B-sample batches after per-column centering and ℓ_2 normalization along the batch dimension. Then

$$\mathcal{L}_{\text{decor}} = \text{mean}\left[\left(\frac{C_d^\top U_d}{\max(1, B-1)}\right)^2\right].$$

For the reported Stage A, $\dim(c) = 256$ and $\dim(u) = 64$; this regularizer does not decorrelate every coordinate pair. This term regularizes routing between the coordinates.

Complete Stage-A objective. The representation stage optimizes

$$\mathcal{L}_A = \lambda_s\mathcal{L}_{\text{state}} + \lambda_\Delta\mathcal{L}_{\text{trans}} + \mathcal{L}_{\text{recon}} + \lambda_d\mathcal{L}_{\text{decor}}.$$

After model selection on training/validation data, E_c, E_u, and the view-specific decoders are frozen. No Stage-B or closed-loop gradient is allowed to reshape the coordinate used in the principal experiments.

STAGE B: CANDIDATE-SENSITIVE RESIDUAL DYNAMICS

Residual prediction. The synchronized view-specific coordinates are combined using the aggregation operator defined in the next subsection. Let

$$c_t = \mathrm{Agg}(c_t^a, c_t^w), \qquad c_{t+H}^\star = \mathrm{Agg}(c_{t+H}^a, c_{t+H}^w),$$

and $\Delta c_t^\star = c_{t+H}^\star - c_t$. The action-conditioned model predicts

$$\Delta \hat{c}_t = T_\theta(c_t, A_{t:t+H-1}), \qquad \hat{c}_{t+H} = c_t + \Delta \hat{c}_t.$$

The dynamics loss constrains both the future endpoint and the residual:

$$\mathcal{L}_{\mathrm{dyn}} = d_\epsilon(\hat{c}_{t+H}, c_{t+H}^\star) + \lambda_r d_\epsilon(\Delta \hat{c}_t, \Delta c_t^\star).$$

The endpoint term limits accumulated prediction error, while the residual term directly regularizes the action-induced change. T_θ is Transformer-based and receives a visual predictive-coordinate token and H action-step tokens. Its audited dimensions and training schedule are given in the supplementary implementation details.

Executed-versus-mismatched action ranking. Regression on demonstrated transitions can be accurate on observed actions while remaining locally insensitive to action changes. For each current state, the executed chunk $A_t^{(0)}$ is the positive action and nonzero cyclic shifts of the mini-batch provide mismatched chunks $A_t^{(k)}$. Each candidate is evaluated from the same current coordinate:

$$\ell_k = -\frac{\left\| \hat{c}_{t+H}(A_t^{(k)}) - c_{t+H}^\star \right\|_2^2}{\tau},$$
$$\mathcal{L}_{\mathrm{rank}} = \mathrm{CE}([\ell_0, \ldots, \ell_K], 0).$$

The complete dynamics objective is

$$\mathcal{L}_B = \mathcal{L}_{\mathrm{dyn}} + \lambda_{\mathrm{DINO}} \mathcal{L}_{\mathrm{DINO}} + \lambda_a \mathcal{L}_{\mathrm{rank}}.$$

The nonzero-weight $\mathcal{L}_{\mathrm{DINO}}$ term is mean future frozen-DINO feature reconstruction error for the two views, an auxiliary feature-space regularizer. Additional optional no-action and variance terms exist in the implementation but have zero weight in the reported default configuration. The mismatched chunks are contrastive training surrogates, not simulator-verified counterfactual outcomes. Consequently, physical validity is not claimed from this loss alone; the counterfactual ranking experiment restores the same simulator state and executes every candidate. The held-out hard-negative protocol is also distinct from training and uses magnitude-controlled, nearest-action, and state-conditioned perturbations shared across methods.

SEQUENTIAL OPTIMIZATION AND MODEL SELECTION

Training proceeds in the following fixed order:

1. Cache synchronized multi-view features with the frozen visual encoder.
2. Optimize Stage A using matched endpoints, cross-composition, and decorrelation.
3. Select the Stage-A checkpoint using training/validation data and freeze E_c, E_u, and R_v.
4. Form consensus current and future coordinates and optimize Stage B on executed action chunks plus in-batch mismatches.
5. Construct each task's goal prototype only from successful terminal training demonstrations.
6. Calibrate the OPD risk models, thresholds, candidate bank, and intervention budget before formal evaluation.
7. For the separate distillation experiment, collect accepted teacher corrections on training/validation initial states and fit the policy-only student.

Test outcomes are not used for checkpoint selection, candidate construction, threshold calibration, or student fitting. The canonical Can checkpoint and the independently trained diagnostic checkpoints are mapped explicitly in Table 11.

CROSS-VIEW COORDINATE AGGREGATION

Stage A produces synchronized view-specific predictive coordinates c_t^a and c_t^w. We instantiate the aggregation operator used by Stage B as

$$\mathrm{Agg}(c_t^a, c_t^w) = \frac{1}{2}\left(c_t^a + c_t^w\right).$$

Thus, the current consensus coordinate and the observed future target are

$$c_t = \frac{1}{2}\left(c_t^a + c_t^w\right), \qquad c_{t+H}^\star = \frac{1}{2}\left(c_{t+H}^a + c_{t+H}^w\right).$$

The same aggregation rule is used during Stage-B training, candidate-action scoring, and closed-loop inference, ensuring a consistent consensus coordinate throughout the pipeline.

For a successful terminal observation j, we first form its consensus coordinate

$$\bar{c}_j^g = \mathrm{Agg}\big(E_c(z_j^a), E_c(z_j^w)\big) = \frac{1}{2}\left(E_c(z_j^a) + E_c(z_j^w)\right).$$

The goal prototype for a single-goal task is then

$$c_g = \frac{1}{|\mathcal{D}_g|}\sum_{j\in\mathcal{D}_g} \bar{c}_j^g,$$

where $\mathcal{D}_g$ denotes successful terminal observations from the training demonstrations. Candidate actions are scored by the distance between their predicted future consensus coordinate and c_g.

The nominal setting assumes both synchronized views are available. Missing-view and substituted-view evaluations use the test-time variants specified in the corresponding robustness protocol; they do not change the aggregation rule used to train the complete multi-view model.

OPD and distillation. The online teacher scores each candidate against a task-specific goal prototype c_g, the mean Stage-A coordinate of successful terminal training states. For ensemble member q and candidate k,

$$J_t^{(q,k)} = \left\| c_t + T_{\theta_q}(c_t, A_t^{(k)}) - c_g \right\|_2^2.$$

Single-goal tasks use one prototype; multiple prototypes or learned value functions would be alternatives for multimodal tasks. The controller filters candidates by ensemble uncertainty, candidate damage, and trust-region distance:

$$\mathcal{K}_t^{\mathrm{adm}} = \{k : U_t^{(k)} < \tau_U,\ D_t^{(k)} < \tau_D,\ r_t^{(k)} < \tau_r\}.$$

It selects $k^\star = \arg\min_{k\in\mathcal{K}_t^{\mathrm{adm}}} \bar{J}_t^k$ and intervenes only when the nominal failure-risk, gain-margin, and budget conditions also hold:

$$g_t = \mathbf{1}\Big[\mathcal{K}_t^{\mathrm{adm}} \neq \emptyset,\ F_t > \tau_F,\ \bar{J}_t^{k^\star} < \bar{J}_t^{\mathrm{nom}} - \eta,\ b_t > 0\Big].$$

Here $\bar{J}_t^k = \mathbb{E}_q[J_t^{(q,k)}]$ and $U_t^{(k)} = \mathrm{Var}_q[J_t^{(q,k)}]$; the executed action is $A_t^{\mathrm{exec}} = g_t A_t^{(k^\star)} + (1 - g_t)A_t^{\mathrm{nom}}$.

The student π_ψ learns from accepted online corrections using

$$\begin{aligned}\mathcal{L}_{\mathrm{distill}} = \mathcal{L}_{\mathrm{BC}}(\mathcal{D}_{\mathrm{demo}}) \\ + \lambda_{\mathrm{corr}}\mathbb{E}_{(o_t, A_t^{\mathrm{corr}})\sim\mathcal{D}_{\mathrm{corr}}}\|\pi_\psi(o_t) - A_t^{\mathrm{corr}}\|_2^2 \\ + \lambda_{\mathrm{anchor}}\mathbb{E}_{(o_t, A_t^{\mathrm{nom}})\sim\mathcal{D}_{\mathrm{corr}}}\|\pi_\psi(o_t) - A_t^{\mathrm{nom}}\|_2^2 + 0.1\,\mathcal{L}_{\mathrm{view}}.\end{aligned}$$

$\mathcal{L}_{\mathrm{view}}$ penalizes disagreement among student predictions from demonstration views. The online teacher supplies accepted corrections; the separate distillation procedure collects them and trains the policy-only student. On corrected states the correction and nominal targets jointly encourage local adjustments. Collection uses training/validation initial-state IDs. $\mathcal{D}_{\mathrm{corr}}$ is split before policy fitting and is disjoint by initial-state ID from the paired Door test set; neither teacher corrections nor outcomes from held-out test states enter student training.

Table 6: Eight-task evaluation scope and benchmark provenance. Door Open v3 is an independent instantiation counted within the MetaWorld-style family.

Task	Dataset	Views	Primary evaluation role
Can PH	RoboMimic PH	agent + wrist	representation analysis, hard-negative ranking, and physical ranking
Square PH	RoboMimic PH	agent + wrist	fine-grained alignment and second-task mechanism replication
Tool Hang PH	RoboMimic PH	agent + wrist	long-horizon multi-stage control and error accumulation
Door Open v3	MetaWorld-style	agent + wrist	formal paired closed-loop OPD and deployment
Peg Insert v3	MetaWorld-style	agent + wrist	complete-system cross-task evaluation
Shelf Place v3	MetaWorld-style	agent + wrist	complete-system cross-task evaluation
Panda Stack	PandaGym	agent + wrist	complete-system cross-task evaluation
Panda Push	PandaGym	agent + wrist	complete-system cross-task evaluation

Table 7: Hard-negative candidate-action ranking on Can PH under the shared $H = 8$, $K = 50$ protocol. DINO-WM uses its stronger wrist-view variant.

Method	Magnitude	Nearest	State-cond.	Mixed	Avg.
DINO-WM	0.7695	0.4699	0.5714	0.5342	0.5863
ReViWo	0.3166	0.1138	0.1558	0.1525	0.1847
MV-MWM	0.6323	0.3216	0.4098	0.3881	0.4380
MVP-LAM	0.3874	0.1683	0.2189	0.1966	0.2428
MVISTA-4D	0.4682	0.2297	0.2864	0.2573	0.3104
Ctrl-World	0.4916	0.2468	0.3041	0.2747	0.3293
VILA	0.5279	0.2785	0.3396	0.3048	0.3627
Ours	**0.8583**	**0.6195**	**0.6885**	**0.6567**	**0.7057**

The admissible set and intervention criteria above are fixed before evaluation. F_t is a binary classifier that estimates nominal failure risk using the current visual coordinate; it is trained on episode outcome labels where failure is defined as task-specific termination without success. $D_t^{(k)}$ is a candidate-conditioned empirical risk estimator trained from short-horizon rollout annotations; during data collection, sampled candidate actions are executed for H steps in simulation to label collision, constraint violation, or task-relevant damage. Both classifiers use cross-entropy loss and a single fixed Stage-A gate encoder. A single nominal-risk classifier and a single candidate-safety classifier are trained once per task for our OPD system. Their parameters, input features, thresholds, and annotations are fixed before evaluation. If $N_D^{(j)}$ candidates are annotated for task j, this contributes exactly $HN_D^{(j)}$ simulator steps beyond the demonstration data. This auxiliary interaction budget is accounted for separately from demonstrations and is used by the reported OPD system. Within its evaluation, the candidate bank, rescue candidates, and per-episode intervention budget are fixed before formal rollouts. The gate conditions on both risk signals: F_t identifies states where the nominal action is likely to fail, while $D_t^{(k)}$ filters harmful candidates. Policy-only deployment does not call the transition ensemble.

Supplementary Protocol and Evaluation Tables

Task and Protocol Scope

Detailed Mechanism Evaluations

With the complete Stage-B input, Action MRR reaches 0.6847, compared with 0.2145 from the action chunk alone and 0.2934 after shuffling the state coordinate. The 0.4702 and 0.3913 absolute drops establish that candidate ranking in this diagnostic uses the state-conditioned visual coordinate. Together with the transition-alignment ablation, this control argues against an action-only shortcut under the tested input interventions.

Complete-System Ablations

The main-paper OPD gate ablation reports intervention rate as the fraction of decision opportunities at which the nominal action is replaced. The nominal-risk gate is the F_t threshold; the candidate-

Table 8: Controlled Stage-B input intervention on Can PH. Removing or shuffling the visual coordinate c_t sharply reduces Action MRR, showing that action-only input performs substantially worse in this control. All rows use the same control checkpoint and differ only in the scorer input; this checkpoint is separate from the canonical model.

Scorer Input	Action MRR ↑
A_t only	0.2145
Full input (c_t, A_t)	0.6847
Shuffled c_t	0.2934

Table 9: Matched Door deployment comparison. The policy-only student is evaluated against its online correction teacher under the same deployment protocol.

Method	Success ↑	Deployment	Online search
Nominal BC	79.1%	policy	no
Always intervene	77.8%	online	yes
Online correction teacher	82.3%	online	yes
Distilled student	**84.1%**	policy	no

Table 10: Tool Hang PH closed-loop rates over 180 episodes per method, repeated from the eight-task comparison with Wilson 95% confidence intervals, including both within-system ablations.

Method	Success	95% CI
BC	109/180 (60.56%)	[53.27%, 67.40%]
DINO-WM	89/180 (49.44%)	[42.23%, 56.68%]
ReViWo	93/180 (51.67%)	[44.41%, 58.86%]
MV-MWM	88/180 (48.89%)	[41.69%, 56.14%]
MVP-LAM	100/180 (55.56%)	[48.26%, 62.62%]
MVISTA-4D	97/180 (53.89%)	[46.60%, 61.01%]
Ctrl-World	102/180 (56.67%)	[49.36%, 63.69%]
VILA	106/180 (58.89%)	[51.59%, 65.82%]
Ours w/o transition	116/180 (64.44%)	[57.22%, 71.07%]
Ours w/o OPD	102/180 (56.67%)	[49.36%, 63.69%]
Ours	**121/180 (67.22%)**	**[60.07%, 73.66%]**

safety gate is the admissibility filter based on uncertainty, damage, and trust-region thresholds. Removing the candidate-safety gate reduces Door success from 378/450 (84.0%) to 374/450 (83.1%), a four-episode or 0.9-point decrease. Removing the nominal-risk gate reduces success to 363/450 (80.7%), a 15-episode or 3.3-point decrease. These point estimates indicate that the complete gated controller performs better than either gate removal on this paired split, with the larger degradation following removal of the nominal-risk gate. This ablation alone does not measure training-seed variability or isolate the gates from their interaction. The controlled ranking experiments test scorer quality separately from this within-system gate ablation.

Complete OPD Gate and Calibration

For each task, the fixed multi-view goal prototype c_g is constructed from successful terminal training observations using the parameter-free aggregation defined in the supplementary section "Cross-View Coordinate Aggregation." Let F_t be nominal-failure risk, $D_t^{(k)}$ candidate damage risk, $r_t^{(k)}$ distance from the nominal chunk, and b_t the remaining override budget. The admissible set and empirical intervention gate are

$$\mathcal{K}_t^{\mathrm{adm}} = \{k : U_t^{(k)} < \tau_U,\ D_t^{(k)} < \tau_D,\ r_t^{(k)} < \tau_r\},$$

Table 11: Checkpoint roles for the principal experiments. "Same recipe" means the architecture, objectives, and selection rule are fixed, but the checkpoint is trained separately for the target task.

Evidence	Checkpoint role	Canonical Can weights?
Table 2	Primary Can mechanism result	Yes
Table 3(a)	Canonical reference plus component removals	Ours row: yes
Table 3(b)	Stage-B loss diagnostic	No
Table 8	Matched scorer-input diagnostic	No
Table 17	Matched transition intervention	Separate control
Table 4	Simulator-executed Can evaluation	Yes
Table 1	Complete-system closed-loop models	Method-specific

$$g_t = \mathbf{1}[\mathcal{K}_t^{\mathrm{adm}} \neq \emptyset,\ F_t > \tau_F,\ \bar{J}_t^{(k_t^\star)} < \bar{J}_t^{(\mathrm{nom})} - \eta,\ b_t > 0].$$

The formal Door implementation uses a shared bank of 64 nominal-centered and 36 rescue candidates, for at most $K = 100$ candidates per decision. Every learned scorer receives this bank and the same nominal-risk and candidate-safety gates, intervention budget, and decision opportunities. The Door scorer repeats each candidate action across 16 prediction steps; the environment renders and re-observes at every step. Under the controlled scorer protocol, all methods receive the same $K = 50$ candidate actions and decision opportunities. Other tasks use a task-specific configuration that is fixed and shared across methods within that task.

Comparison Protocol Boundaries

The eight-task comparison in Table 1 fixes task definitions, paired test states, episode counts, and success criteria. Each row reports its method-specific system configuration.

Complementary evaluation protocols. The eight-task evaluation reports closed-loop success under shared test conditions. The controlled scorer evaluation directly fixes held-out states and candidate actions and measures predictive quality without closed-loop intervention. These are distinct outcome levels.

Implementation, Data, and Reproducibility Details

Audited implementation settings. The canonical Can PH mechanism model uses frozen DINOv2 patch-token caches and a prediction horizon of eight control steps. Its Stage-A checkpoint has a 256-dimensional transition representation and a 64-dimensional auxiliary reconstruction context. Stage A is frozen before Stage B, whose audited action-contrast checkpoint is trained for 20,000 updates with $\lambda_{\mathrm{action}} = 0.01$. These values describe the reported Can PH checkpoint with Action MRR 0.7057. It is reused in the primary scorer, Stage-A ablation reference, and physical-ranking evaluation. Closed-loop systems use the same finalized recipe but are trained separately for each task.

Checkpoint and protocol map. Table 11 makes explicit which experiments reuse the canonical Can model and which are independently trained diagnostics.

Negative construction and leakage control. For Stage-B training, each nonzero cyclic shift pairs a state with an action chunk executed elsewhere in the same mini-batch. This construction uses no test-time future observation or simulator outcome to label negatives. The $K = 50$ hard-negative evaluation is separate from this training construction: all methods receive the same precomputed candidate bank, and simulator-grounded ranking restores each state and executes every candidate.

OPD protocol boundary. In the formal Door runtime, the scorer evaluates up to 100 candidates by repeating each candidate action across a 16-step prediction horizon. The outer loop renders and re-observes after each environment step. The selected action can be reused for a configurable burst subject to renewed safety checks; the protocol does not execute a fixed three-action open-loop chunk. The candidate bank, rescue pool, learned risk quantities, goal prototype, thresholds, trust region, intervention budget, and decision opportunities are identical across scorers and fixed before the formal

Table 12: C-SWM-style future-state retrieval on Can PH at $H = 8$. Values are point estimates aggregated over 5 candidate-bank seeds.

Method	Global H@1 ↑	Global MRR ↑	Local H@1 ↑	Local MRR ↑
DINO-WM	**0.8373**	**0.9095**	**0.2606**	**0.4678**
ReViWo	0.0977	0.2339	0.0432	0.1482
MV-MWM	0.5620	0.7161	0.1489	0.3297
MVP-LAM	0.3100	0.4700	0.0900	0.2200
MVISTA-4D	0.6800	0.8000	0.1800	0.3700
Ctrl-World	0.2674	0.4186	0.0783	0.1987
VILA	0.1800	0.3400	0.0600	0.1700
Ours	0.7152	0.8302	0.1930	0.3881

paired rollouts. The environment executes one action before each re-observation, although selection may retain a gated burst action across successive steps. Formal OPD is the verified online deployment mode; the matched policy-only student is evaluated separately, with results reported in Table 9. Complete predictor architectures, calibration splits, threshold values, optimizer settings, batch sizes, and all loss coefficients are reported in the complete training and evaluation configurations included in the accompanying anonymous artifact.

Recent Baseline Implementation and Evaluation

The MVP-LAM, MVISTA-4D, Ctrl-World, and VILA rows report comparisons under the task, held-out-state, candidate-bank, horizon, and success definitions used in their respective tables.

Data, resources, and research practice. The RoboMimic PH demonstrations are previously collected public data, while the remaining evaluations use simulator-generated observations and actions. We collect no personal data or human-subject measurements. The anonymous artifact includes task splits, seeds, training and evaluation configurations, checkpoint identifiers, candidate-bank indices, and paired episode outcomes.

Additional Experimental Analysis

C-SWM-Style Future-State Retrieval

Following the C-SWM retrieval protocol (Kipf et al., 2020), we evaluate the Can PH official validation split using 2164 queries at horizon $H = 8$. Each query contains 100 candidate futures, and results are aggregated over 5 candidate-bank seeds. Global retrieval uses random validation futures, whereas local retrieval uses temporally nearby futures from the same trajectory. The target is the frozen, mean-pooled, ℓ_2-normalized DINOv2 wrist-view future feature. ReViWo, MV-MWM, and our method use a frozen ridge readout fitted on the training split; no world-model parameters are updated.

Square Mechanism Replication

Additional Candidate-Action Results

The category-level action-ranking results are reported in Table 7. They consistently favor the full transition-relevant representation, while the simulator-grounded analysis provides the stronger simulator-grounded check by executing every candidate.

Action-Ranking Statistics

All confidence intervals in Table 14 bootstrap 2,164 held-out transition states, not individual candidates. Paired differences use the same resampled state indices and 100,000 bootstrap replicates.

Table 13: Square PH mechanism replication under the matched state, candidate, and simulator-execution protocol. State/Delta Match summarize cross-view agreement, Action MRR ranks demonstrations, and physical outcomes evaluate candidate ordering.

Method	State Match	Delta Match	Action MRR	Physical ρ	Regret
DINO-WM	0.3460	0.0001	0.3307	**0.2638**	0.4103
ReViWo	0.8698	0.0015	0.1793	0.1462	0.7867
MV-MWM	0.3663	0.0492	0.3673	0.2144	0.4372
MVP-LAM	0.8037	0.2476	0.2718	0.1834	0.4872
MVISTA-4D	0.9186	0.6432	0.3521	0.2043	0.4175
Ctrl-World	0.6573	0.1086	0.3194	0.1917	0.4518
VILA	0.8841	0.4127	0.3076	0.1728	0.4726
Ours	**0.9996**	**0.9954**	**0.4305**	0.2101	**0.3801**

Table 14: Paired state-level bootstrap analysis for hard-negative Action MRR.

Quantity	Point estimate	95% CI
Average MRR, Ours	0.7057	[0.69272, 0.71865]
Ours − DINO-WM	$+0.1194$	$[+0.10418, +0.13469]$
Ours − MV-MWM	$+0.2677$	$[+0.25094, +0.28440]$
Ours − no trans. align.	$+0.3521$	$[+0.33630, +0.36759]$
Ours − no cross-comp.	$+0.1143$	$[+0.10006, +0.12836]$
Ours − capacity match	$+0.3636$	$[+0.34741, +0.37948]$

Table 15: BC-centered candidate-pool coverage and model-ranking error on Square PH over 64 restored states and 50 executed candidates per state.

Selector	Regret ↓	Coverage	Efficiency ↑
Nominal BC	0.3391	1.0000	–
Random	0.3397	1.0000	0.1643
DINO-WM	0.3387	1.0000	0.2383
ReViWo	0.4802	1.0000	0.2758
MV-MWM	0.5370	1.0000	0.1436
Ours	**0.3293**	1.0000	**0.3417**
Physical Oracle	0.0000	1.0000	1.0000

PHYSICAL-RANKING DECOMPOSITION

Spearman ρ, Top-1 physical rank, and regret are bootstrapped over restored simulator states. Candidate Coverage is the fraction of states whose candidate bank contains an action with a better executed physical score than the nominal action. Physical Oracle selects with the executed simulator outcome. Selection Efficiency measures the fraction of the oracle improvement from the nominal action recovered by a selector.

Every state contains a candidate better than nominal BC. Ours has the lowest regret and highest selection efficiency among learned selectors. Candidate coverage alone is therefore not the remaining bottleneck.

MISSING-VIEW ROBUSTNESS

The model is trained with synchronized multi-view input and evaluated with a missing or substituted view; this is not a separately trained single-view comparison.

Table 16: Test-time missing-view robustness of the multi-view-trained Ours system. Each task uses 180 episodes.

Test-time observation	Square PH	Tool Hang PH	Equal-weight SR
Full views	**78.89%**	**67.22%**	**73.06%**
Agent dropped	73.33%	61.67%	67.50%
Wrist dropped	75.56%	65.56%	70.56%
Masked missing view	74.44%	58.89%	66.67%
Held-last missing view	76.11%	58.89%	67.50%

Dropping or substituting a view decreases equal-weight SR by 2.50–6.39 points relative to the Table 1 nominal rates. This evaluation measures sensitivity to missing synchronized inputs rather than a separately trained single-view system.

CONTROLLED TRANSITION INTERVENTION

These controlled runs use a matched fixed-capacity Stage-B/scorer to isolate the effect of transition alignment; their absolute values are therefore not directly comparable to the full-model results.

Table 17: Controlled transition intervention under a matched fixed-capacity Stage-B/scorer protocol. Demo MRR is averaged over three seeds. Absolute values are not directly comparable to the canonical full-model checkpoint.

Metric	State-only	+ Transition
State Match ↑	0.999361	0.999798
Delta Match ↑	0.786187	0.988415
Demo MRR ↑	0.2748	0.4522
Physical ρ ↑	0.1518	0.3017
Regret ↓	0.1442	0.0890

Transition alignment increases mean controlled Demo MRR from 0.2748 to 0.4522 and shared-bank physical ρ from 0.1518 to 0.3017, while reducing regret from 0.1442 to 0.0890.