JEPA-*x*: CROSS-PREDICTIVE PHYSICS GROUNDING FOR FORECASTABLE LATENT DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models plan by predicting the outcomes of candidate actions in a learned representation space. In self-predictive models, however, the encoder and predictor are optimized jointly and can co-adapt to latent transitions that are easy to predict in the training distribution but weakly constrained by the physical evolution of the scene. We introduce the cross-predictive JEPA (JEPA-*x*), which grounds latent dynamics in privileged physical trajectories. JEPA-*x* treats visual observations and physical states as corresponding views of the same action-conditioned trajectory, advances both through a shared predictor, and matches each prediction to the future representations of both modalities. This encourages the action-conditioned predictor to learn a common transition rule across the two views. Privileged physical state is used only during training, leaving a visual-only model at deployment. Empirical results show that JEPA-*x* reduces the rollout drift of a newly fitted predictor from 0.361 to 0.104 and increases mean control success from 53.6% to 78.2% on a multi-task suite spanning six evaluation sub-families. We additionally show that direct full-state regression improves position decodability without improving forecastability or control, indicating that the benefit comes from shaping latent dynamics rather than merely encoding physical variables. Supplementary material: project website.

1 INTRODUCTION

Planning with a latent world model casts control as search in representation space (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025; Hansen et al., 2024): the agent encodes the current scene, predicts the outcomes of candidate actions, and selects the actions whose predicted outcomes most closely approach the goal. This requires the world model to learn representations with reliable action-conditioned dynamics. Unlike DINO-WM, which keeps its pretrained encoder frozen (Zhou et al., 2025), end-to-end self-predictive models such as LeWM learn the encoder and predictor jointly (Maes et al., 2026). The encoder and predictor can therefore co-adapt to a latent transition structure that is easy for the co-trained predictor to model. Although this produces temporal consistency within the learned latent space, it does not directly constrain the transitions to follow the physical evolution of the scene and potentially limits the model's generalization capacity on similar physical interactions but looks distinct.

Privileged physical states have been widely used to shape learned representations and improve downstream behavior through state regression, distillation, cross-modal alignment, and training-time asymmetric critics (Chen et al., 2020; Gupta et al., 2016; Tian et al., 2020; Kumar et al., 2021; Pinto et al., 2018). Despite their differences, these approaches primarily use privileged information to supervise what a representation encodes or how it is used downstream. Our empirical results, however, show that making physical state decodable from visual latents does not necessarily make their action-conditioned evolution easier to predict. This gap reflects a distinction among three properties: decodability asks whether physical variables can be recovered from an individual latent state; forecastability asks whether the representation supports reliable action-conditioned prediction beyond the predictor with which it was trained; and control utility depends on whether predicted outcomes preserve the distinctions needed to rank candidate actions. Improving one property does not necessarily improve the others. We therefore treat visual observations and physical states as corresponding views of the same action-conditioned trajectory, allowing the physical future to constrain how visual representations evolve under action rather than only what physical variables an

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.

individual latent state encodes. We then examine whether this trajectory-level constraint produces more forecastable visual dynamics and whether those dynamics improve control.

We introduce the *cross-predictive JEPA* (JEPA-*x*) to apply this trajectory-level constraint. The same action-conditioned predictor advances latent histories from either modality, and each prediction is matched to future representations in both modalities. The within-modal terms preserve self-prediction, while the cross-modal terms couple the predicted transitions across modalities. Sharing the predictor encourages the two encoders to expose a common action-conditioned transition rule rather than only align individual states. The physical branch is used only during training. At deployment, it is removed, leaving the same visual encoder–predictor architecture and planner as the visual-only baseline.

To test whether the improvement lies in the representation rather than its co-trained predictor, we freeze each visual encoder and train the same predictor from scratch. JEPA-*x* reduces rollout drift from 0.361 to 0.104 under this evaluation. A single model trained across multiple object–interaction pairs also raises mean control success from 53.6% to 78.2%, with improvements across all six evaluation subfamilies. In contrast, regression to a flat vector containing all physical-state inputs used by JEPA-*x* improves decodability but leaves forecastability and control near the visual-only baseline. Ablations further show that correct visual–physical pairing is important for control, while direct cross-modal prediction improves forecastability beyond frame-level alignment.

In summary, our contributions are:

- **Cross-predictive JEPA.** We introduce a vision–physics prediction objective that uses a shared predictor to couple action-conditioned transitions across modalities, with privileged state required only during training.
- **A unified privileged-state interface.** We design a shared rigid-body representation that allows one physical encoder to operate across diverse manipulation scenes without task-specific labels.
- **More forecastable representations and better control.** By evaluating frozen representations with newly trained predictors, we show that JEPA-*x* improves visual-dynamics forecastability at the representation level. These gains are accompanied by higher control success across the multi-task suite.

2 Related Work

Predictive latent world models. World models differ in the structure that their learned representations preserve. Reconstruction-based approaches learn dynamics through pixels or generative latent variables (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2025), whereas value-centric methods shape representations around reward and control objectives (Schrittwieser et al., 2020; Hansen et al., 2024). Representation-predictive methods instead model future features directly: DINO-WM plans in frozen pretrained visual features (Zhou et al., 2025), PLDM learns latent dynamics from reward-free offline data (Sobal et al., 2025), and LeWM jointly learns visual representations and predictive dynamics (Maes et al., 2026). More broadly, joint-embedding objectives learn representations by predicting feature-space targets rather than reconstructing observations (LeCun, 2022; Assran et al., 2023; Bardes et al., 2023; 2024). Our work builds on this predictive paradigm and asks how corresponding physical trajectories can constrain the transition structure learned by a jointly optimized visual encoder and predictor.

Training-time privileged supervision. Learning with privileged information uses signals available during training but unavailable at deployment (Vapnik & Izmailov, 2015). Prior work transfers such information through distillation, cross-modal alignment, state regression, privileged sensing, or asymmetric actor–critic training (Chen et al., 2020; Lee et al., 2020; Gupta et al., 2016; Tian et al., 2020; Kumar et al., 2021; Pinto et al., 2018). These approaches establish several ways to exploit additional physical information during training, but our multi-task setting introduces a further requirement: the privileged representation must retain a consistent meaning across heterogeneous object–interaction configurations. We therefore represent scenes through a shared rigid-body schema based on object geometry, extent, pose, and effector state, allowing a single physical encoder to provide the privileged stream across all configurations without task-specific labels.

Physical grounding of predictive dynamics. Several recent world-model approaches bring privileged physical information closer to the learned dynamics. TWIST transfers state-based dynamics to an image-based student (Yamada et al., 2024), Scaffolder uses privileged sensors during policy learning (Hu et al., 2024), PIGDreamer aligns world-model representations with privileged information (Huang et al., 2025), and Pri4R introduces privileged 4D prediction during vision–language–action training (Kim et al., 2026). Closest to our setting, the concurrent Phys-JEPA imposes physical consistency on latent states and transitions for multivariate time-series forecasting (Nie et al., 2026), while PhyLatent adds training-only physical grounding and future-alignment objectives to the LeWM backbone (Zeng et al., 2026). JEPA-*x* instead treats privileged state as a second online predictive view: histories from either modality predict the corresponding future representations in both modalities, directly coupling visual–physical pairing with action-conditioned evolution. Unlike one-way distillation toward a fixed physical target, both representations remain online, and the privileged branch is removed entirely at deployment. We further study this coupling in the multi-task setting, where one model and one unified privileged-state interface span many object–interaction configurations.

3 PRELIMINARIES

Joint-embedding predictive architectures. Joint-embedding predictive architectures (JEPAs; LeCun, 2022) learn by predicting target embeddings from encoded contexts rather than reconstructing observations. Temporal JEPAs use latent histories and can condition predictions on actions. Encoders and predictors are trained jointly with a mechanism to prevent representational collapse.

LeWM. LeWM (Maes et al., 2026) encodes visual observations as $z_t^o = f_\theta(o_t)$ and predicts future latents from visual history Z_t^o and action chunk a_t. Its training objective is

$$\mathcal{L}_{\text{LeWM}} = \left\| g_\psi(Z_t^o, a_t) - z_{t+1}^o \right\|_2^2 + \beta\Omega(z^o), \tag{1}$$

where Ω is the SIGReg isotropy regularizer (Balestriero & LeCun, 2025). For planning, candidate action sequences are rolled forward in the learned latent dynamics and scored against an encoded goal. We use LeWM as the visual-only baseline and retain its encoder, predictor, and planning procedure.

4 CROSS-PREDICTIVE JEPA

JEPA-*x* extends LeWM with a physical encoder and four within- and cross-modal prediction terms. The physical branch is used only during training and removed at deployment. Figure 1 provides an overview.

Core objective. For each visual trajectory, the simulator provides synchronized privileged physical states. A physical encoder h_ϕ maps each s_t to $z_t^s = h_\phi(s_t)$ with the same dimensionality as z_t^o. The unified interface below describes the instantaneous scene configuration, excluding velocities, contact forces, wrenches, and goal-relative quantities. Motion must therefore be inferred from the state history and actions.

Let Z_t^m denote a latent history from modality $m \in \{o, s\}$. A shared action-conditioned predictor g_ψ advances either modality's history under action chunk a_t, and its prediction is supervised by future representations from both modalities. The complete objective is

$$\mathcal{L}_{\text{JEPA-}x} = \sum_{m,n\in\{o,s\}} \left\| g_\psi(Z_t^m, a_t) - z_{t+1}^n \right\|_2^2 + \beta \left[\Omega(z^o) + \Omega(z^s)\right], \tag{2}$$

where actions modulate the predictor through AdaLN (Peebles & Xie, 2023). The isotropy regularizer Ω is applied independently to each branch.

The four source–target terms retain LeWM's visual self-prediction, learn predictive dynamics within the physical branch, and couple the modalities in both directions. Visual-to-physical prediction matches a transition predicted from visual history to the corresponding physical future; physical-to-visual prediction provides the reverse constraint, keeping the coupling symmetric.

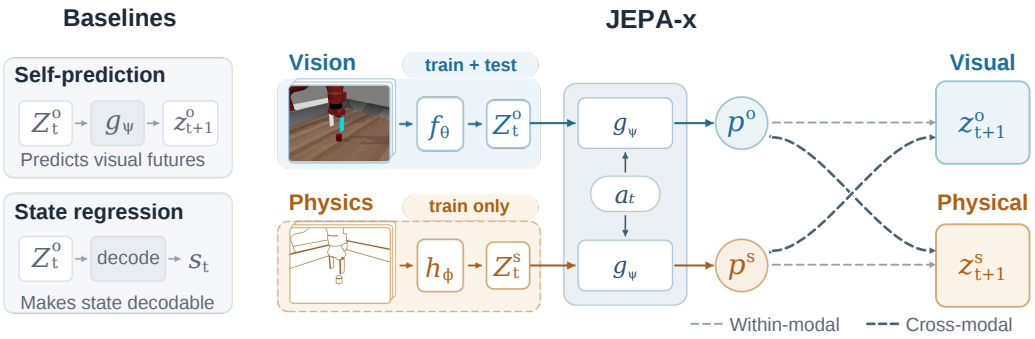

Figure 1: **Cross-predictive physical grounding in JEPA-*x*.** Left: self-prediction and direct state-regression baselines. Right: JEPA-*x* applies a shared action-conditioned predictor g_ψ separately to paired visual and physical histories. Each prediction is matched to both future latents through within-modal (gray dashed) and cross-modal (blue-gray dashed) losses. Geometry thumbnails illustrate physical states; the physical encoder receives structured state tokens. The physical branch (amber) is training-only; the visual encoder (blue) and shared predictor (blue-gray) are retained at deployment.

Importantly, both targets describe the same future scene under the same action: $g_\psi(Z_t^o, a_t)$ must match both z_{t+1}^o and z_{t+1}^s, as must predictions from Z_t^s. The objective couples the modalities through action-conditioned evolution, rather than only aligning individual states. Sharing g_ψ further encourages both modalities to expose a common transition rule under the same actions.

All targets come from online encoders, with f_θ, h_ϕ, and g_ψ optimized jointly. The physical branch is not a fixed teacher or predefined embedding: it adapts with the visual branch under constraints from the privileged trajectory. Independent isotropy regularization prevents either branch from satisfying the prediction losses through collapse.

A unified privileged-state interface across tasks. Grounding across heterogeneous tasks requires a consistent privileged-state representation. We describe every scene using the unified rigid-body schema in Figure 2.

Each object contributes a token

$$\tau_i = [\, G_i \,\|\, e_i \,\|\, R_i \,\|\, t_i \,],$$

where G_i is a fixed-width descriptor of its canonical geometry (a pooled signed-distance field; Park et al., 2019), e_i contains its half-extents, and (R_i, t_i) specifies its pose. An effector token contains the end-effector rotation, position, and finger opening, while a learned table token completes the set. Scenes are padded to six object slots, with unused slots masked from both self-attention and pooling.

Because objects are represented by their geometry rather than their identity, the schema requires neither category labels nor task-specific state fields. A single physical encoder h_ϕ can therefore process every configuration in the suite. Appendix D provides the complete dimensions and normalization procedure.

This consistent physical description enables cross-predictive coupling across configurations, allowing the shared predictor to learn common rigid-body and interaction dynamics without adapting to task-specific state formats.

Corresponding trajectories as a constraint on predictive evolution. The unified interface pairs physical histories and futures with their visual counterparts at every step. Given either visual or physical history, the shared predictor is trained to predict both future visual and physical embeddings. Thus, JEPA-*x* constrains how shared representation geometry evolves under action rather than merely aligning individual visual and physical states.

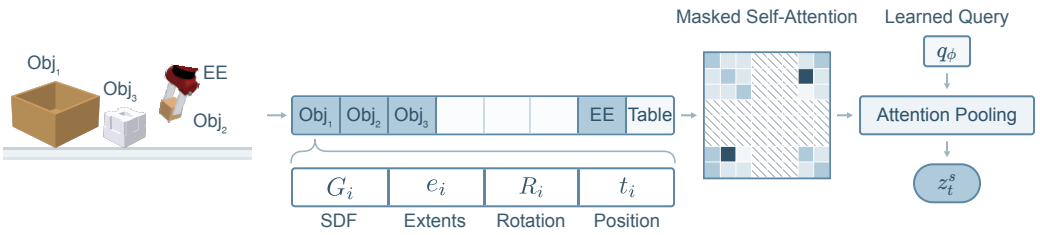

Figure 2: **Unified privileged-state interface.** A scene is represented by a fixed set of object slots, together with end-effector (EE) and table tokens; unused object slots are masked. Each object token encodes SDF geometry G_i, extents e_i, rotation R_i, and position t_i. Masked self-attention followed by learned-query attention pooling maps the resulting token set to the physical latent z_t^s.

Objective decomposition. The interaction between alignment and predictive evolution becomes explicit under uniform weighting of the four prediction terms. Define

$$p^m = g_\psi(Z_t^m, a_t), \qquad \mu_{t+1} = \tfrac{1}{2}\left(z_{t+1}^o + z_{t+1}^s\right).$$

By the parallelogram identity,

$$\sum_{m,n\in\{o,s\}} \left\|p^m - z_{t+1}^n\right\|_2^2 = 2\left\|p^o - \mu_{t+1}\right\|_2^2 + 2\left\|p^s - \mu_{t+1}\right\|_2^2 + \left\|z_{t+1}^o - z_{t+1}^s\right\|_2^2. \tag{3}$$

Equation 3 exposes two consequences of correctly paired cross-prediction. The final term encourages representations of the same future scene to align; the first two require either history to predict that shared future under action. JEPA-x thus combines cross-modal alignment with a constraint on how shared latent geometry evolves. Online targets make this coupling symmetric rather than one-way distillation toward a fixed physical target.

Deployment. Privileged state is used only during training. Discarding h_ϕ at deployment leaves (f_θ, g_ψ) with the same observations, inference architecture, and computational cost as the visual-only baseline.

Starting from encoded observation history $\hat{Z}_0$, a candidate sequence of T action chunks is rolled forward for $i = 0, \ldots, T-1$ and scored by the distance between its predicted terminal latent and the encoded goal:

$$J(a_{0:T-1}) = \left\|\hat{z}_T(a_{0:T-1}) - f_\theta(o_g)\right\|_2^2, \qquad \hat{z}_{i+1} = g_\psi(\hat{Z}_i, a_i). \tag{4}$$

The selected sequence is executed, and planning repeats from the resulting state. Privileged state affects planning entirely through the learned visual representation and dynamics; it is neither observed nor reconstructed at test time. More forecastable dynamics reduce one source of error in this rollout-based score, although successful control also requires latent distance to remain aligned with physical outcomes.

The multi-task suite uses z-CEM: CEM searches the latent space of a small unconditional variational autoencoder trained on action chunks from the same corpus. Decoding candidates into temporally coherent chunks avoids incoherent raw-action candidates far from the training distribution. The autoencoder observes neither task nor scene, and every method uses the same action prior. Single-task experiments retain LeWM's released action-space CEM planner. Appendix E specifies the autoencoder and planner hyperparameters.

5 Experiments

We evaluate JEPA-x through multi-task control, representation forecastability, and objective ablations, followed by separate single-task benchmarks.

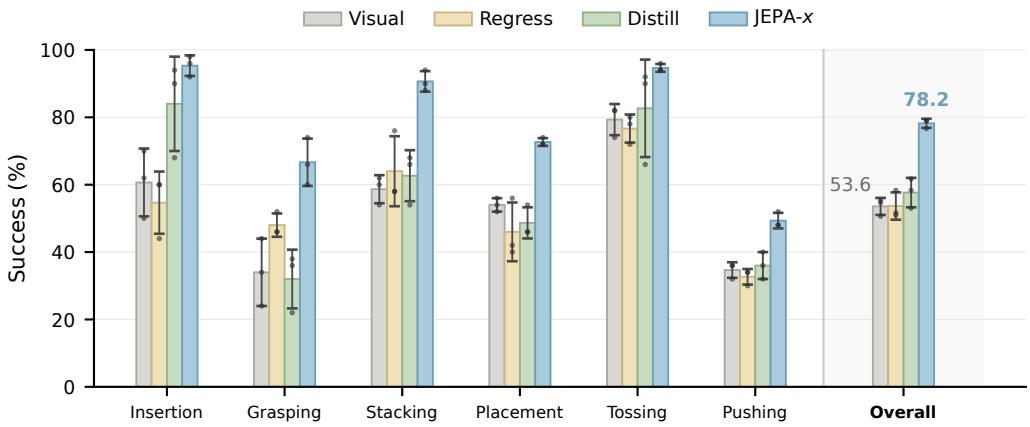

Figure 3: **JEPA-*x* improves control across all six evaluated interaction families.** One model is trained jointly across configurations. Overall success equally weights the six families and increases from 53.6% to 78.2% under the same deployment planner, action prior, and search budget. Bars show means $\pm$ sample SD over three training seeds; dots show individual seeds.

Experimental setup. Our primary evaluation uses a tabletop suite built on Meta-World (Yu et al., 2020; Todorov et al., 2012). One model is trained across 22 object–task configurations spanning 13 assets, using approximately 1.44M transitions from clean and perturbed expert rollouts. Control is evaluated on six interaction families: insertion, grasping, stacking, placement, tossing, and pushing. These families group evaluation subtasks; for example, a grasp–move–place configuration contributes both grasping and placement scenarios. Each family contains 50 held-out scenarios with identical initial states across methods, and overall success averages the families equally. Representation probes use held-out episodes spanning all configurations. Appendix A details the suite and success predicates.

We compare JEPA-*x* with the visual-only LeWM objective (VISUAL; Maes et al., 2026), per-frame regression to the full physical state flattened into a vector (REGRESS), and pointwise matching to stop-gradient physical latents with within-modality prediction (DISTILL). Core comparisons share the training corpus, visual backbone, optimization schedule, and deployment planner. All methods plan from visual observations and a goal image, using the same unconditional action prior and z-CEM budget. The planner executes 25 environment actions before observing the resulting state and replanning. Success is assessed from the executed physical trajectory. Results report means and sample standard deviations over three training seeds; Appendices E and F provide implementation details.

Multi-task control. JEPA-*x* raises overall success from 53.6% to 78.2% (Figure 3), with improvements across all evaluated interaction families. A single model achieves these gains under the same deployment architecture and planning budget as VISUAL, demonstrating the value of training-time physical grounding for visual control. Appendix B reports paired scenario-level tests.

The largest gains occur in insertion, grasping, and stacking, where small errors in relative pose can change whether an object enters a cavity, remains grasped, or is supported. Figure 5 makes this sensitivity concrete: VISUAL fails to preserve the object–support relation after release under fixed actions. Such errors matter when the planner executes a full action chunk before observing the scene again. Gains also extend to placement, tossing, and pushing, while pushing and grasping remain the most challenging families.

Direct full-state regression leaves overall control close to the visual baseline, while pointwise distillation improves insertion but yields less consistent gains across families. In contrast, JEPA-*x* achieves stronger overall performance and improves every family. These results suggest that the control gains do not arise simply from injecting physical state information, motivating a closer ex-

amination of how JEPA-*x* shapes the visual representation for action-conditioned prediction. Full per-family results and paired examples of executed behavior appear in Appendices B and H.

Multi-task representation analysis. To examine predictive structure in the representation, we freeze each visual encoder, discard its co-trained predictor, and fit the same lightweight action-conditioned predictor from scratch. The probe rolls forward for 20 steps on held-out episodes. At each horizon, we normalize the drift error by a temporal-persistence baseline that assumes the latent remains fixed at its initial value throughout the rollout. Lower normalized drift therefore indicates greater forecastability under the same predictor architecture and fitting procedure. Appendix C gives the full procedure.

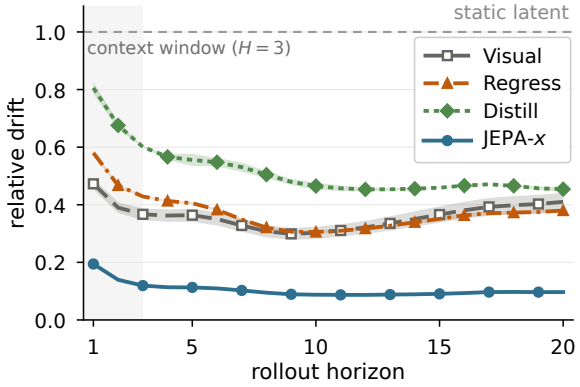

Figure 4: **The forecastability advantage persists with a newly fitted predictor.** Multi-task rollout error relative to temporal persistence, evaluated on frozen visual representations over three seeds.

Table 1: **Position decodability does not track forecastability or control.** Manipulated-object position R^2 of a probe fitted on single frozen visual latent frames of the multi-task suite, mean $\pm$ sample SD over three training seeds. REGRESS and DISTILL decode position best, yet their latent dynamics are the least forecastable and their control matches VISUAL; JEPA-*x* decodes no better than VISUAL while improving both.

Method	Obj. pos. $R^2 \uparrow$
VISUAL	$0.978_{\pm 0.000}$
REGRESS	$\mathbf{0.991}_{\pm 0.000}$
DISTILL	$\mathbf{0.991}_{\pm 0.001}$
JEPA-*x*	$0.978_{\pm 0.002}$

JEPA-*x* reduces relative drift from 0.361 to 0.104, with lower error throughout the rollout (Figure 4). The advantage persists even after the probe's context consists entirely of its own predicted latents. With the visual encoder frozen and the predictor trained from scratch, this indicates that JEPA-*x* learns visual representations that support more accurate autoregressive prediction beyond the original encoder–predictor pairing. These results support the view that grounding predictions in corresponding physical futures constrains encoder–predictor co-adaptation, encouraging visual latent dynamics that are compatible with the scene's physical evolution. The gap persists under rank-controlled projection, suggesting that the organization of the retained features matters beyond their effective dimensionality (Appendix C).

Table 1 adds position decodability for the same four objectives and shows why making physical state readable can leave control unchanged. The probe is fitted separately to each frozen encoder; REGRESS training supervises the full physical-state vector. REGRESS and DISTILL achieve the highest position decodability, yet REGRESS leaves forecastability and control near the visual baseline, and DISTILL drifts more than VISUAL at every horizon (Figure 4), with the largest gap at short horizons where the probe still conditions on ground-truth context. One reading is that matching visual latents to stop-gradient physical latents organizes individual states without making their transitions under actions easy to predict. Both the co-trained visual predictor and a predictor fitted from scratch must learn these difficult transitions in the representation shaped by the physical branch. JEPA-*x* improves forecastability and control with the same position decodability as VISUAL. A nonlinear decoder can recover position even when latent distances poorly reflect physical proximity or transitions remain difficult to predict. The planner, however, directly uses predicted latents and their distance to the goal. Physical grounding helps by organizing the representation for prediction and comparison, beyond making state variables accessible to a separate head. Auxiliary raw-state future prediction also yields little improvement (Appendix G), suggesting that this organization is not automatically induced by predicting a future target.

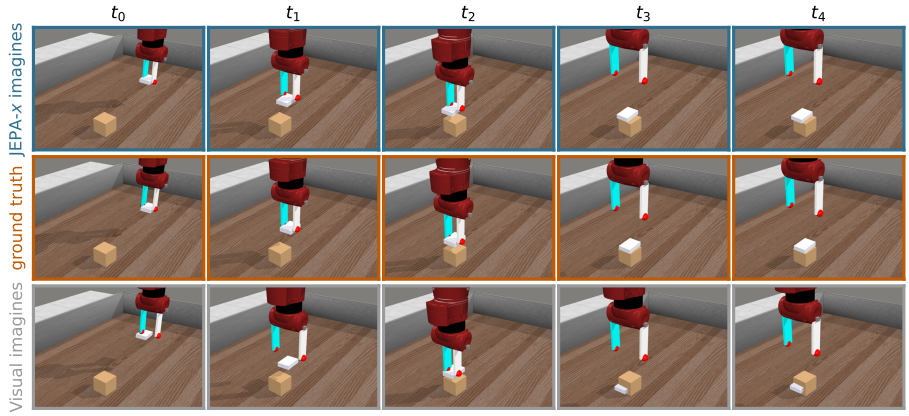

Figure 5: **JEPA-x preserves the object–support relation after release.** Columns show successive times in one stacking episode; rows show JEPA-x, ground truth, and VISUAL. From the same observation and identical expert actions, each model's co-trained predictor produces latent rollouts, decoded into object and end-effector positions for simulator rendering. At t_3–t_4, JEPA-x predicts the eraser remaining on the block, matching ground truth; VISUAL places it beside the block, below the support surface.

Figure 5 exposes an error in a task-defining relation: whether the released eraser remains supported by the block. Both models predict the approach, but VISUAL loses the support relation after release, whereas JEPA-x preserves it as in ground truth. Thus, identical successful actions produce different imagined outcomes. Misrepresenting success as failure can undervalue a useful action sequence when the planner scores predicted terminal latents against a goal. Together with the fresh-predictor results, this qualitatively supports a dynamics-based explanation for the stacking gain from 58.7% to 90.7% under the same planner (Figure 3). These are decoded forecasts, not control executions; Appendix C provides a trajectory-level diagnostic and decoder checks.

Contributions of alignment and cross-prediction. We next examine how alignment, cross-prediction, and predictor sharing contribute to the multi-task results. ALIGN-ONLY replaces cross-modal prediction with symmetric frame-level alignment while retaining shared within-modality prediction. CROSS-ONLY keeps all four prediction terms with separate modality predictors, and SHARE-ONLY uses the shared predictor with within-modality prediction alone. Table 2 summarizes the results; Appendix F specifies the objectives.

Table 2: **Alignment recovers much of the control gain; cross-prediction further improves forecastability.** Multi-task results, mean ± sample SD over three training seeds. All grounded variants retain within-modality prediction. Bold marks the best mean.

Variant	Cross-modal coupling / predictor	Control (%) ↑	Drift ↓
VISUAL	None / visual only	$53.6_{\pm 2.5}$	$0.361_{\pm 0.024}$
SHARE-ONLY	None / shared	$56.7_{\pm 3.5}$	$0.309_{\pm 0.020}$
ALIGN-ONLY	Alignment / shared	$72.6_{\pm 3.6}$	$0.158_{\pm 0.001}$
CROSS-ONLY	Cross-prediction / separate	$73.0_{\pm 0.9}$	$\mathbf{0.101_{\pm 0.001}}$
JEPA-x	Cross-prediction / shared	$\mathbf{78.2_{\pm 1.3}}$	$0.104_{\pm 0.003}$

ALIGN-ONLY recovers much of the control gain, suggesting that alignment already improves the geometry used for planning. By bringing visual states into agreement with physical representations, it makes latent goal distance more sensitive to object configuration and spatial relationships. JEPA-x further reduces drift and achieves higher mean control by coupling this alignment to future prediction. A possible explanation is that frame-level agreement leaves substantial freedom in the transitions between aligned states, while cross-prediction also constrains how those states evolve under actions. This interpretation mirrors the objective decomposition in Section 4: alignment organizes corresponding states, and prediction toward their common future organizes their evolution.

Table 3: **Single-task forecastability improves consistently, while control gains depend on the environment.** One model is trained per environment. Drift uses a predictor fitted from scratch to each frozen encoder. Values are means ± sample SD over three training seeds. Bold marks the best mean within each metric and environment.

	Control success (%) ↑			Relative drift ↓		
Environment	JEPA-*x*	VISUAL	REGRESS	JEPA-*x*	VISUAL	REGRESS
Two-Room	$\mathbf{94.9_{\pm 1.0}}$	$74.7_{\pm 4.1}$	$75.8_{\pm 3.8}$	$\mathbf{0.221_{\pm 0.021}}$	$0.503_{\pm 0.010}$	$0.480_{\pm 0.008}$
OGBench-Block	$\mathbf{78.7_{\pm 2.7}}$	$64.9_{\pm 1.4}$	$70.2_{\pm 0.8}$	$\mathbf{0.269_{\pm 0.007}}$	$0.507_{\pm 0.026}$	$0.393_{\pm 0.018}$
Push-T	$94.1_{\pm 0.4}$	$94.9_{\pm 1.1}$	$\mathbf{96.9_{\pm 2.7}}$	$\mathbf{0.258_{\pm 0.007}}$	$0.399_{\pm 0.118}$	$0.343_{\pm 0.039}$
Reacher	$\mathbf{83.6_{\pm 1.4}}$	$82.7_{\pm 2.1}$	$82.6_{\pm 0.4}$	$\mathbf{0.233_{\pm 0.009}}$	$0.313_{\pm 0.009}$	$0.309_{\pm 0.003}$

CROSS-ONLY preserves most of the control gain and achieves drift comparable to JEPA-*x*, while parameter sharing alone yields little improvement. Cross-modal coupling is thus the main contributor in these comparisons. Interestingly, JEPA-*x* achieves the highest mean control despite slightly higher mean drift than CROSS-ONLY. The probe measures error across the full latent representation under recorded actions, whereas control depends on goal-distance comparisons among candidate actions. Predictor sharing improves the consistency of these goal-directed comparisons even when average probe error changes little. Appendix F further examines mismatched privileged supervision through SHUFFLE, which severely degrades both forecastability and control.

Additional single-task benchmarking. We additionally train one model per environment on *Two-Room*, *OGBench-Block* (Park et al., 2025), *Push-T* (Chi et al., 2025), and *Reacher* (Tassa et al., 2018), using the public datasets and action-space CEM evaluation (Maes et al., 2026). Table 3 combines control and forecastability results; Table 8 reports detailed decodability.

JEPA-*x* achieves the lowest drift in every single-task environment and improves control on Two-Room and OGBench-Block. Mean success is similar to VISUAL on Push-T and Reacher; REGRESS has the highest mean on Push-T, whose strong visual baseline leaves little room for improvement. Improved forecasting changes behavior chiefly when errors distort the ranking of useful actions; lower average drift may matter less when candidate coverage or goal geometry is the stronger bottleneck. This suggests that physical grounding's control value depends on both the magnitude and location of prediction errors in planning. Appendix B reports paired control tests.

6 DISCUSSION AND CONCLUSION

JEPA-*x* grounds visual world models through cross-prediction between paired visual and physical trajectories. Its central contribution is a training objective that couples representation alignment with action-conditioned evolution, using a shared predictor across both modalities. The physical branch shapes the learned dynamics during training and is discarded at deployment, leaving the original visual planning architecture.

The experiments demonstrate the practical value of this design. One jointly trained model improves control across all evaluated manipulation families, and its visual representation remains more forecastable after the predictor is fitted anew. The ablations show that paired alignment contributes substantially to control, while cross-prediction further improves forecastability and achieves higher mean control than alignment alone. Direct full-state regression improves position readout without comparable predictive or control gains, highlighting the importance of how physical information enters the learning objective.

These findings position physical grounding as a way to improve both visual representation learning and model-based manipulation. The consistent forecastability gains across multi-task and single-task settings provide a basis for studying how latent dynamics support goal scoring and action selection. Our current evaluation uses simulation, paired physical trajectories, and a rigid-body interface; extending this approach to real-world observations and broader physical interactions is a natural next step.

AI USE STATEMENT

We used generative AI tools, including LLM-based coding assistants, to assist with code implementation and to provide feedback on ablation study design and evaluation protocols. We also used these tools for grammar correction and language polishing. We did not use generative AI to generate synthetic datasets or formulate the core methodology of this work. The remaining disclosure categories such as developing theoretical models, formulating mathematical claims or writing proofs, translation, and qualitative or thematic data analysis are not applicable. The authors take full responsibility for all content, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

Detailed implementation and evaluation procedures are provided in the appendices. The code is included in the supplementary material and will be released publicly with the camera-ready version. A public download link for the dataset will be provided upon acceptance at ICLR.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. MC-JEPA: A joint-embedding predictive architecture for self-supervised learning of motion and content features. *arXiv preprint arXiv:2307.12698*, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=QaCCuDfBk2.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Proceedings of the Conference on Robot Learning*, volume 100, pp. 66–75. PMLR, 2020.

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*, 44(10–11):1684–1704, 2025. doi: 10.1177/02783649241273668.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2827–2836, 2016.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Edward S. Hu, James Springer, Oleh Rybkin, and Dinesh Jayaraman. Privileged sensing scaffolds reinforcement learning. In *International Conference on Learning Representations*, 2024.

Dongchi Huang, Jiaqi Wang, Yang Li, Chunhe Xia, Tianle Zhang, and Kaige Zhang. PIG-Dreamer: Privileged information guided world models for safe partially observable reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25859–25875. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25ai.html.

Jisoo Kim, Jungbin Cho, Sanghyeok Chu, Ananya Bal, Jinhyung Kim, Gunhee Lee, Sihaeng Lee, Seung Hwan Kim, Bohyung Han, Hyunmin Lee, Laszlo A. Jeni, and Seungryong Kim. Pri4R: Learning world dynamics for vision-language-action models with privileged 4D representation. *arXiv preprint arXiv:2603.01549*, 2026.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems*, 2021. doi: 10.15607/RSS.2021.XVII.011.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf. Version 0.9.2, June 27, 2022.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020. doi: 10.1126/scirobotics.abc5986.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Weizhi Nie, Weichao Liu, Honglin Guo, and Yuting Su. Phys-JEPA: Physics-informed latent world models for multivariate time-series forecasting. *arXiv preprint arXiv:2606.16076*, 2026.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174. IEEE, 2019.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.008.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 43905–43941. Curran Associates, Inc., 2025. doi: 10.52202/085713-1465.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, volume 12356 of *Lecture Notes in Computer Science*, pp. 776–794. Springer, 2020. doi: 10.1007/978-3-030-58621-8_45.

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.

Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16(61):2023–2049, 2015. URL https://jmlr.org/papers/v16/vapnik15b.html.

Jun Yamada, Marc Rigter, Jack Collins, and Ingmar Posner. TWIST: Teacher-student world model distillation for efficient sim-to-real transfer. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9190–9196. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610450.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100, pp. 1094–1100. PMLR, 2020.

Xi Zeng, Haojie Ren, and Ziying Song. PhyLatent: Learning dynamics-relevant representations for JEPA world models. *arXiv preprint arXiv:2608.05720*, 2026.

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. URL https://proceedings.mlr.press/v267/zhou25t.html.

A Suite and Evaluation Details

Suite composition. Our Meta-World-based corpus contains 22 object–task configurations over 13 distinct assets, with 450 clean episodes per configuration before noise augmentation. A single model is trained jointly across all configurations.

Table 4: The 22 configurations in the multi-task suite.

Interaction group	Configurations
Grasp–move–place	battery, block, can, tennis ball
Peg / plug insertion	peg-align-insert, peg-insert-pull, plug-insert
Planar pushing	block, book, eraser
Stacking	battery-on-phone, block-on-book, eraser-on-block
Righting / toppling	stand-up-can, stand-up-thermos; topple-battery, topple-can, topple-thermos
Tossing	battery, block, eraser, tennis ball

Success predicates. Success is determined from the executed physical trajectory rather than latent distance to the goal. Positional tolerances are asset-specific, and orientation is evaluated modulo each asset's declared symmetry.

Table 5: Success predicates for the six evaluated interaction families.

Family	Success condition
Grasping	Object at least 5 cm above its initial table height over the final 1 s, with in-grasp slip at most 1 cm.
Stacking	Support contact between the stacked object and its base after a 2 s settle window, with the base orientation preserved.
Insertion	Insertion depth at least 90% of the target cavity depth, axis tilt at most 3°, and gripper open at termination.
Placement	Object at rest within the asset-specific positional tolerance of the target, with orientation compared modulo symmetry.
Pushing	Object within the asset-specific tolerance of the planar goal, with orientation compared to the initial pose modulo symmetry.
Tossing	Object inside the container interior at the end of the trajectory.

Scenario sampling. Each evaluated interaction family contains 50 held-out scenarios per training seed, giving 300 scenarios per model. Scenarios are stratified across configuration–subtask pairs and sampled with a fixed seed, so all methods are evaluated from the same initial states. The 300 scenarios are drawn from 266 source episodes; statistical tests therefore cluster segments originating from the same episode. Because each family has the same number of scenarios, the overall success rate also equals the unweighted mean of the six family success rates.

The interaction groups in Table 4 label corpus configurations, whereas the six evaluated families are subtask labels annotated within each episode. Scenarios are keyed by configuration–subtask pair, and each family's quota is filled round-robin across every configuration containing that subtask. A grasp–move–place configuration therefore contributes segments to the grasping and placement families, the peg configurations to insertion, and the pushing, stacking and tossing configurations to their like-named families.

For the four LeWM environments, we use the benchmark's released success predicates and evaluation loop unchanged. Each controller is evaluated on the same $N = 150$ paired scenarios.

B Full Control Results

Per-family results. Table 6 reports the per-family results underlying Figure 3, together with the additional SHUFFLE comparison.

Table 6: Per-family control success (%), mean ± sample SD over three training seeds. Families are ordered by the JEPA-*x*–VISUAL gap, matching Figure 3.

Family	JEPA-*x*	VISUAL	REGRESS	DISTILL	SHUFFLE
Insertion	$95.3_{\pm 3.1}$	$60.7_{\pm 10.1}$	$54.7_{\pm 9.2}$	$84.0_{\pm 14.0}$	$16.0_{\pm 6.0}$
Grasping	$66.7_{\pm 7.0}$	$34.0_{\pm 10.0}$	$48.0_{\pm 3.5}$	$32.0_{\pm 8.7}$	$1.3_{\pm 1.2}$
Stacking	$90.7_{\pm 3.1}$	$58.7_{\pm 4.2}$	$64.0_{\pm 10.4}$	$62.7_{\pm 7.6}$	$0.7_{\pm 1.2}$
Placement	$72.7_{\pm 1.2}$	$54.0_{\pm 2.0}$	$46.0_{\pm 8.7}$	$48.7_{\pm 4.6}$	$2.0_{\pm 2.0}$
Tossing	$94.7_{\pm 1.2}$	$79.3_{\pm 4.6}$	$76.7_{\pm 4.2}$	$82.7_{\pm 14.5}$	$1.3_{\pm 2.3}$
Pushing	$49.3_{\pm 2.3}$	$34.7_{\pm 2.3}$	$32.7_{\pm 2.3}$	$36.0_{\pm 4.0}$	$0.0_{\pm 0.0}$
All six	$\mathbf{78.2_{\pm 1.3}}$	$53.6_{\pm 2.5}$	$53.7_{\pm 4.1}$	$57.7_{\pm 4.4}$	$3.6_{\pm 1.3}$

Statistical tests. For each training seed, suite comparisons use an episode-clustered paired sign-flip test with 10^5 Monte Carlo assignments. Scenarios originating from the same source episode share a sign. The per-seed JEPA-*x*–VISUAL differences are $+28.3$, $+22.0$, and $+23.7$ percentage points, while the respective JEPA-*x*–DISTILL differences are $+17.3$, $+18.3$, and $+26.0$ percentage points. All six seed-level comparisons yield $p_{\mathrm{MC}} \leq 10^{-5}$, the smallest value 10^5 assignments can resolve. These tests quantify paired scenario-level differences for each trained model; consistency across training seeds is reported separately.

Single-task comparisons use exact McNemar tests on the same $N = 150$ paired scenarios. JEPA-*x* exceeds VISUAL in every seed on both Two-Room and OGBench-Block, with the largest p-value equal to 0.023.

For the comparison between JEPA-*x* and CROSS-ONLY, the training run is the unit of analysis. Mean suite control is $72.3/74.0/72.7$ for CROSS-ONLY and $79.0/76.7/79.0$ for JEPA-*x*. We report the observed mean difference without claiming statistical significance for this comparison.

C FORECASTABILITY PROTOCOL AND ADDITIONAL RESULTS

Fresh-predictor probe. We freeze each encoder, cache its latents, discard the original predictor, and train a new predictor from scratch. Predictor capacity and fitting procedure are matched across methods within each experimental setting. This tests the predictive structure accessible to a common probe family rather than the performance of the co-trained transition model.

For the single-task experiments, the probe is action-conditioned. Its input concatenates the $H = 3$ most recent latent frames with the aligned $H = 3$ action chunks. A three-layer MLP of width 512 with GELU activations predicts the next latent as a residual update to the most recent frame. Latents are standardized per dimension using statistics from the fitting episodes; actions are used as stored. We train with AdamW at learning rate 10^{-3} and weight decay 10^{-5} for $30{,}000$ steps, with batch size 256 on 1000 episodes and a randomized fit/evaluation episode split.

At evaluation, the probe is rolled out autoregressively for 20 steps, feeding back its own predictions while consuming the recorded action sequence. Normalization by a constant-latent baseline measures the improvement over temporal persistence.

The multi-task results in Table 7 and Figure 4 use the same probe family with a direct rather than residual output head, $20{,}000$ training steps, batch size 512, cosine learning-rate decay, and an episode-level fit/evaluation split.

Probe fitting and evaluation always use disjoint episodes. Relative drift is computed independently at each rollout horizon:

$$\mathrm{drift}(h) = \frac{\mathbb{E}\|\hat{z}_{t+h} - z_{t+h}\|_2}{\mathbb{E}\|z_t - z_{t+h}\|_2}. \tag{5}$$

The reported aggregate is the mean of these per-horizon ratios over $h = 1, \ldots, 20$.

Manipulated-object decoding probe. Decodability is measured using the same frozen encoder checkpoints and episode-level split as the dynamics probe, so the drift and decoding columns of Table 7 evaluate the same representations under matched data partitions. A single three-layer MLP of width 512 with GELU activations maps one frozen latent frame to physical variables; it is trained with AdamW at learning rate 10^{-3} and weight decay 10^{-5} for 6000 steps at batch size 1024, and evaluated on the held-out episodes. The decoding target is a 15-dimensional vector comprising the manipulated object's normalized position, its orientation as a flattened 3×3 rotation matrix, and the normalized end-effector position. We compute $R^2 = 1 - \mathrm{SSE}/\mathrm{SST}$ separately for each target group over all held-out frames, with the total sum of squares taken about the held-out mean. The tables report R^2 for the object-position group. Canonical geometry, bounding-box extents, and masked object slots are not included in this evaluation probe's target. The probe is fitted separately after encoder training and does not define the supervision used by REGRESS, which predicts the full flattened physical state (Appendix F).

Multi-task forecastability and decodability. Table 7 gives the raw rollout error, temporal-persistence denominator, normalized drift, and position decodability; its R^2 column repeats Table 1 and adds SHUFFLE. JEPA-*x* reduces drift without increasing the reported mean position decodability over VISUAL. REGRESS and DISTILL increase position decodability but do not reduce drift. The ordering of position readout therefore differs from the ordering of forecastability.

SHUFFLE has the largest raw rollout error (8.41, against 4.43 for VISUAL) and the largest Copy denominator (15.51, compared with approximately 12 for the other methods in this table). It also has the lowest position decodability, 0.958. Thus, the mismatch intervention degrades position readout, forecastability, and control together. Appendix F explains which correspondences this intervention changes.

Table 7: Forecastability and manipulated-object position decodability on the multi-task suite. Both metrics use the same frozen encoders and held-out split, with a separately fitted probe for each metric. Drift and R^2 are means $\pm$ sample SD over three training seeds; bold marks the best reported mean. Raw and Copy are averaged over horizons and seeds; Drift is the mean of the per-horizon ratios and therefore differs from their ratio.

Arm	Raw ↓	Copy	Drift ↓	Manip.-object pos. R^2 ↑
JEPA-*x*	1.18	12.17	$\mathbf{0.104_{\pm 0.003}}$	$0.978_{\pm 0.002}$
VISUAL	4.43	12.42	$0.361_{\pm 0.024}$	$0.978_{\pm 0.000}$
REGRESS	4.28	12.07	$0.373_{\pm 0.005}$	$\mathbf{0.991_{\pm 0.000}}$
DISTILL	6.11	12.50	$0.516_{\pm 0.004}$	$\mathbf{0.991_{\pm 0.001}}$
SHUFFLE	8.41	15.51	$0.540_{\pm 0.023}$	$0.958_{\pm 0.002}$

Single-task forecastability and decodability. Table 8 preserves the full single-task representation results. JEPA-*x* achieves the lowest drift in every environment, but it does not always have the highest position R^2: REGRESS has the highest mean on OGBench-Block, and all three methods tie at the reported precision on Reacher. The control results are reported alongside drift in Table 3.

Table 8: **Full single-task representation results.** Forecastability uses a predictor fitted from scratch to each frozen representation. Object-position R^2 measures position readout from the frozen visual latent. Values are means $\pm$ sample SD over three training seeds. Bold marks the best mean in each comparison; exact ties are all bold.

	Relative drift ↓		
Environment	JEPA-*x*	VISUAL	REGRESS
Two-Room	$\mathbf{0.221_{\pm 0.021}}$	$0.503_{\pm 0.010}$	$0.480_{\pm 0.008}$
OGBench-Block	$\mathbf{0.269_{\pm 0.007}}$	$0.507_{\pm 0.026}$	$0.393_{\pm 0.018}$
Push-T	$\mathbf{0.258_{\pm 0.007}}$	$0.399_{\pm 0.118}$	$0.343_{\pm 0.039}$
Reacher	$\mathbf{0.233_{\pm 0.009}}$	$0.313_{\pm 0.009}$	$0.309_{\pm 0.003}$
	Object-position R^2 ↑		
Environment	JEPA-*x*	VISUAL	REGRESS
Two-Room	$\mathbf{0.998_{\pm 0.000}}$	$0.921_{\pm 0.011}$	$0.936_{\pm 0.007}$
OGBench-Block	$0.982_{\pm 0.003}$	$0.879_{\pm 0.009}$	$\mathbf{0.983_{\pm 0.001}}$
Push-T	$\mathbf{0.991_{\pm 0.001}}$	$0.944_{\pm 0.009}$	$0.968_{\pm 0.007}$
Reacher	$\mathbf{0.999_{\pm 0.000}}$	$\mathbf{0.999_{\pm 0.000}}$	$\mathbf{0.999_{\pm 0.000}}$

Imagined trajectories under a fixed action sequence. The fresh-predictor analysis measures forecastability independently of the co-trained predictor. As a complementary illustration, Figure 6 visualizes decoded predictions from each model's own predictor on a stacking episode. At the start of each planning cycle, both models are rolled forward from the same observation over the same recorded expert action chunks. The manipulated object's position is decoded from the predicted latents by a probe fitted on training episodes only ($R^2 = 0.984$ and 0.988 for the two models); first-step decoding error remains below 2 cm in every cycle. With identical actions and starting observations, the plots compare decoded rollout behavior; residual state-decoder error remains. Mean terminal error across the three cycles is 1.4 cm for JEPA-*x* and 3.1 cm for VISUAL.

Rank-controlled forecastability. Relative drift compares a fitted predictor against a temporal-persistence baseline, so a representation of lower intrinsic dimension could in principle be easier for a fixed-capacity probe to fit. JEPA-*x* produces a lower-dimensional latent: measured as the participation ratio of the eigenspectrum, its effective rank is $56.5_{\pm 0.2}$ against $95.7_{\pm 2.4}$ for VISUAL, out of 192 dimensions, and this holds for every training seed. To examine whether dimensionality explains the gap, we project each representation onto its leading k principal components and refit the identical probe. The basis is fitted on the training episodes only and then applied to all episodes, so the held-out split does not enter the projection. Everything else is held fixed: the same episodes, split, actions, probe architecture, and optimization budget, with three probe seeds per condition.

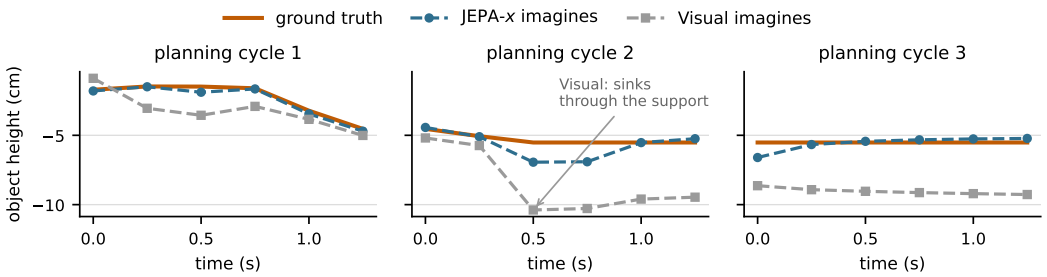

Figure 6: **Under the same actions, JEPA-x predicts the object coming to rest on its support.** Decoded object height during a stacking episode: ground truth against each model's own rollout over identical expert action chunks, one panel per planning cycle. VISUAL's decoded object passes through the support and stays there; JEPA-x has a mean terminal error of 1.4 cm across the three cycles.

Table 9: **Matching effective rank does not close the forecastability gap.** Relative drift after projection onto the leading k principal components, reported as mean ± sample SD over three training seeds.

	Effective rank		Relative drift ↓	
Projection	JEPA-x	VISUAL	JEPA-x	VISUAL
None (native)	56.5	95.7	$\mathbf{0.104_{\pm 0.004}}$	$0.359_{\pm 0.018}$
$k = 57$	51.7	52.8	$0.121_{\pm 0.008}$	$0.453_{\pm 0.022}$
$k = 32$	30.7	30.4	$0.175_{\pm 0.011}$	$0.402_{\pm 0.031}$
$k = 16$	15.7	15.6	$0.193_{\pm 0.011}$	$0.379_{\pm 0.025}$
$k = 8$	7.9	7.9	$0.214_{\pm 0.016}$	$0.407_{\pm 0.033}$

At $k = 57$, chosen to approximate JEPA-x's native effective rank, VISUAL and JEPA-x reach measured ranks of 52.8 and 51.7, with drift of 0.453 and 0.121, respectively. Under more aggressive projection both arms degrade and the ratio narrows, from 3.5× at native width to 1.9× at rank 7.9, but the gap never closes. The retained variance shows why the two arms respond differently: at $k = 57$, JEPA-x retains 95.1% of its variance while VISUAL retains 67.9%, so the same nominal width discards far more of the visual-only representation. Compression never improves the visual baseline, and JEPA-x at $k = 8$ remains more forecastable than VISUAL at any tested width. These results indicate that effective rank alone does not explain the forecastability gap.

D PRIVILEGED-STATE INTERFACE

This section gives the exact dimensions and normalization of the privileged-state interface introduced in Section 4. Each object token concatenates a fixed-width canonical-geometry descriptor, bounding-box half-extents, a flattened 3×3 rotation matrix, and position. Our experiments use an 8^3 average-pooled signed-distance field as the geometry descriptor. The effector token carries end-effector rotation, position, and finger opening, while a learned table token completes the set. Scenes are padded to six object slots with unused slots masked from both self-attention and pooling, giving eight tokens in total. All physical quantities are normalized using training-set statistics.

The privileged input describes only the instantaneous physical configuration and excludes velocities, wrenches, contact forces, and goal-relative quantities. Both branches receive length-H histories, so they infer motion from temporal context and action.

E IMPLEMENTATION DETAILS

Encoders. The visual encoder is a ViT-Tiny (Dosovitskiy et al., 2021) with 192-dimensional features, patch size 14, and 224×224 input resolution (5.50M parameters). The physical encoder

is a three-layer, four-head Transformer of width 192 with hidden width 256 (1.55M parameters), consuming the token set defined in Appendix D.

Predictor and objective. The action-conditioned predictor is a six-layer, 16-head Transformer operating on $H = 3$ latent frames, with head dimension 64, feed-forward width 2048, and dropout 0.1. Actions enter through zero-initialized AdaLN. Each action chunk spans five environment steps with a one-step prediction offset. The four source–target terms in Equation 2 receive equal weight. SIGReg is applied separately to the two branches with weight $\beta = 0.09$, 17 knots, and 1024 projections.

Both future latents are online encoder outputs; the JEPA-*x* prediction path contains no stop-gradient, exponential-moving-average encoder, or separate target network.

Optimization. Core matched methods use AdamW with learning rate 7×10^{-5} and weight decay 10^{-3}, batch size 256, bfloat16 precision, gradient clipping at 1.0, and a linear-warmup cosine-annealing learning-rate schedule. All multi-task models are trained for 40 epochs over the augmented corpus. No image augmentation is applied at training time; input diversity comes from the corpus-level action-noise augmentation described under *Data* below. Training-time auxiliary components differ only where required by the stated objective or ablation.

Planning. The multi-task experiments use a horizon of five macro-steps, each containing five environment actions, with 300 candidates, 30 CEM refinement iterations, and 30 elites (10%). The search distribution is initialized at zero mean and unit variance in the latent action space and refitted to the elites at each iteration, with the current mean always evaluated as one candidate. The planner executes the entire five-macro-step sequence, that is 25 environment actions, before replanning from the resulting state; the executed segment is therefore open-loop and planning is repeated once per segment rather than at every macro-step. The unconditional action autoencoder used by z-CEM is a variational autoencoder with a 32-dimensional latent whose encoder and decoder are two-hidden-layer MLPs of width 512, mapping a flattened 125-dimensional action chunk to and from the latent. It is trained on the same corpus for 20,000 steps with AdamW at learning rate 10^{-3}, weight decay 10^{-5}, and cosine decay, under a reconstruction plus KL objective with KL weight $\beta = 0.1$. It receives neither observations nor goals, and the same action prior is used for every method.

Directly sampling raw action sequences can produce temporally incoherent candidates far from the action distribution seen during training. Searching in the autoencoder latent space instead provides temporal structure: each latent candidate decodes to a coherent action chunk, and CEM refines a distribution over latent codes without conditioning candidates on the current task or scene.

Single-task experiments use the released LeWM action-space CEM planner and its original evaluation budget.

Data. The multi-task corpus is the union of one clean shard and four action-noise shards, giving 29,260 episodes and approximately 1.44M transitions across the 22 configurations. The clean shard contributes 9,900 episodes (450 per configuration) collected with noise disabled. The four noise shards contribute 5,104, 4,752, 4,752 and 4,752 episodes and differ only in noise scale.

Noise is injected into the expert controller during collection rather than added to a recorded trajectory, so every noisy episode is a physically consistent rollout of a perturbed policy. It acts at two levels. At each replanning tick the target waypoint is displaced by $\mathcal{N}(0, \sigma_p^2 I_3)$ and the target yaw by $\mathcal{N}(0, \sigma_{\text{yaw}}^2)$; the displacement is held fixed for the whole replan segment, so replanning cannot average it away, and the perturbed target is then clipped into the reachable workspace. At every control step an Ornstein–Uhlenbeck process perturbs the action itself,

$$x \leftarrow (1-\theta)\, x + \varepsilon, \qquad \varepsilon \sim \mathcal{N}\big(0, \sigma_{\text{step}}^2\, s^2\big), \qquad \sigma_{\text{step}} = \sigma_a \sqrt{2\theta - \theta^2}, \tag{6}$$

with $\theta = 0.15$ and per-dimension scale $s = [1, 1, 1, 1, 0.25]$, so the gripper channel receives a quarter of the amplitude. The stationary standard deviation before action clipping is $\sigma_a |s_j|$ in channel j, equal to $\sigma_a/4$ for the gripper. The sample is added to the controller's output and the sum is clipped to $[-1, 1]$.

The σ values carry a per-family calibration factor k, fitted by bisection so the expert's failure rate on that family lands in a target band. The four shards are the resulting tiers: the band shard targets a 20–30% failure rate, and the s50, s20 and s0 shards target 45–55%, 75–85% and at least 98%, giving $k = 0.0625, 0.0859, 0.1094$ and 0.325 with (σ_p, σ_a) of $(2.4\times10^{-4}, 6.3\times10^{-3})$, $(3.3\times10^{-4}, 8.6\times10^{-3})$, $(4.2\times10^{-4}, 1.1\times10^{-2})$ and $(1.2\times10^{-3}, 3.3\times10^{-2})$. Yaw waypoint noise is disabled throughout ($\sigma_{\text{yaw}} = 0$). Noise is therefore graded by how much it degrades the expert, not by a fixed magnitude, and the corpus spans from a lightly perturbed expert to one that essentially never succeeds.

The LeWM datasets are used as released. Code, the corpus-generation pipeline, and evaluation harnesses will be released.

F Ablation Details

All ablations use the same training data, visual backbone, latent dimensionality, optimization hyperparameters, and training schedule as JEPA-*x*. Predictor parameterization and cross-modal prediction terms are changed only where required by the intervention being tested. Table 10 includes the full comparison, including DISTILL and SHUFFLE, which are omitted from the compact main-text ablation table.

Table 10: **Full multi-task ablation results.** Design columns indicate cross-modal prediction, predictor sharing, and correctly paired cross-modal coupling (✓ present, × mispaired, – absent). Paired coupling includes alignment or distillation as well as cross-prediction; it does not simply indicate whether synchronized streams are available. Values are means ± sample SD over three training seeds; bold marks the best mean.

Variant	Cross-pred.	Shared	Paired coupling	Control (%) ↑	Drift ↓
JEPA-*x*	✓	✓	✓	$\mathbf{78.2_{\pm 1.3}}$	$0.104_{\pm 0.003}$
CROSS-ONLY	✓	–	✓	$73.0_{\pm 0.9}$	$\mathbf{0.101_{\pm 0.001}}$
SHARE-ONLY	–	✓	–	$56.7_{\pm 3.5}$	$0.309_{\pm 0.020}$
ALIGN-ONLY	–	✓	✓	$72.6_{\pm 3.6}$	$0.158_{\pm 0.001}$
DISTILL	–	–	✓	$57.7_{\pm 4.4}$	$0.516_{\pm 0.004}$
SHUFFLE	✓	✓	×	$3.6_{\pm 1.3}$	$0.540_{\pm 0.023}$
VISUAL	–	–	–	$53.6_{\pm 2.5}$	$0.361_{\pm 0.024}$

CROSS-ONLY. CROSS-ONLY retains all four source–target prediction terms from JEPA-*x* but replaces the shared predictor with separate visual and physical predictors. This removes predictor-parameter sharing while preserving corresponding cross-prediction and direct cross-modal transition coupling. It reaches 73.0% control and relative drift of 0.101, showing that most of the control gain and the low drift remain without predictor sharing.

SHARE-ONLY. SHARE-ONLY retains the shared action-conditioned predictor but removes the two cross-modal prediction terms. Each modality is trained only to predict its own future representation through the common predictor. It reaches 56.7% control and relative drift of 0.309, compared with 53.6% and 0.361 for VISUAL.

REGRESS. REGRESS keeps the visual branch and its self-prediction term unchanged and adds a per-frame state-regression head. A two-layer MLP of width 512 maps each visual latent to a flat vector u_t containing all physical-state fields supplied to the privileged encoder: the canonical-geometry descriptor, bounding-box half-extents, rotation, and position of each object, together with the effector's rotation, position, and finger opening. The target retains the six object slots of the unified interface:

$$\mathcal{L}_{\text{reg}} = \frac{\sum_d m_d \left([r_\omega(z_t^o)]_d - u_{t,d}\right)^2}{\sum_d m_d}, \tag{7}$$

where r_ω is the regression head and m_d masks the dimensions of unoccupied object slots; the effector dimensions are always included. Targets use the same training-set normalization as the privileged branch. The term is added to the visual self-prediction and isotropy losses with weight 1.0. No physical encoder or cross-modal prediction is present, and the regression head is unused at deployment.

REGRESS therefore receives the same physical-state information as JEPA-*x*, represented as a flat regression target rather than structured input tokens. This full-state training objective is distinct from the pose-decoding probe used for evaluation; the reported object-position R^2 measures only the position component of that separate probe.

DISTILL. DISTILL replaces predictive cross-modal coupling with pointwise regression. Each branch retains its own within-modality prediction term and is advanced by its own predictor, each followed by its own output projection that maps the predictor's hidden state back to the embedding dimension. These projections belong to the prediction path only: they are applied to predictor outputs and play no part in the cross-branch coupling. Both within-modality prediction terms ($m = n$ in Equation 2) remain active, and the physical encoder is trained by its own prediction term together with the isotropy regularizer. The modalities are coupled by aligning the raw encoder outputs, with no projection on either side, regressing the visual latent onto a stop-gradient physical latent:

$$\|z_t^o - \mathrm{sg}(z_t^s)\|_2^2 . \tag{8}$$

The physical target remains paired with its observation. This variant reaches 57.7% control and drift of 0.516, failing to reproduce JEPA-*x*'s gains. Its comparison with ALIGN-ONLY changes gradient direction, predictor sharing, and prediction-path details, so it does not isolate a single reason for the difference between their results.

ALIGN-ONLY. ALIGN-ONLY keeps the shared predictor and both branches' within-modality prediction terms but excludes the two cross-modal terms from the loss, and adds symmetric alignment:

$$\lambda_{\text{align}} \left\| z^o - z^s \right\|_2^2, \tag{9}$$

with $\lambda_{\text{align}} = 1$. The term is applied to the full encoded window rather than only to the prediction targets, and neither branch is detached. Its 72.6% control success recovers most of the gain over VISUAL, while its drift of 0.158 exceeds JEPA-*x*'s 0.104.

SHUFFLE. SHUFFLE retains all four JEPA-*x* source–target terms and the shared predictor, but reads the privileged stream from a fixed partner episode:

$$Z_t^s \leftarrow Z_t^{s,\pi(e)}, \qquad z_{t+1}^s \leftarrow z_{t+1}^{s,\pi(e)}, \tag{10}$$

where e is the current episode and π is a derangement of the training episodes, drawn once before training and held fixed. No episode is paired with itself, and each visual trajectory sees the same incorrect partner throughout training. Partners are drawn across the whole corpus, so a visual trajectory may be paired with a privileged trajectory from a different configuration; privileged geometry, slot occupancy, and pair-validity masks follow the partner, and the partner window is taken at the same relative position within the partner episode. Because context and future come from the same partner, each privileged sequence remains an intact recorded trajectory. Pixels and actions stay on the true episode, however, so those physical transitions are generally inconsistent with the conditioning actions as well as with the visual stream.

Control falls to 3.6% and relative drift rises to 0.540, compared with 53.6% and 0.361 for VISUAL. This fixed mismatch is more damaging than omitting the privileged branch in the tested construction. This intervention jointly disrupts visual–physical pairing, action consistency, and potentially configuration identity, so the result reflects their combined effect.

G PREDICTING THE PHYSICAL FUTURE IN RAW STATE SPACE

We additionally test whether an auxiliary action-conditioned head predicting the raw physical future can reproduce the gains. Visual self-prediction and SIGReg are unchanged from REGRESS; the auxiliary MLP maps (z_t^o, a_t) to a future target using the same full flattened physical-state vector u_t, instead of predicting the current state from z_t^o. Deployment remains visual-only, with the auxiliary head discarded.

We train two variants for three seeds each at the same budget. FUT-ABS regresses u_{t+1}, while FUT-DELTA regresses $u_{t+1} - u_t$. Both targets use all physical-state fields supplied to the privileged

encoder, including geometry and extents as well as object pose and effector state. The delta target reduces the opportunity to satisfy the objective by decoding the present state. In the corpus's normalized units, u_t already accounts for about 99% of the squared norm of u_{t+1}; the full vector includes time-invariant geometry and extent descriptors alongside pose and effector quantities that can change over the 250 ms interval. The position-decoding results below come from a separate probe on the frozen visual encoder, not from restricting these training targets to pose.

Table 11: Per-family control success (%) for the two future-state auxiliary variants, with VISUAL and REGRESS for reference. Mean ± sample SD over three training seeds, using the same protocol as Table 6.

Family	FUT-DELTA	FUT-ABS	VISUAL	REGRESS
Insertion	$54.0_{\pm 6.9}$	$56.7_{\pm 22.7}$	$60.7_{\pm 10.1}$	$54.7_{\pm 9.2}$
Grasping	$39.3_{\pm 19.4}$	$40.0_{\pm 6.9}$	$34.0_{\pm 10.0}$	$48.0_{\pm 3.5}$
Stacking	$60.0_{\pm 2.0}$	$60.0_{\pm 5.3}$	$58.7_{\pm 4.2}$	$64.0_{\pm 10.4}$
Placement	$52.0_{\pm 4.0}$	$56.0_{\pm 4.0}$	$54.0_{\pm 2.0}$	$46.0_{\pm 8.7}$
Tossing	$72.7_{\pm 4.6}$	$77.3_{\pm 6.4}$	$79.3_{\pm 4.6}$	$76.7_{\pm 4.2}$
Pushing	$30.0_{\pm 3.5}$	$36.7_{\pm 8.1}$	$34.7_{\pm 2.3}$	$32.7_{\pm 2.3}$
All six	$51.3_{\pm 4.8}$	$54.4_{\pm 3.1}$	$53.6_{\pm 2.5}$	$53.7_{\pm 4.1}$

Neither variant reproduces the overall control or forecastability gains of JEPA-*x*. Control success is $51.3 \pm 4.8\%$ for FUT-DELTA and $54.4 \pm 3.1\%$ for FUT-ABS, compared with $53.6 \pm 2.5\%$ for VISUAL and $53.7 \pm 4.1\%$ for REGRESS. Relative drift is 0.359 ± 0.010 and 0.386 ± 0.005, respectively, against 0.361 ± 0.024 for VISUAL and 0.104 ± 0.003 for JEPA-*x*. Manipulated-object position decodability is 0.978 ± 0.001 and 0.986 ± 0.000; the absolute variant improves position readout over VISUAL's 0.978, consistent with a target dominated by the current configuration, without a corresponding improvement in mean control or drift.

These results show that the tested raw-state future-prediction heads are insufficient to reproduce JEPA-*x*'s gains. The comparison evaluates designs that differ in temporal inputs, use of a physical encoder, and prediction losses; ALIGN-ONLY also obtains substantial control gains without direct cross-modal prediction.

H QUALITATIVE ROLLOUTS

Figures 7 and 8 show paired comparisons of executed controllers across all six evaluated interaction families. For each family, both methods start from the same initial state and use the same planner. Each example is selected from scenarios where JEPA-*x* succeeds and VISUAL fails; these illustrations do not estimate the frequency of particular failure modes. Frames are cropped to the working volume because the manipulated object occupies only a small fraction of the uncropped camera view, making criteria such as a 5 cm lift difficult to inspect at print scale. Quantitative comparisons are reported in Figure 3 and Table 6.

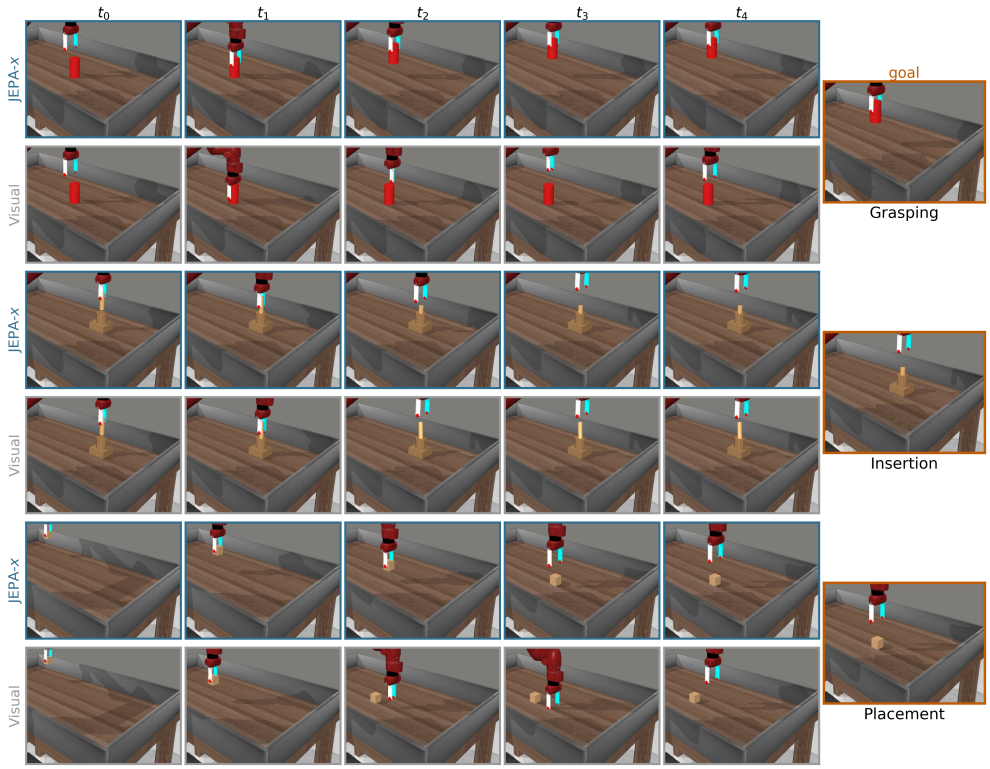

Figure 7: **JEPA-x against VISUAL on the same scenario: grasping, insertion, and placement.** Each family contributes a pair of executed trajectories from the same initial state under the same planner, JEPA-x above and VISUAL below. The shared goal is shown once per comparison. These are selected disagreement cases in which JEPA-x succeeds and VISUAL fails; per-family success rates are in Table 6.

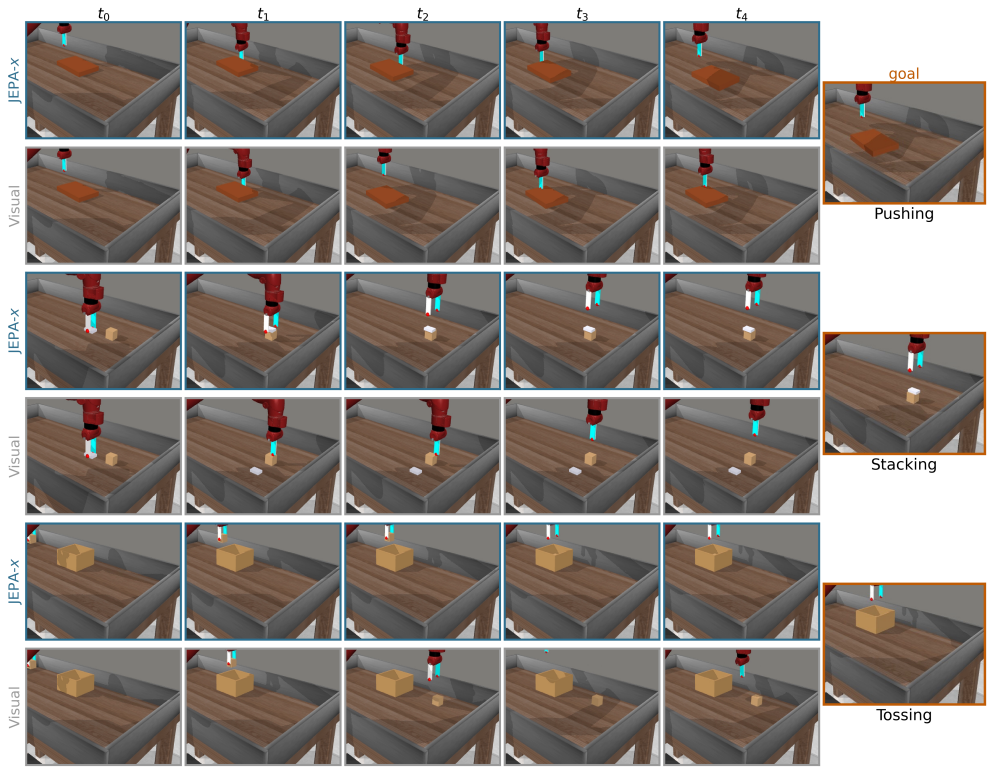

Figure 8: **JEPA-*x* against VISUAL on the same scenario: pushing, stacking, and tossing.** Same protocol as Figure 7: paired executions from one initial state, JEPA-*x* above and VISUAL below, with the shared goal shown once per comparison. In stacking, VISUAL leaves the eraser beside the block; in tossing, it leaves the object outside the carton.