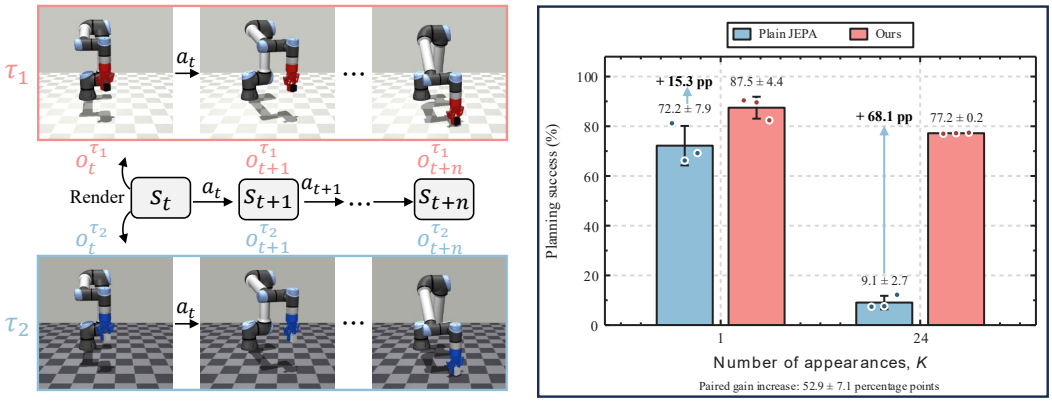
LEARNING INTRINSIC DYNAMICS IN LATENT WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning intrinsic dynamics—the relationship between actions and their consequences—is essential for world models. However, visual variations unrelated to the underlying control process can obscure this relationship in learned representations. We investigate the connection between *action predictability*, defined as the ability to reconstruct an action from consecutive latent states, and the preservation of intrinsic dynamics. Our central hypothesis is that *encouraging action predictability discourages representations from discarding the distinctions between different actions and their consequences*. We provide two complementary forms of evidence for this connection. Theoretically, we derive a positive lower bound on squared inverse prediction error when the encoder merges distinguishable outcomes of different actions from the same state and appearance, and show that forward prediction can still be exact on those transitions. Empirically, controlled appearance-diversity comparisons, supervision interventions, and latent probes support the role of action prediction in preserving control-relevant transition information. We instantiate action predictability with Concat IDM, which predicts actions from concatenated latent states and updates the shared encoder using action labels already available in offline trajectories, without changing the test-time planner. Across three independent training runs on Push-T with 24 appearances, Concat IDM increases planning success from 9.1% to 77.2%; its paired improvement is 52.9 percentage points larger than that observed with a single appearance. Together, these findings support action predictability as an inductive bias toward intrinsic dynamics and demonstrate the practical benefits of Concat IDM for latent planning under visual diversity.

1 INTRODUCTION



(a) Same dynamics, different appearances

(b) Larger gains under visual diversity

Figure 1: **Visual diversity amplifies the benefit of action prediction.** (a) Schematic of appearance variation with physical trajectories held fixed. (b) Under this controlled variation on Push-T, the gain of Concat IDM over Plain JEPA grows from 15.3 pp at K=1 to 68.1 pp at K=24.

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.

An action-conditioned latent world model should capture how the underlying system responds to actions. We refer to this action-conditioned transition structure as **intrinsic dynamics**. This relationship concerns the underlying control process rather than the visual appearance through which it is observed. However, images contain variations unrelated to that process: object color, background, lighting, and rendering style can change observations without changing the physical transition. We refer to these variations as **external factors** (Figure 1a). Learning from visually diverse observations therefore presents a challenge: the representation must retain action-dependent transition information even as the appearance of the same underlying process varies.

When the encoder and predictor are trained jointly, the encoder also defines the prediction targets. Accurate latent prediction does not, by itself, ensure that the representation preserves distinctions in how the underlying system responds to different actions. We therefore investigate whether **action predictability** can help preserve these distinctions. We define action predictability as the ability to reconstruct the executed action a_t from consecutive latent states z_t and z_{t+1}. Predicting actions from transitions is an established inverse-dynamics objective in representation learning and latent planning (Agrawal et al., 2016; Pathak et al., 2017; Paster et al., 2021; Sobal et al., 2025). Our focus is on understanding how this requirement can help preserve the relationship between actions and their consequences when external factors vary.

Our central hypothesis is that *encouraging action predictability helps representations distinguish the consequences of different actions*. Suppose two different actions taken from the same state lead to different outcomes. If the encoder maps both outcomes to the same latent state, the latent transition no longer tells us which action was taken. Training the model to predict the action therefore encourages the encoder to keep these outcomes distinct.

We provide two complementary forms of evidence connecting action predictability to learning intrinsic dynamics. **Theoretically**, if the encoder maps distinguishable outcomes of different actions from the same state and appearance to the same latent state, squared inverse prediction error has a positive lower bound, while forward prediction can remain exact (Proposition 3.1). Action prediction thus penalizes the loss of action–consequence distinctions that forward prediction may overlook. **Empirically**, controlled comparisons show larger planning gains from action prediction under greater appearance diversity, with physical trajectories held fixed. Supervision interventions link these gains to encoder updates from correctly paired action labels, while frozen-encoder probes show that actions are decoded more accurately from transitions than from either endpoint alone. Together, these findings support action predictability as an inductive bias toward intrinsic dynamics, rather than a guarantee of complete dynamics recovery.

We instantiate action predictability with **Concat IDM**, which predicts a_t from concatenated latent states $[z_t; z_{t+1}]$. Its inverse-dynamics loss updates the shared encoder, so that action supervision directly constrains the representation used by the world model. The method uses action labels already available in offline trajectories and leaves the test-time planner unchanged. We implement Concat IDM on the LeWorldModel backbone (Maes et al., 2026) and compare it against Plain JEPA, which uses the same backbone without the inverse-dynamics objective. Across three independent training runs on Push-T with 24 appearances, Concat IDM increases planning success from 9.1% to 77.2%. Its paired improvement over Plain JEPA is 52.9 percentage points larger than that observed with a single appearance (Figure 1b). Because the physical trajectory pool is held fixed, this comparison shows that the benefit grows sharply with visual diversity in this controlled setting. These results demonstrate the practical value of Concat IDM while complementing the theoretical and empirical evidence for action predictability as an inductive bias toward intrinsic dynamics.

The **central contribution** of this paper is to clarify the connection between action predictability and intrinsic dynamics in latent world models. Specifically, we **(1)** hypothesize that encouraging action predictability helps preserve distinctions between the consequences of different actions that action-conditioned forward prediction alone may overlook; **(2)** provide theoretical and empirical evidence supporting this hypothesis; and **(3)** instantiate this principle with Concat IDM and evaluate its effects on learned representations and downstream planning.

2 Related Work

Learning latent dynamics. Latent world models support online planning in PlaNet, behavior learning through imagination in Dreamer, and task-oriented predictive control in TD-MPC (Hafner et al., 2019; 2020; Hansen et al., 2022). Self-predictive representations also use multi-step latent prediction as an auxiliary objective for visual RL (Schwarzer et al., 2021). Joint-embedding predictive architectures learn prediction targets in representation space without pixel reconstruction (Assran et al., 2023; Bardes et al., 2024); V-JEPA 2 extends video pretraining to action-conditioned robotic planning (Assran et al., 2025). DINO-WM predicts frozen DINOv2 features (Zhou et al., 2025), whereas PLDM and LeWorldModel jointly learn the encoder and latent dynamics (Sobal et al., 2025; Maes et al., 2026). PLDM already combines inverse dynamics with VICReg (Bardes et al., 2022); LeWorldModel combines next-embedding prediction with SIGReg (Balestriero & LeCun, 2025). Concurrent Temporal-Distance JEPA addresses the mismatch between short-horizon representation learning and multi-step planning costs (Bai & Xiong, 2026). We study a different axis: how visual diversity over fixed physical trajectories affects jointly learned representations and the benefit of inverse supervision.

Inverse dynamics and action predictability. Inverse dynamics provides supervision for representations of physical interaction and control (Agrawal et al., 2016; Pathak et al., 2017; Paster et al., 2021). Multistep inverse models underlie provable RL with exogenous distractors (Efroni et al., 2022), control-endogenous state discovery in AC-State (Lamb et al., 2023), and offline agent-centric representation learning in ACRO (Islam et al., 2023). ACDF identifies failures of multistep inverse prediction alone and combines it with forward modeling (Levine et al., 2024). Our analysis isolates how inverse prediction penalizes the merging of distinguishable action consequences. Concurrent Delta-JEPA predicts actions from latent displacements (Zhang et al., 2026); our matched Δz-IDM comparison changes only the inverse input, holding the backbone, regularizer, data, and planner fixed. Concurrent ACID uses inverse dynamics at decision time to score imagined transitions (Seo et al., 2026). In contrast, our inverse loss constrains the shared encoder during training; the inverse head is discarded for planning.

External visual factors. Domain randomization and adversarial domain learning address visual variation (Tobin et al., 2017; Ganin et al., 2016). CURL learns contrastive representations from augmented views (Laskin et al., 2020a), while RAD and DrQ improve pixel-based control through data augmentation (Laskin et al., 2020b; Kostrikov et al., 2020). Bisimulation-based approaches preserve control-relevant state equivalence (Zhang et al., 2021); Toso et al. (2026) apply this principle over pretrained visual features for latent world-model planning under background changes and distractors. Concurrent ACPC diagnoses predictive consistency between matched clean and perturbed histories under the same actions (An et al., 2026). We study action–consequence distinctions during joint encoder–predictor training by varying visual appearances over fixed offline trajectories, without an explicit appearance-invariance objective or matched-view supervision.

3 Theory and Method

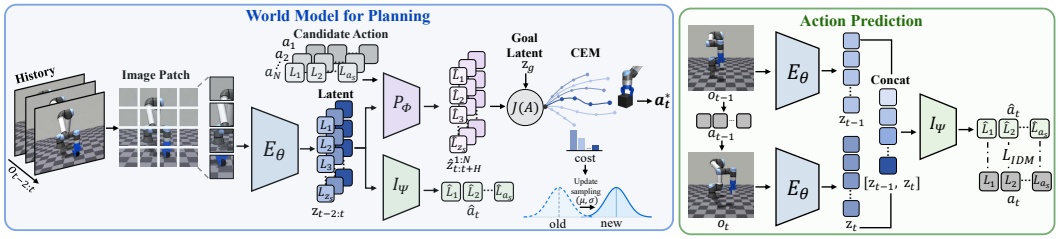
3.1 Intrinsic dynamics and external factors

Let s_t denote the physical state, a_t the executed action, and c the factors that affect only the visual observation. For the controlled environments considered here,

$$o_t = R(s_t, c), \qquad s_{t+1} = F(s_t, a_t). \tag{1}$$

The action-conditioned transition structure induced by F is the **intrinsic dynamics**; the variables in c are **external factors** because they alter the rendering without altering the physical transition.

We construct appearance-diverse datasets by rendering a fixed pool of control trajectories with a finite palette of size K. Each trajectory receives one rendering, and matched renderings of the same trajectory are not provided during training. The physical state and action sequences are unchanged as K varies; increasing K redistributes the same control data across more visual realizations. We also vary the episode-level assignment while holding the physical trajectory pool and palette fixed (Section 4.3).



(a) World model and planning

(b) Action supervision

Figure 2: **Action-supervised latent world modeling and planning.** E_θ: visual encoder; P_ϕ: latent predictor; I_ψ: inverse dynamics head. (a) CEM ranks predicted rollouts by terminal latent goal cost. (b) Concat IDM predicts actions from concatenated observed latents, supervising the shared encoder during training.

3.2 Action predictability and transition distinctions

Action predictability constrains the information retained between consecutive states: their encodings should remain predictive of the executed action. To characterize this constraint, consider a fixed encoder E_θ and an inverse decoder g that predicts actions in $\mathbb{R}^{d_a}$. Fix a physical state s and appearance c in Eq. (1). Let $o = R(s, c)$ and $o_i^+ = R(F(s, a_i), c)$ for distinct actions a_1, a_2. Assume $o_1^+ \neq o_2^+$, so the two outcomes are distinguishable before encoding. Consider a distribution assigning probabilities $p \in (0, 1)$ and $1 - p$ to these two transitions, and define

$$u_i = [E_\theta(o); E_\theta(o_i^+)], \qquad \mathcal{R}_{\text{pair}} = \frac{p\|g(u_1) - a_1\|_2^2 + (1-p)\|g(u_2) - a_2\|_2^2}{d_a}. \tag{2}$$

Proposition 3.1 (Penalty for merging distinguishable outcomes)**.** *Under the conditions above, if $E_\theta(o_1^+) = E_\theta(o_2^+)$, then every inverse decoder satisfies*

$$\mathcal{R}_{\text{pair}} \geq \frac{p(1-p)}{d_a}\|a_1 - a_2\|_2^2 > 0. \tag{3}$$

The bound is attained by predicting $\bar{a} = pa_1 + (1-p)a_2$ at the shared latent input. For this pair alone, a forward predictor can nevertheless attain zero latent prediction error by outputting the common successor encoding for both actions.

Proposition 3.1 isolates an information-loss mode of latent prediction: distinct physical outcomes can share an exactly predictable latent target, whereas inverse supervision penalizes this merger. The result holds at a fixed appearance; Section 4.3 tests empirically how the benefit of action supervision changes as appearance diversity varies. Appendix A provides the proof, a general decomposition of inverse prediction error, and an extension to nearby encodings.

3.3 Concat IDM and training objective

We use an action-conditioned JEPA with a shared visual encoder and a latent predictor. Figure 2 summarizes the planning pipeline and action-supervised training. Our implementation follows LeWorldModel (Maes et al., 2026). The encoder produces $z_t = E_\theta(o_t)$, and the predictor forecasts

$$\hat{z}_{t+1} = P_\phi(z_{\leq t}, a_{\leq t}), \tag{4}$$

where the observation history is limited to three frames. Here and in the training losses below, a_t denotes a standardized action-block vector; the preprocessing and temporal sampling convention are specified in Appendix B. The next-embedding prediction loss is

$$\mathcal{L}_{\text{pred}} = \frac{1}{N_{\text{pred}} d_z} \sum_t \|\hat{z}_{t+1} - z_{t+1}\|_2^2, \tag{5}$$

where N_{pred} counts the predicted transitions and d_z is the embedding dimension. As in LeWorldModel, the same encoder produces context and target embeddings, and gradients pass through both

branches; there is no target network or stop-gradient in this loss. We use SIGReg (Balestriero & LeCun, 2025) to discourage representational collapse by regularizing random one-dimensional projections of the encoded latents toward a standard Gaussian distribution. We denote this regularizer by $\mathcal{L}_{\mathrm{SIG}}$; implementation details are given in Appendix B.

We instantiate action predictability with an inverse dynamics head I_ψ. Concatenation makes both endpoint representations available to the decoder without restricting it to their displacement. Our main variant, **Concat IDM**, predicts

$$\hat{a}_t = I_\psi\big([z_t; z_{t+1}]\big), \qquad \mathcal{L}_{\mathrm{idm}} = \frac{1}{N_{\mathrm{idm}} d_a} \sum_t \|\hat{a}_t - a_t\|_2^2 , \tag{6}$$

where N_{idm} counts adjacent latent pairs and d_a is the action-target dimension. Both squared losses are averaged over latent pairs and target coordinates. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{pred}} + \beta\, \mathcal{L}_{\mathrm{SIG}} + \lambda\, \mathcal{L}_{\mathrm{idm}}, \tag{7}$$

with SIGReg weight $\beta{=}0.09$ and IDM weight $\lambda{=}1$ in the main experiments. SIGReg regularizes the marginal latent distribution, while the inverse loss supervises action-dependent relations between endpoint encodings. The inverse loss updates the IDM head and propagates through both endpoint encodings to the shared encoder; it does not update the forward predictor directly.

Δz-IDM. To compare input parameterizations, we replace the concatenated endpoints by their displacement and train $I_\psi(z_{t+1} - z_t)$ while keeping the remaining model and objective fixed.

Intervention ablations. (i) the stop-gradient intervention replaces both IDM inputs with detached latents, allowing the inverse head to train while preventing its objective from updating the encoder. (ii) the shuffled-target intervention permutes action labels within each batch, preserving the architecture while breaking their correspondence with the transitions. Together, these interventions test whether the gain depends on representation updates induced by correctly paired action supervision.

3.4 Latent planning

At test time, the encoder and predictor are frozen. Given a goal observation o_g, we compute $z_g = E_\theta(o_g)$. Starting from the encoded observation history, the predictor rolls each candidate action sequence $A = (a_t, \ldots, a_{t+H-1})$ forward autoregressively to obtain $\hat{z}_{t+H}$. The main planner evaluates the terminal prediction in the full latent space:

$$J(A) = \|\hat{z}_{t+H} - z_g\|_2^2 . \tag{8}$$

CEM iteratively updates a distribution over action sequences to minimize J. The agent executes the first action block from the selected sequence, receives a new observation, and replans in closed loop. Exact horizon, sampling, and replanning settings are reported in Appendix B.

4 Experiments

4.1 Experimental setup and metrics

Our experiments ask whether action supervision improves latent planning, how its benefit depends on visual diversity, and what action information the learned representations retain.

Protocol. Primary planning comparisons use three independently trained models per condition, each evaluated on 500 episodes in 10 runs of 50 episodes. Paired methods share the physical trajectory pool, evaluation tasks, and planner. The Push-T diversity comparison pairs training runs across $K{=}1$ and $K{=}24$; the other tasks compare three-model means at $K{=}1$ and their task-specific high-diversity settings. Evaluation starts and goals are sampled from the offline trajectory pool used for world-model learning, with current and goal observations sharing an appearance from the training palette. Separate evaluations change the training appearance assignment or use a disjoint test palette. Architecture, optimization, task, and CEM settings are detailed in Appendix B.

Table 1: **Planning gains across appearance conditions.** Gain is Concat IDM minus Plain JEPA, computed from the displayed success means in percentage points (pp). Bold marks the higher success mean, including ties, and positive gains.

	Single appearance (K=1)			Multiple appearances			
Task	Plain JEPA	Concat IDM	Gain	K	Plain JEPA	Concat IDM	Gain
Push-T	72.2±7.9	**87.5±4.4**	**+15.3**	24	9.1±2.7	**77.2±0.2**	**+68.1**
TwoRoom	88.5±1.3	**91.1±1.0**	**+2.6**	36	53.2±1.4	**97.3±3.2**	**+44.1**
Cube	60.9±1.2	**71.1±2.2**	**+10.2**	24	60.7±1.2	**75.4±1.3**	**+14.7**
Reacher	**100.0±0.0**	**100.0±0.0**	0.0	24	**100.0±0.0**	**100.0±0.0**	0.0

Metrics. Our primary metric is closed-loop success. Unless specified in the table, aggregate results are mean ± sample SD across three independently trained models per condition or appearance assignment. Action-decoding R^2 measures accessible action information; predicted-versus-realized latent-cost agreement measures rollout-ranking consistency. Normalized final error (NFE) and best progress provide complementary summaries of the same state-space distance; their definitions and per-model results are in Appendix B and Table 5.

4.2 Performance across control tasks

Planning gains across tasks. Concat IDM improves planning on Push-T, TwoRoom, and Cube without changing the test-time planner (Table 1). Its advantage is larger under high visual diversity on Push-T and TwoRoom: gains rise from 15.3 to 68.1 percentage points and from 2.6 to 44.1 points, respectively. On Cube, Plain JEPA remains near 61% while Concat IDM gains 10.2 points at K=1 and 14.7 at K=24. Reacher is saturated at 100% for both methods.

Inverse-dynamics designs. Compared with Δz-IDM, Concat IDM has higher mean success on Push-T (77.2±0.2% versus 65.8±14.7%) and the same mean on TwoRoom (97.3±3.2% versus 97.3±2.4%). Against PLDM, Concat IDM gains 24.0 percentage points and wins all three matched Push-T training pairs (Table 2a). This cross-design comparison retains PLDM's own optimization configuration; Appendix C.4 details the implementation.

Dependence on regularization. The benefit of inverse supervision also appears with VICReg (Bardes et al., 2022). In a control that replaces SIGReg with VICReg and keeps the backbone, data, training schedule, and planner fixed, adding Concat IDM increases success from 5.2% to 26.0% and reduces NFE from 1.039 to 0.890 (Table 2b).

Action supervision through the encoder. We isolate the contribution of the inverse loss by blocking its encoder gradients or disrupting action–transition correspondence (Figure 3). Detaching the IDM inputs reduces success from the matched Concat IDM control of 75.5±2.8% to 21.1±17.2%; shuffling action targets reduces it to 9.4±3.7%. Both interventions also worsen NFE (Table 7). These reductions support the role of correctly paired action supervision in shaping representations useful for planning.

4.3 Role of appearance diversity

A larger gain at higher visual diversity. On Push-T, we render the same physical trajectory pool under K=1 and K=24, matching training blocks across conditions. The high-minus-low difference in the Concat-minus-Plain gain is +52.9±7.1 percentage points, positive in all three blocks (Figure 1b; Table 4).

Robustness across appearance assignments. Concat IDM maintains high success when the mapping from physical trajectories to appearances changes. We train three Plain/Concat pairs under each of three independent episode-to-appearance assignments, keeping the trajectory pool, K=24 palette, and canonical evaluation tasks and renderings fixed. All nine Concat IDM models achieve 74.2–78.4% success, close to the canonical 77.2±0.2%, while Plain JEPA spans 5.6–61.2%. Concat

Table 2: Planning success (%) and normalized final error (NFE) for inverse-dynamics designs and regularizers on Push-T (K=24).

(a) IDM-based alternatives

Method	Success (%) ↑	NFE ↓
Plain JEPA	9.1±2.7	1.353±0.108
PLDM	53.2±11.8	0.783±0.103
Δz-IDM	65.8±14.7	0.777±0.177
Concat IDM	**77.2±0.2**	0.652±0.032

(b) Regularizer control

Reg.	No IDM		Concat IDM		Gain
	Success ↑	NFE ↓	Success ↑	NFE ↓	(pp)
SIGReg	9.1±2.7	1.353±0.108	77.2±0.2	0.652±0.032	+68.1
VICReg	5.2±3.0	1.039±0.059	26.0±7.1	0.890±0.050	+20.8

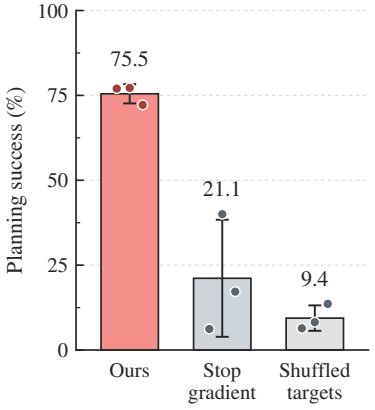
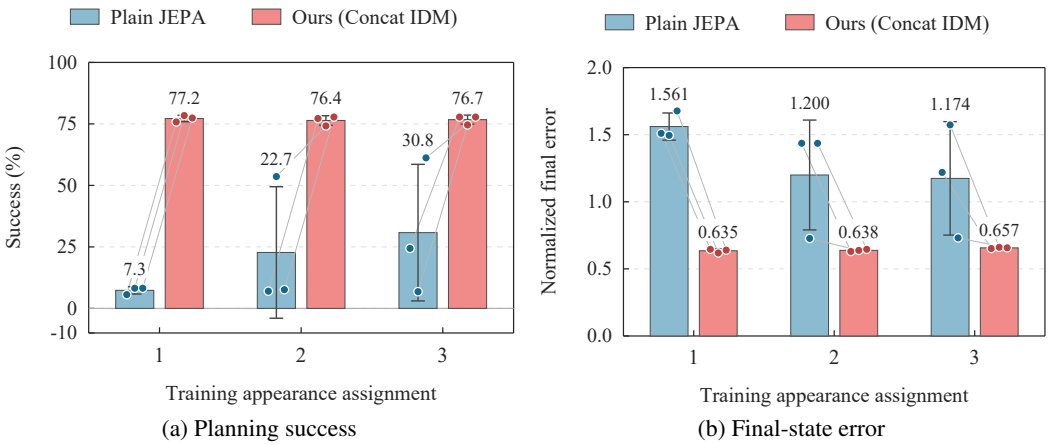


Figure 3: **Action supervision through the encoder.** Blocking encoder gradients or shuffling action labels reduces Push-T success relative to a matched Concat IDM control (K=24).

IDM improves success and reduces NFE in all nine matched pairs (Figure 4a,b). Appendix C.2.1 reports the per-assignment results.

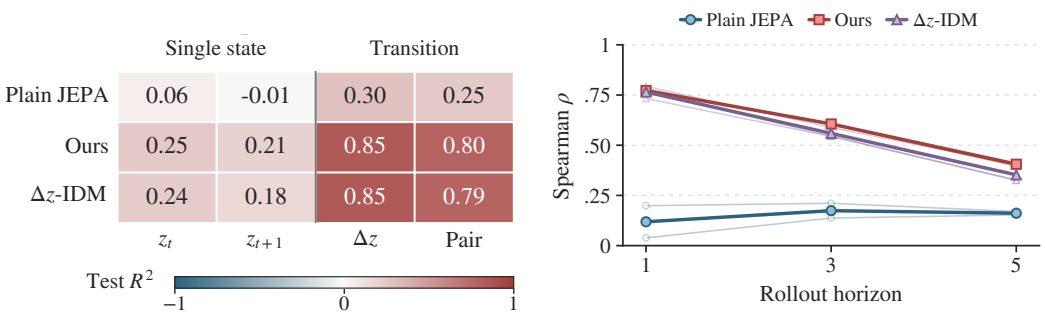
4.4 Action information and planning costs

Action information in latent transitions. We train fresh MLP probes on frozen encoders to predict actions from z_t, z_{t+1}, Δz, or $[z_t; z_{t+1}]$, using a common trajectory-level training, validation, and test split within the world-model data pool (Appendix C.3.1). For concatenated endpoints, test R^2 is 0.799±0.001 with Concat IDM versus 0.251±0.209 with Plain JEPA (Figure 5a). Concat IDM supports displacement decoding at 0.846±0.002, compared with 0.254±0.003 from its better individual endpoint. The Δz-IDM encoder attains a similar displacement score (0.850±0.002), despite lower mean Push-T planning success (Table 2a). The probes thus establish improved action accessibility in both representations, motivating a separate assessment of candidate-cost ordering.



(a) Planning success (b) Final-state error

Figure 4: **Robustness to training appearance assignments on Push-T (K=24).** (a) Planning success. (b) Normalized final error. Bars show means with sample SDs; dots denote trained models and lines connect matched runs.



(a) Frozen-encoder action probes

(b) Latent-cost agreement

Figure 5: **Action information and latent rollout consistency on Push-T.** (a) Frozen-encoder action-probe test R^2; *Pair* denotes $[z_t; z_{t+1}]$. (b) Spearman correlation between predicted and realized latent costs. Thin lines show individual models; thick lines show means.

Ordering candidate plans. We compare each predicted terminal cost with the realized latent cost obtained by executing the same candidate and encoding its outcome with the same model (Appendix C.3.2). This diagnostic measures whether imagined and realized outcomes induce consistent candidate rankings in the learned latent space. At horizon H=5, Concat IDM achieves Spearman correlations of 0.395–0.411, compared with 0.153–0.170 for Plain JEPA (Figure 5b). Together, the probes and ranking comparisons show more accessible action information and more consistent latent rollouts.

4.5 Sensitivity and Scope

IDM weight and training progress. At the default λ=1, Concat IDM improves success from 9.1% to 77.2% and reduces NFE from 1.353 to 0.652 (Table 3a). The additional weight settings yield success rates of 71.0% at λ=0.1 and 85.8% at λ=10. Across the three training runs, the advantage emerges early: at epoch 3, Concat IDM reaches 63.1% success versus 10.6% for Plain JEPA, with lower NFE at every reported epoch (Table 3b).

Table 3: IDM weight and training progress on Push-T (K=24). In (a), n=3 for $\lambda \in \{0, 1\}$ and n=1 otherwise. In (b), each epoch uses the same three training runs per method. Success is in percent; NFE is lower-is-better.

(a) IDM loss weight

λ	Success ↑	NFE ↓
0	9.1±2.7	1.353±0.108
0.1	71.0	0.713
1	77.2±0.2	0.652±0.032
10	85.8	0.610

(b) Training progress

	Plain JEPA		Concat IDM	
Epoch	Success ↑	NFE ↓	Success ↑	NFE ↓
1	4.3±1.4	1.616±0.041	10.6±1.6	0.914±0.090
3	10.6±0.9	1.287±0.088	63.1±4.0	0.694±0.028
5	24.4±11.4	1.039±0.225	70.8±1.6	0.678±0.015
10	9.1±2.7	1.353±0.108	77.2±0.2	0.652±0.032

Scope and limitations. The principal finding concerns planning with seen appearances and matching current and goal renderings. Appearance remains decodable, and success on a disjoint palette is 4.2% for Plain JEPA and 13.4% for Concat IDM, leaving substantial room for transfer to unseen appearances (Appendix D). Directly measuring action-outcome merging in trained models would further connect the proposition to planning behavior. Appendix C.5 presents a selected Push-T rollout comparing Concat IDM with Plain JEPA and Δz-IDM.

5 Conclusion

Action prediction penalizes the merging of distinguishable action outcomes that forward latent prediction can leave unpenalized. Under controlled appearance variation over fixed physical trajectories,

Concat IDM improves planning, with larger gains at high visual diversity on Push-T and TwoRoom and consistent benefits across independent appearance assignments. Encoder interventions, action probes, and candidate-cost comparisons link these gains to correctly paired action supervision, accessible transition information, and improved latent rollout consistency. These findings motivate supervising action–consequence distinctions alongside forward prediction when learning planning representations from visually diverse observations.

AI USE STATEMENT

In this work, we used generative AI tools for drafting and editing prose, assisting with the development and checking of the theoretical analysis, and supporting code development and experiment management. We have reviewed all AI-assisted text and code; all scientific claims, experimental protocols, numbers, and analyses were verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI. See the ICLR 2027 AI Policy for Authors for more details.

REPRODUCIBILITY STATEMENT

Code, configuration files, and evaluation scripts will be included in the anonymized supplementary material and released publicly with the camera-ready version. Training and evaluation hyperparameters, CEM planner settings, run identifiers, and dataset construction details (nested palettes, fixed trajectory pools) are documented in Appendix B. We report closed-loop success together with normalized final error as defined in Eq. (16), and apply the numerical validity checks in Appendix B.1 before aggregation. Every task, method, and appearance-set size in Table 1 uses three independently trained models, each evaluated on 500 episodes. The additional Push-T assignment study uses three training repetitions per method under each of three episode assignments on the same 500 tasks as the canonical models. Sensitivity analyses, diversity sweeps, and additional controls use the same training and evaluation protocol.

REFERENCES

Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Guo An, Zijing Wu, Honghua Dong, Yuhao Yan, Zixuan Gui, Haochong Chen, Shanzhao Ruan, Xiang Wang, Yurong Ling, and Qi Tian. Diagnosing JEPA world models with action-conditioned predictive consistency, 2026.

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

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Jiaxin Bai and Jiaxuan Xiong. Temporal-distance JEPA: Plan-aware representation learning for latent world model predictive control, 2026.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

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. V-JEPA.

Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provable RL with exogenous distractors via multistep inverse dynamics. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RQLLzMCefQu.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

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, 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. URL https://openreview.net/forum?id=S1lOTC4tDS.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. 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. 8387–8406, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009. URL https://hastie.su.domains/ElemStatLearn/.

Riashat Islam, Manan Tomar, Alex Lamb, Yonathan Efroni, Hongyu Zang, Aniket Rajiv Didolkar, Dipendra Misra, Xin Li, Harm van Seijen, Remi Tachet des Combes, and John Langford. Principled offline RL in the presence of rich exogenous information. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14390–14421, 2023. URL https://proceedings.mlr.press/v202/islam23a.html.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020. URL https://arxiv.org/abs/2004.13649.

Alex Lamb, Riashat Islam, Yonathan Efroni, Aniket Rajiv Didolkar, Dipendra Misra, Dylan J. Foster, Lekan P. Molu, Rajan Chari, Akshay Krishnamurthy, and John Langford. Guaranteed discovery of control-endogenous latent states with multi-step inverse models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=99J6lWgXCl.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5639–5650, 2020a. URL https://proceedings.mlr.press/v119/laskin20a.html.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems*, volume 33, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/e615c82aba461681ade82da2da38004a-Abstract.html.

Alexander Levine, Peter Stone, and Amy Zhang. Multistep inverse is not all you need. *Reinforcement Learning Journal*, 2:884–925, 2024. URL https://rlj.cs.umass.edu/2024/papers/Paper117.html.

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

Keiran Paster, Sheila A. McIlraith, and Jimmy Ba. Planning from pixels using inverse dynamics models. In *International Conference on Learning Representations (ICLR)*, 2021.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Gawon Seo, Dongwon Kim, and Suha Kwak. ACID: Action consistency via inverse dynamics for planning with world models, 2026.

Vlad 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. *arXiv preprint arXiv:2502.14819*, 2025. PLDM.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

Leonardo F. Toso, Davit Shadunts, Yunyang Lu, Nihal Sharma, Donglin Zhan, Nam H. Nguyen, and James Anderson. Learning invariant visual representations for planning with joint-embedding predictive world models, 2026.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhenghao Zhang, Yuanxiang Wang, Zhenyu Guan, Yujia Yang, Bingkang Shi, Tianyu Zong, Hongzhu Yi, Guoqing Chao, Xingchen Chen, Tiankun Yang, Chenxi Bao, Tao Yu, Jingjing Zhou, and Jungang Xu. Delta-JEPA: Learning action-sensitive world models via latent difference decoding, 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 International Conference on Machine Learning (ICML)*, 2025.

A WHAT ACTION PREDICTION CONSTRAINS

We analyze the squared inverse loss in Eq. (6) to specify how it penalizes the loss of action-related distinctions. The analysis concerns a fixed encoder and the information available to an inverse decoder; it does not assume that training finds a global optimum. Actions are expressed in the same coordinates as the training targets, and d_a denotes their dimension.

A.1 INVERSE PREDICTION ERROR AND INFORMATION LOST THROUGH ENCODING

Let $X = (O, O^+)$ be a random pair of consecutive observations, let $A \in \mathbb{R}^{d_a}$ satisfy $\mathbb{E}\|A\|_2^2 < \infty$, and let $U = [E_\theta(O); E_\theta(O^+)]$ for a fixed measurable encoder. For a measurable decoder g with finite squared risk, define

$$\mathcal{R}(g, E_\theta) = \frac{1}{d_a}\mathbb{E}\|A - g(U)\|_2^2. \tag{9}$$

Write $\mu_X = \mathbb{E}[A \mid X]$ and $\mu_U = \mathbb{E}[A \mid U]$. The conditional mean minimizes squared prediction error, a standard result in regression (Hastie et al., 2009). Applying this identity before and after encoding gives the following decomposition.

Lemma A.1 (Excess inverse prediction error)**.** *Let $\mathcal{R}_X^* = \inf_h d_a^{-1}\mathbb{E}\|A - h(X)\|_2^2$ and $\mathcal{R}_E^* = \inf_g \mathcal{R}(g, E_\theta)$, where each infimum is over all measurable decoders of finite risk. Then*

$$\mathcal{R}(g, E_\theta) = \mathcal{R}_X^* + \frac{1}{d_a}\mathbb{E}\|\mu_X - \mu_U\|_2^2 + \frac{1}{d_a}\mathbb{E}\|\mu_U - g(U)\|_2^2, \tag{10}$$

$$\mathcal{R}_E^* - \mathcal{R}_X^* = \frac{1}{d_a}\mathbb{E}\|\mu_X - \mu_U\|_2^2. \tag{11}$$

Proof. Since U is a function of X, the tower property gives $\mathbb{E}[\mu_X \mid U] = \mu_U$. Decompose

$$A - g(U) = (A - \mu_X) + (\mu_X - \mu_U) + (\mu_U - g(U)). \tag{12}$$

The first term has conditional mean zero given X, so it is orthogonal in expectation to the other two terms, both of which are functions of X. The second term has conditional mean zero given U, so its cross term with the third also vanishes. Expanding the squared norm therefore yields Eq. (10), with $\mathcal{R}_X^* = d_a^{-1}\mathbb{E}\|A - \mu_X\|_2^2$. Taking $g(U) = \mu_U$ attains the infimum and gives Eq. (11). □

The three terms in Eq. (10) separate ambiguity already present in the observations, additional error caused by encoding, and error due to the chosen inverse decoder. Thus, the representation penalty concerns action-predictive information that the observations actually contain. Equality $\mathcal{R}_E^* = \mathcal{R}_X^*$ requires preservation of the conditional action mean, not the entire physical state or even the full conditional action distribution.

A.2 PROOF AND EXTENSIONS FOR MERGED TRANSITIONS

We prove Proposition 3.1 from Section 3.2, using the transition pair and risk defined in Eq. (2).

Proof. The shared current observation and merged successor encodings imply $u_1 = u_2 = u$. For $b = g(u)$, completing the square gives

$$p\|b - a_1\|_2^2 + (1-p)\|b - a_2\|_2^2 = \|b - \bar{a}\|_2^2 + p(1-p)\|a_1 - a_2\|_2^2. \tag{13}$$

Divide by d_a and use nonnegativity of the first term. Taking $b = \bar{a}$ proves attainability. In contrast, both forward targets equal $z^+ = E_\theta(o_1^+) = E_\theta(o_2^+)$, so a predictor outputting z^+ for either action has zero error on both targets. □

For equal mixture weights, the lower bound is $\|a_1 - a_2\|_2^2/(4d_a)$. The forward prediction statement concerns this pair and the prediction term; it is not a claim about the minimum of the full SIGReg-regularized training objective. If a larger discrete distribution assigns masses $w_1, w_2 > 0$ to this pair, nonnegativity of the remaining loss terms gives the unconditional lower bound $w_1 w_2\|a_1 - a_2\|_2^2/[d_a(w_1 + w_2)]$. No positive mass is assumed for individual points in a continuous distribution; Lemma A.1 applies directly in that setting.

Corollary A.2 (Nearby encodings)**.** *For the pair in Eq. (2), suppose the decoder satisfies* $\|g(u_1) - g(u_2)\|_2 \le L\|u_1 - u_2\|_2$ *for some* $L \ge 0$*. Then, writing* $[x]_+ = \max(x, 0)$*,*

$$\mathcal{R}_{\text{pair}} \ge \frac{p(1-p)}{d_a} \left[\|a_1 - a_2\|_2 - L\|u_1 - u_2\|_2\right]_+^2 . \tag{14}$$

Proof. Set $e_i = \|g(u_i) - a_i\|_2$. The triangle inequality and the decoder assumption imply $e_1 + e_2 \ge [\|a_1 - a_2\|_2 - L\|u_1 - u_2\|_2]_+$. Moreover,

$$pe_1^2 + (1-p)e_2^2 - p(1-p)(e_1 + e_2)^2 = \big(pe_1 - (1-p)e_2\big)^2 \ge 0. \tag{15}$$

Combining the two inequalities and dividing by d_a proves the claim. □

At a common current state, the distance between the concatenated inputs equals the distance between successor encodings. The same is true for displacement inputs, so these bounds do not predict an advantage for one decoder parameterization over the other. The nearby-encoding result is conditional on the stated Lipschitz bound; our training does not constrain or estimate L.

A.3 External factors and scope of the analysis

If appearance C is independent of A and a latent transition retains only appearance, $U = f(C)$, then $\mathbb{E}[A \mid U] = \mathbb{E}[A]$. Its best inverse predictor is consequently the constant action mean, with risk $d_a^{-1}\mathbb{E}\|A - \mathbb{E}[A]\|_2^2$. Under this independence assumption, appearance alone cannot reduce the inverse loss below that constant-prediction baseline. Independence is a distributional assumption: it does not follow solely from F being unaffected by c, and random assignment does not eliminate all correlations in a realized finite dataset.

The pair construction specifies a penalty for represented distinctions, not a requirement to supply counterfactual pairs during training. Offline coverage determines which distinctions the empirical objective can constrain. When different actions have indistinguishable observed consequences, inverse ambiguity can persist even without information loss through encoding. Conversely, a conditional action mean can be preserved using policy correlations or without retaining other state variables needed for planning. Related work on multistep inverse models also distinguishes single-step action decoding from sufficient representations for control (Islam et al., 2023). Our bounds therefore explain a constraint supplied by the inverse objective; they do not establish identification of F, appearance invariance, or a planning guarantee. The dependence of the planning gain on visual diversity is an empirical finding evaluated in Section 4.

B Implementation details

Architecture. We use a ViT-Tiny visual encoder (patch size 14, image size 224, embedding dimension 192) with a JEPA predictor of depth 6, 16 heads, MLP width 2048, and embedding dropout 0. Observations are stacked with history size 3. The inverse dynamics head is an MLP with hidden width 512. The SIGReg term $\mathcal{L}_{\text{SIG}}$ uses weight 0.09, 17 knots, and 1024 projections.

Action preprocessing. Actions are standardized coordinatewise as $\tilde{a}_j = (a_j^{\text{raw}} - \mu_j)/\sigma_j$, using the sample mean and standard deviation of each action coordinate in the offline dataset, computed before its training/validation split. The same standardized actions are used as predictor inputs and IDM targets. The primitive actions between consecutive sampled observations are concatenated into an action-block vector, whose dimension is denoted by d_a. In the training equations, a_t denotes this standardized vector.

Optimization. Unless noted: AdamW with learning rate 2×10^{-5}, weight decay 10^{-3}, batch size 128, bf16 mixed precision, gradient clipping at 1.0, and 10 training epochs. IDM loss weight λ=1.0 for Concat IDM and Δz-IDM. The primary method uses no routing or latent partition.

Planning. Closed-loop evaluation uses CEM with 300 samples, 30 optimization steps, top-30 elites, planning horizon 5, receding horizon 5, and action block 5 (frame skip). Each trained model is evaluated on 10 runs of 50 episodes using the fixed evaluation tasks. The CEM cost is the full latent prediction cost (`plan_cost=full`).

Task metrics. Success follows each environment's predefined task criterion. For Push-T, success requires the combined agent and block position error to be below 20 pixels and the symmetry-aware block orientation error to be below $\pi/9$. Auxiliary normalized final error is

$$\mathrm{NFE} = \frac{d(s_T, g)}{d(s_0, g) + \epsilon}, \tag{16}$$

where $\epsilon = 10^{-8}$ and s_T is the recorded state at first success or the end of the evaluation budget. Best progress is $1 - \min_{0 \le t \le T} d(s_t, g)/(d(s_0, g) + \epsilon)$. Push-T NFE uses the full state-vector distance $d(s, g) = \|s - g\|_2$, with $s = (x_a, y_a, x_b, y_b, \theta_b, v_{a,x}, v_{a,y})$: agent position, block position, block orientation, and agent velocity. Positions are in pixels, orientation is in radians, and velocity components retain their simulator units. This distance uses the raw state components without componentwise standardization or an angular wrap correction; the symmetry-aware orientation criterion applies to success, not to NFE. NFE and best progress therefore depend on component scales and angular coordinates; we use them as within-task auxiliary diagnostics. TwoRoom uses $d(s, g) = \|p_{\mathrm{agent}} - p_{\mathrm{target}}\|_2$. Cube uses a single-cube goal pose and Reacher uses a target joint configuration; their continuous-error results are not reported in Table 1. All methods use the same ϵ and normalization criteria: finite recorded initial and final errors and initial distance at least 10^{-6}. NFE and best progress average this eligible episode set; success includes all episodes. NFE is unclipped, so values above one indicate a larger final than initial error. Best progress measures the closest recorded approach, including the initial state: values near zero indicate little improvement and values near one indicate proximity to the goal at some point during execution. TwoRoom Euclidean distance provides a continuous complement to success but omits wall-induced path constraints. NFE magnitudes are compared within each environment.

Data. The datasets keep official state and action trajectories fixed and render them under K appearances. The Push-T pool contains 18,685 episodes and 2,336,736 recorded time steps. Training batches do not contain matched appearance views. TwoRoom uses $K \in \{1, 36\}$; Cube and Reacher use $K \in \{1, 24\}$; Push-T uses $K \in \{1, 24\}$. Push-T palettes are nested prefixes of a fixed color list, so increasing K adds appearances without changing the lower-K palette entries. Increasing K redistributes the same physical trajectory pool across more visual realizations rather than adding new control trajectories. For Push-T at $K{=}24$, the canonical appearance index is $(e \bmod 24) + 1$, where e is the zero-based episode index. The additional assignment study replaces this mapping by an episode-level assignment sampled independently for each dataset, without changing the palette or physical state and action sequences. Unlike a global permutation of palette entries over fixed residue classes, these assignments can change which episodes share an appearance.

Visual conditions. The configuration names `hard` and `hard2` refer to appearance palettes rather than additional physical obstacles or longer planning horizons. TwoRoom combines three background colors, three agent colors, two wall colors, and two wall-border colors to form 36 high-contrast appearances (`hard`). Cube combines three cube colors, four two-color floor palettes, and two robot colors to form 24 appearances, with light intensity fixed at 0.4 (`hard2`). Its cube colors are close to floor colors in some combinations, reducing visual contrast. Reacher combines three arm colors and eight two-color floor palettes, also at light intensity 0.4, to form 24 appearances (`hard`). Physical trajectories remain unchanged when these appearance conditions are applied.

Evaluation tasks. The main evaluation samples initial states from the same offline trajectory pool used for model learning and takes the goal state and image 25 environment steps later in the same episode. The simulator is initialized at the sampled state and rendered with that episode's appearance; subsequent observations come from the closed-loop rollout, while the goal image remains fixed. Each task has a budget of 50 environment steps. The corresponding appearance is shared by the current and goal observations throughout each rollout. For the Push-T assignment study, every model uses the canonical evaluation dataset (`multi_pusht_k24_v1_repack.lance`), regardless of its training assignment. The evaluation manifest uses identical episode indices for corresponding evaluation runs and covers all 24 appearances. Both the goal image and simulator rendering use the canonical appearance index $(e \bmod 24) + 1$.

Intervention ablations. The stop-gradient intervention detaches consecutive latents before the IDM head, blocking gradients from the IDM objective to the encoder. The shuffled-target intervention

Table 4: Paired planning gains and the Push-T diversity interaction. Entries are mean ± sample standard deviation across three matched training blocks within each comparison. Success gain is $\Delta S = S_{\text{method}} - S_{\text{Plain}}$, measured in percentage points (pp). NFE improvement is $\Delta E = \text{NFE}_{\text{Plain}} - \text{NFE}_{\text{Concat}}$; positive values favor Concat IDM. The interaction is the high-diversity gain minus the low-diversity gain.

(a) Success gains at high visual diversity

Task	K	Method	Gain (pp) ↑
Push-T	24	Δz-IDM	$+56.7 \pm 13.7$
		Concat IDM	$+68.1 \pm 2.5$
TwoRoom	36	Δz-IDM	$+44.1 \pm 1.3$
		Concat IDM	$+44.1 \pm 3.9$

(b) Push-T appearance-diversity interaction

Concat IDM improvement	$K{=}1$	$K{=}24$	High minus low
Success gain (pp)	$+15.3 \pm 7.3$	$+68.1 \pm 2.5$	$+52.9 \pm 7.1$
NFE improvement	$+0.107 \pm 0.116$	$+0.701 \pm 0.115$	$+0.594 \pm 0.138$

applies a batch permutation to the action labels, breaking their correspondence with the transitions while preserving the IDM architecture.

B.1 Numerical validity checks

Before aggregation, every checkpoint is checked for finite model parameters, encoded features, predictions, planner costs, and episode metrics. Only checkpoints satisfying these criteria are included in the reported aggregates.

C Supporting experiments and diagnostics

C.1 Paired effects and individual training runs

Table 5 consolidates the per-model results for the primary tasks and the matched Push-T diversity comparison. Runs A, B, and C identify the corresponding training repetitions.

C.2 Appearance diversity and training assignments

Repeated single-appearance controls. The TwoRoom hard, Cube hard2, and Reacher hard controls at $K{=}1$ use training seeds $\{42, 100, 2026\}$ for both Plain JEPA and Concat IDM, giving 18 independently trained models across the three tasks. Each model is evaluated in 10 runs of 50 episodes, and Table 1 reports mean ± sample SD across the three trained models per condition. The cross-task comparison summarizes low- and high-diversity model means; the paired gain-difference analysis in Table 4(b) uses the matched Push-T training blocks.

C.2.1 Independent training appearance assignments

We train three paired Plain JEPA and Concat IDM models under each independent episode-to-appearance assignment, using generator identifiers 11, 13, and 17 and training initializations 42, 100, and 7. All assignment-specific epoch-10 checkpoints share the canonical evaluation tasks and mapping $(e \bmod 24) + 1$ for both goal images and simulator renderings. Table 6 combines success and distance diagnostics for all models underlying Figure 4; every paired success gain is positive ($+16.6$–$+70.2$ percentage points). The mapping and checkpoint checks are documented in the supplementary evaluation records.

Table 5: Per-model planning results for the primary tasks and matched Push-T diversity comparison. Success includes all 500 episodes; distance diagnostics include all normalization-valid episodes, including failures.

Method	Run	Success (%) ↑	NFE ↓	Best progress ↑
Push-T (K=1)				
Plain JEPA	A	81.2	0.623	0.673
	B	66.2	0.832	0.639
	C	69.2	0.821	0.648
Concat IDM	A	90.4	0.650	0.650
	B	89.6	0.643	0.658
	C	82.4	0.665	0.661
Push-T (K=24)				
Plain JEPA	A	7.4	1.334	0.433
	B	7.6	1.256	0.484
	C	12.2	1.469	0.526
Δz-IDM	A	48.8	0.980	0.626
	B	75.2	0.660	0.675
	C	73.4	0.691	0.676
Concat IDM	A	77.0	0.689	0.667
	B	77.2	0.631	0.674
	C	77.4	0.636	0.681
TwoRoom (K=36)				
Plain JEPA	A	54.0	0.798	0.394
	B	54.0	0.783	0.388
	C	51.6	0.812	0.378
Δz-IDM	A	97.4	0.458	0.568
	B	99.6	0.437	0.575
	C	94.8	0.496	0.557
Concat IDM	A	93.6	0.479	0.557
	B	99.6	0.442	0.577
	C	98.6	0.451	0.569

Table 6: Per-model Push-T results under independent training appearance assignments (K=24). Each row summarizes the common 500-task evaluation, including failures. Runs A–C identify matched training repetitions within each assignment.

Assignment	Method	Run	Success (%) ↑	NFE ↓	Progress ↑
1	Plain JEPA	A	5.6	1.510	0.397
		B	8.2	1.495	0.469
		C	8.2	1.677	0.391
	Concat IDM	A	75.8	0.646	0.673
		B	78.4	0.619	0.681
		C	77.4	0.640	0.664
2	Plain JEPA	A	7.0	1.436	0.407
		B	53.6	0.727	0.670
		C	7.6	1.436	0.439
	Concat IDM	A	77.2	0.630	0.674
		B	74.2	0.638	0.682
		C	77.8	0.646	0.672
3	Plain JEPA	A	24.4	1.219	0.586
		B	6.8	1.573	0.425
		C	61.2	0.731	0.672
	Concat IDM	A	77.8	0.652	0.672
		B	74.6	0.661	0.675
		C	77.8	0.657	0.669

Table 7: Push-T interventions and their matched Concat IDM control. Success change is intervention minus control, measured in percentage points (pp).

Condition	Success (%) ↑	NFE ↓	Change vs. Concat IDM (pp)
Concat IDM	75.5±2.8	0.664±0.030	—
stop-gradient	21.1±17.2	1.178±0.490	−54.3±16.8
shuffled targets	9.4±3.7	1.430±0.278	−66.1±6.5

C.3 Action supervision and planning diagnostics

C.3.1 Standardized action probe

We freeze each encoder and train separate MLP probes to predict a_t from z_t, z_{t+1}, $z_{t+1} - z_t$, or $[z_t; z_{t+1}]$. All methods use the same fixed trajectory-level training, validation, and test split (manifest hash `e220c0d92f06e26d`). The probe samples come from the offline dataset used for world-model learning; this partition defines held-out data for the probes, not an independent world-model test set. Each action dimension is standardized using training-split statistics only, probe checkpoints are selected on the validation split, and all reported values are computed on the held-out test split. The architecture, training budget, and multivariate R^2 aggregation are identical across inputs and encoders; no probe reuses a training-time IDM head.

Table 8: Held-out action-prediction R^2 on Push-T. Entries are mean ± sample standard deviation across independently trained encoders. Columns vary only the probe input.

Method	z_t	z_{t+1}	Δz	$[z_t; z_{t+1}]$
Plain JEPA	0.056±0.063	−0.010±0.004	0.296±0.198	0.251±0.209
Δz-IDM	0.235±0.015	0.178±0.014	0.850±0.002	0.790±0.003
Concat IDM	0.254±0.003	0.210±0.008	0.846±0.002	0.799±0.001

Per-run R^2, normalized MSE, and training, validation, and test sample counts are provided in the supplementary CSV.

C.3.2 Candidate-cost rankings

For each of 128 initial-state and goal pairs, we evaluate a common set of 96 candidate action sequences at horizons $H \in \{1, 3, 5\}$. We compare the model-predicted latent cost J_{pred} with the latent cost J_{real} obtained by replaying the same sequence in the environment and re-encoding the realized observations. We report within-task Spearman rank agreement, selection regret normalized by the 90th–10th percentile range of realized costs, and overlap between the predicted and realized top-10% candidates.

Table 9: Cost-ranking diagnosis across rollout horizons. Entries are ranges across training runs, using 128 initial-goal pairs and 96 shared candidates per pair. Lower selection regret and higher rank agreement or overlap are better.

Method	H	Spearman ↑	Selection regret ↓	Top-10% overlap ↑
Plain JEPA	1	0.039–0.199	1.849–3.618	0.188–0.278
Concat IDM	1	0.770–0.778	0.386–0.418	0.619–0.630
Δz-IDM	1	0.733–0.793	0.490–0.561	0.596–0.630
Plain JEPA	3	0.137–0.211	2.038–3.682	0.182–0.190
Concat IDM	3	0.604–0.610	0.697–0.794	0.378–0.387
Δz-IDM	3	0.544–0.588	0.797–0.951	0.350–0.377
Plain JEPA	5	0.153–0.170	1.664–3.107	0.147–0.170
Concat IDM	5	0.395–0.411	0.961–1.067	0.263–0.277
Δz-IDM	5	0.324–0.405	0.910–1.004	0.247–0.281

Rank agreement decreases at $H{=}3$ and $H{=}5$ but remains higher for both IDM parameterizations than for Plain JEPA. Rank agreement measures ordering consistency and is invariant to monotone rescaling of the costs.

C.4 External PLDM implementation

Matched repeated-training comparison. We train three PLDM models on the same Push-T $K{=}24$ dataset and evaluate their epoch-10 checkpoints on the same 500 tasks as the primary Concat IDM models. The planner uses the same horizon of 5, 30 CEM iterations, 300 candidates, 30 elites, and execution budget of 50 steps. Training seeds 42, 100, and 7 attain success rates of 66.8%, 45.2%, and 47.6%, respectively, giving the $53.2{\pm}11.8\%$ aggregate in Table 2a.

Implementation. PLDM learns a ViT-Tiny encoder from scratch with 192-dimensional embeddings, a three-step history, and its own predictor, projection heads, variance and covariance regularization, temporal alignment, and inverse dynamics objective. Its inverse head takes concatenated 192-dimensional embeddings through a 512-unit hidden layer to predict the 10-dimensional action block; the IDM loss weight is 1. Training uses AdamW with learning rate $5{\times}10^{-5}$, weight decay 10^{-3}, batch size 128, bf16 precision, and gradient clipping at 1.0 for 10 epochs. PLDM thus retains its own training configuration, including a higher learning rate than the LeWorldModel-based models ($2{\times}10^{-5}$).

C.5 Closed-loop behavior across task geometries

Six tasks were selected by initial-to-goal changes in block position and orientation, independently of method outcomes: two emphasize position, two orientation, and two both. Using Run A from each method at $K{=}24$, Concat IDM succeeds on five of these six tasks; Plain JEPA and Δz-IDM succeed on none. These selected trajectories illustrate behavior rather than estimate aggregate success. Within each task, methods share the initial simulator state, history, goal image, and appearance, with a goal offset of 25 steps and a 50-step execution budget. Figure 6 shows one successful Concat IDM trajectory alongside the two baselines on the same task.

The physical readouts use joint pusher/block position error $\|[p_t; b_t] - [p_g; b_g]\|_2$ and wrapped orientation error $|\operatorname{atan2}(\sin(\theta_t - \theta_g), \cos(\theta_t - \theta_g))|$. Success requires both errors to fall below 20 pixels and $\pi/9$ radians at the same step. Images are recorded simulator frames; curves use saved states through termination without smoothing or extension after early success.

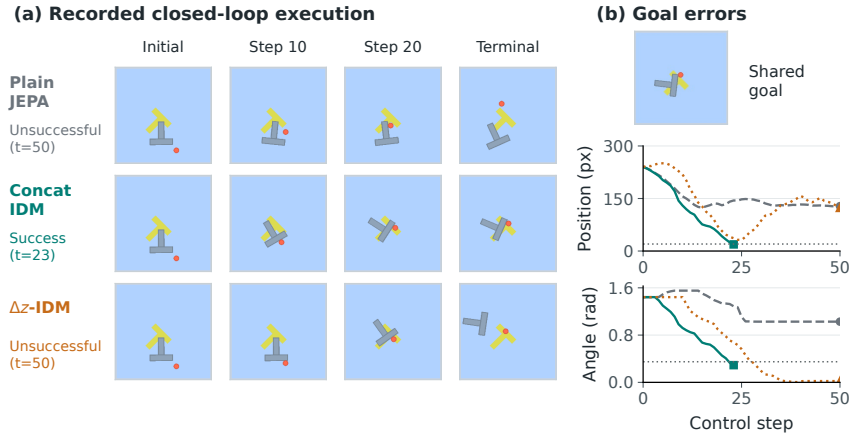


Figure 6: **Selected Push-T behavior, episode 16532.** The task requires translation and rotation. Concat IDM succeeds at step 23, while Plain JEPA and Δz-IDM remain outside the joint success tolerances. Dotted lines mark the physical success tolerances; traces stop at termination.

C.6 SELECTED TWOROOM ROLLOUT

Figure 7 presents a selected contrast in recorded outcomes, rather than an estimate of aggregate performance.

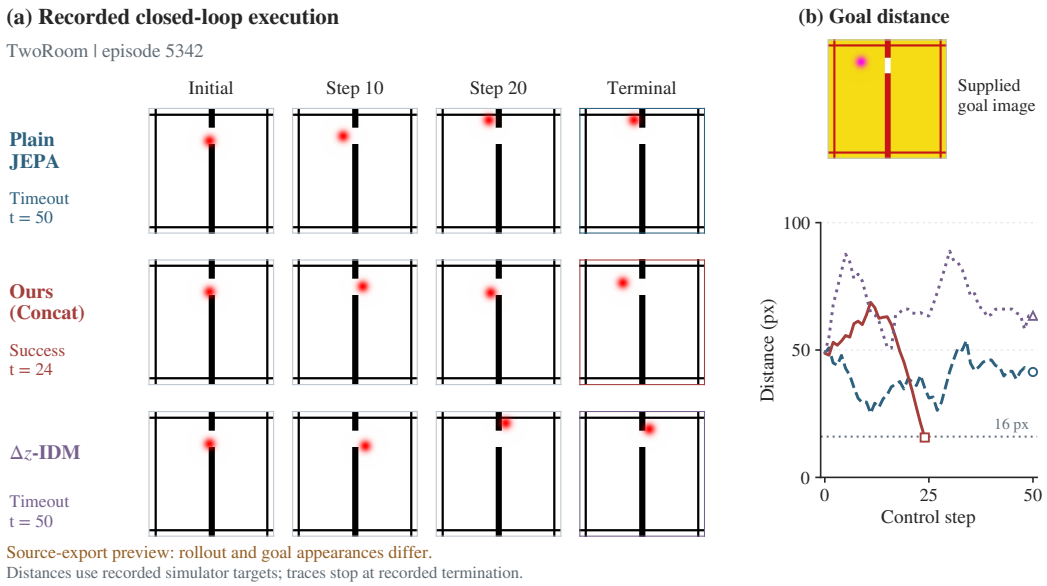


Figure 7: **Selected TwoRoom behavior, episode 5342.** Concat IDM reaches the 16-pixel goal tolerance at step 24, while Plain JEPA and Δz-IDM do not within 50 steps. Rows share the recorded initial state and goal image; curves stop at termination. *The exported rollout and goal have different appearances, so this is a diagnostic illustration rather than a verified matched-appearance evaluation.*

D ADDITIONAL SCOPE ANALYSES

Appearance probes. With the encoder frozen, linear probes regress factor RGB colors from z. On Push-T $K{=}24$, train-split background+goal R^2 remains high for Plain JEPA (0.978), Concat IDM (0.984), and Δz-IDM (0.989), showing that appearance information remains linearly decodable from the training data.

Transfer to a disjoint palette. Models trained at $K{=}24$ and evaluated on a disjoint palette obtain $4.2{\pm}1.9\%$ (Plain JEPA) and $13.4{\pm}4.2\%$ (Concat IDM). Concat IDM improves success on the disjoint palette, although absolute performance remains low for both methods.