REPRESENTATION SELECTION IN END-TO-END FEATURE-PREDICTIVE WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models support planning by evaluating candidate action sequences in compact visual representations. While fixed-target models predict future features in a fixed representation space, joint-embedding architectures avoid explicit reconstruction through end-to-end latent prediction; By dynamically updating the target encoder, they inherently turn the prediction objective into an implicit feature selector for predictable information. Under finite representation capacity and structural assumptions, this preference inherently marginalizes less predictable, task-relevant factors. We formalize this phenomenon as *predictability bias*. We then analytically derive how explicit target supervision redirects this capacity allocation, giving a sufficient condition under which retaining target variables improves alignment between the native planning score and task cost, even when pure prediction error rises. Controlled exact-whitening and action-conditioned experiments empirically verify this predictability-driven factor selection. In a separate four-task continuous control evaluation, countering this bias via explicit target supervision improves native closed-loop success under identical deployment conditions, suggesting that beyond merely fitting temporal dynamics, the joint prediction objective can act as an implicit filter that shapes the composition of the latent space.

1 INTRODUCTION

Feature-predictive world models combine two established ideas: learning visual representations by predicting future latent features (van den Oord et al., 2018) and using action-conditioned latent dynamics for planning (Hafner et al., 2019). When the prediction loss also updates the encoder that supplies the future target, reducing that loss can change both predictor fit and represented content. With limited capacity, this coupling may favor predictable factors over less predictable task-relevant factors, a representation-level form of the mismatch between prediction accuracy and task utility (Lambert et al., 2020; Nair et al., 2020). We call this effect *predictability bias*.

Prior task-aware model learning uses specified value functions, goals, or rewards to determine which model distinctions matter for a decision problem (Lambert et al., 2020; Farahmand et al., 2017; Grimm et al., 2020; Nair et al., 2020; Fu et al., 2021). Separately, visual world models differ in the plasticity of their prediction target spaces. Generative models evaluate predictions against fixed pixel observations (Alonso et al., 2024), while other approaches predict representations from a frozen foundational encoder (Zhou et al., 2025). In both cases, the target space is externally defined. To avoid explicit reconstruction, joint-embedding architectures learn their target spaces, but constrain target-encoder updates using stop-gradients and moving averages (Assran et al., 2023). In contrast, fully end-to-end architectures like LeWorldModel jointly optimize the future-feature encoder and the predictor using the prediction loss (Maes et al., 2026). Unlike task-aware methods that rely on explicit signals, we analyze the implicit upstream preference induced when the joint prediction objective shapes the target space. Figure 1 summarizes the resulting representation selection problem.

To formalize this upstream selection problem, we mathematically characterize how predictability bias limits the retention of task-relevant factors. Under certain theoretical assumptions, near-optimal self-prediction bounds the energy of less predictable factors. We derive how specifying a task target counteracts this bias: optimizing for an external target changes the selection geometry, reallocating representation dimensions toward task-relevant coordinates. Under stated operator conditions, we

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

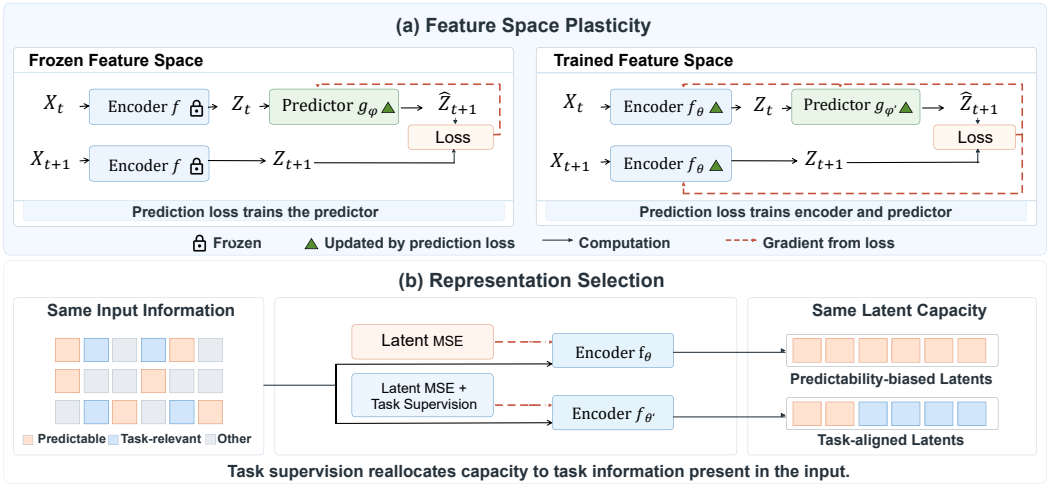


Figure 1: How optimization scope dictates representation selection. (a) Feature Space Plasticity: In a frozen feature space, prediction loss updates the predictor to fit a fixed target. In a jointly trained space, gradients backpropagate to the encoder, allowing the prediction objective to alter the latent space itself. (b) Representation Selection: Given the same input information and finite latent capacity, the training objective determines content selection. Pure latent mean-squared error (MSE) favors predictable factors, yielding predictability-biased latents. Adding task supervision counteracts this bias, reallocating capacity toward task-aligned latents using information already present in the input.

show this target-driven reallocation can improve the native latent score's alignment with downstream task costs, requiring no changes to the original goal-distance metric or deployment planner.

We validate this representation-selection mechanism through a sequence of empirical studies. By isolating prediction from action complexity, we demonstrate that the baseline objective shifts encoded capacity toward predictable scene factors. Introducing target supervision counteracts this shift, reallocating representation energy from predictable backgrounds toward specified targets. We confirm that this selection persists under full action-conditioned prediction on continuous physical trajectories.

Finally, we investigate whether reshaping this upstream selection changes candidate-plan evaluation and impacts native closed-loop planning. Using full simulator-state labels as a deliberately simple training-time intervention, we evaluate the native image-based CEM planner across four continuous control tasks. The complete intervention improves mean closed-loop success in all tasks without changing the deployment planner or providing it with simulator state. Detailed candidate and execution diagnostics reveal that these downstream gains manifest through task-dependent behaviors.

Our contributions are:

- We formally characterize predictability bias by proving that end-to-end self-prediction structurally favors predictively advantaged factors over less predictable task variables, and we establish how explicit target supervision reshapes this upstream selection preference.
- We analyze the geometric consequences of target weighting in a quadratic finite-space class, providing a sufficient operator-alignment condition under which this target-driven reallocation mathematically improves native-score alignment.
- We empirically validate predictability-driven factor selection through controlled interventions, and demonstrate in a four-task case study that upstream target specification improves native closed-loop planning, while revealing that these downstream gains manifest through heterogeneous, task-dependent mechanisms.

2 Related Work

Task relevance in model learning. Task-aware model learning addresses a central mismatch: predictive accuracy need not preserve the distinctions required for control (Lambert et al., 2020). This

issue directly affects latent world models used for planning and imagined control (Ha & Schmidhuber, 2018; Hafner et al., 2019; Hansen et al., 2024). Value-aware model learning and value equivalence define fidelity through value functions (Farahmand et al., 2017; Grimm et al., 2020; Voelcker et al., 2025), whereas goal-aware prediction and task-informed abstractions use goal or reward information (Nair et al., 2020; Fu et al., 2021). Auxiliary probes provide another source of supervision (Zahorodnii, 2025). These approaches specify what a model should retain for a task, but they do not characterize the factor preference when prediction also trains its feature target.

Prediction targets in visual world models. Visual world models differ in what they predict and whether dynamics training can change that target. Pixel-based models predict observed images (Alonso et al., 2024). DINO-WM trains against a frozen DINOv2 encoder (Zhou et al., 2025), and V-JEPA 2-AC freezes its video encoder when fitting an action-conditioned predictor (Assran et al., 2025). LeWorldModel instead updates both the predictor and the encoder supplying future features (Maes et al., 2026). Its LeJEPA-based regularizer constrains representation geometry without specifying the encoded factors (Balestriero & LeCun, 2025). The key distinction is whether prediction operates in a prescribed target space or also helps select that space.

Predictability and dynamical representations. Dynamical representation learning asks which coordinates expose predictable temporal structure. Slow feature analysis selects slowly varying functions under constraints (Wiskott & Sejnowski, 2002). Variational methods identify coherent coordinates (Noé & Nüske, 2013; Wu & Noé, 2020; Mardt et al., 2018), while Koopman methods seek approximately linear evolution (Brunton et al., 2016; Lusch et al., 2018; Otto & Rowley, 2019). Related work connects JEPA to Koopman invariants (Ruiz-Morales et al., 2026) and proves latent-variable recovery for LeJEPA under stated assumptions (Klindt et al., 2026).

3 When Predictability Does Not Imply Control Utility

In this section, we mathematically formalize predictability bias. We demonstrate that because self-prediction naturally rewards predictable content, finite representational capacity can structurally favor a predictively advantaged but task-irrelevant factor. We then theoretically characterize how external target supervision redirects this allocation preference, and identify a sufficient geometric condition under which this shift improves native score alignment without assuming a universal downstream control mechanism.

3.1 The learned prediction target also defines the planning coordinates

A joint-embedding world model learns an encoder f and a latent predictor g. Given a visual observation X_t and action a_t, the one-step prediction is defined as

$$z_t = f(X_t) \in \mathbb{R}^d, \qquad \hat{z}_{t+1} = g(z_t, a_t), \qquad z_{t+1} = f(X_{t+1}). \tag{1}$$

History inputs are suppressed here for notation. The future encoder output supplies the prediction target. The baseline combines latent mean-squared error (MSE) with a representation regularizer:

$$\mathcal{L}_{\text{latent}} = \mathbb{E}\, d^{-1} \|\hat{z}_{t+1} - z_{t+1}\|^2, \qquad \mathcal{J}_{\text{pred}} = \mathcal{L}_{\text{latent}} + \alpha \mathcal{L}_{\text{SIGReg}}, \tag{2}$$

where α weights the regularizer and the expectation averages over training examples and sampled prediction positions. At deployment, the Cross-Entropy Method (CEM) (Rubinstein & Kroese, 2004) rolls out candidate action sequences U over horizon H and scores the predicted terminal latent against the encoding of goal image X_{goal},

$$\hat{D}_H(U) = \|\hat{z}_H(U) - f(X_{\text{goal}})\|^2. \tag{3}$$

The model evaluates candidate plans through their predicted outcomes, and these evaluations inform planning. Since f governs both the future target and the native scoring coordinates, minimizing $\mathcal{J}_{\text{pred}}$ alters the underlying metric of the planner rather than merely improving predictor fit. We formalize this inherent representation-selection mechanism, providing the theoretical foundation for our subsequent target-specification analysis.

3.2 A KOOPMAN FRAMEWORK FOR PREDICTION RISK AND FINITE PREDICTABLE CONTENT

Consider an uncontrolled or fixed-action stationary Markov process with marginal μ. Its stochastic Koopman operator maps an observable to its conditional expectation at the next step:

$$(\mathcal{K}h)(x) = \mathbb{E}[h(X_{t+1}) \mid X_t = x]. \tag{4}$$

Let $L(f, g)$ be one-step MSE and $\mathcal{C}(f) = \mathbb{E}\|\mathcal{K}f(X_t)\|^2$ its *predictable content*. R_{miss} measures predictable content inaccessible from that encoding and R_{fit} measures deviation from the best encoded-input predictor. Conditional-expectation orthogonality gives the general identity

$$L(f, g) = \underbrace{\mathbb{E}\|f(X_{t+1})\|^2 - \mathcal{C}(f)}_{\text{innovation}} + \underbrace{R_{\text{miss}}(f)}_{\text{input insufficiency}} + \underbrace{R_{\text{fit}}(f, g)}_{\text{predictor fitting}}. \tag{5}$$

This decomposition requires square integrability (Appendix A.1). Fitting g cannot remove the first two terms at fixed f; changing f can lower MSE by selecting an easier target.

For the exact finite-dimensional selection results below, we additionally restrict the analytical class by *population whitening*, $\mathbb{E}f(X) = 0$ and $\mathrm{Cov}(f(X)) = I_d$. This normalization fixes latent energy at d and excludes collapse. SIGReg encourages isotropy (Balestriero & LeCun, 2025), but the four-task checkpoints are anisotropic (Appendix B.4); these exact results therefore describe a controlled mechanism. For a linear latent predictor, let the *closure cost* $\delta^2(f)$ be the squared part of $\mathcal{K}f$ outside the span of the current encoder coordinates. Optimizing that linear update gives

$$L^*_{\text{lin}}(f) = d - \mathcal{C}(f) + \delta^2(f). \tag{6}$$

Thus self-prediction rewards *predictable and self-contained* content, $\mathcal{C} - \delta^2$.

Fix an n-dimensional $\mathcal{K}$-invariant observable space ($n \geq d$), and let $S_d = \sum_{i=1}^{d} \sigma_i^2$ denote the sum of its top d squared singular values. Letting $C_d(\varepsilon)$ denote the maximum predictable content achievable by a whitened d-coordinate encoder satisfying $\delta^2(f) \leq \varepsilon$, the standard singular-value bound yields $\mathcal{C}(f) \leq C_d(\varepsilon) \leq S_d$. The resulting content–closure gap, $\Pi_d(\varepsilon) = S_d - C_d(\varepsilon)$, can remain strictly positive: exact whitening does not preclude stationary processes where $\Pi_d(\varepsilon) > 0$ (see Appendix A.1). While this establishes a capacity bottleneck, determining the actual semantics of the omitted content requires analyzing specific factor advantages, which we formalize next.

3.3 PREDICTIVE ADVANTAGE CAN FAVOR A NON-TASK FACTOR

Let B denote an observable factor that serves as a candidate nuisance variable weakly relevant to the downstream task, instantiated empirically as background textures in our controlled studies. To isolate the encoded content beyond B, we define its residual energy via the conditional variance:

$$v(f) = \mathbb{E}\|f(X) - \mathbb{E}[f(X) \mid B]\|^2. \tag{7}$$

Rather than isolating a specific task signal, this *residual factor energy* measures encoded variation not explained by B, including other features and their interactions.

Let $\mathcal{N}$ be the centered B-only subspace and $\mathcal{D}$ the residual subspace. Assume that these subspaces are mutually invariant under the Koopman operator, a condition characteristic of independent stationary factors. The reference factor supplies a realizable top d frame whose linear update belongs to the predictor class. We define its predictive advantage as:

$$\gamma = \sigma_d^2(\mathcal{K}|_{\mathcal{N}}) - \|\mathcal{K}|_{\mathcal{D}}\|_{\text{op}}^2, \tag{8}$$

and its closure cost is ζ; $\xi \geq 0$ bounds excess risk over the same model class's infimum L^*.

Theorem 1 (Near-optimal self-prediction selects the advantaged factor)**.** *Under population whitening and the separation, dimension, and realizability conditions just described, suppose $\gamma > 0$. Every model with $L(f, g) \leq L^* + \xi$ obeys*

$$\boxed{\gamma v(f) \leq \zeta + \xi.} \tag{9}$$

A strong predictability advantage and small slacks bound the residual energy. With zero slacks, every attained optimum depends only on B, even if that factor is not perfectly predictable. A nonpositive gap gives no preference. We defer the proof, alongside a stronger linear variant, to Appendix A.2.

Rather than characterizing the transient optimization dynamics, this conditional bound formalizes the structural limits of capacity allocation. Building upon the general capacity constraints, Equation (9) explicitly identifies how predictive advantage dictates the favored semantic content.

3.4 General supervision reshapes representation and score geometry

The selection theorem evaluates predictive advantage. To specify an alternative content preference, we mathematically introduce a square-integrable training target Y. Using an affine target-prediction head and defining the centered target $Y_c = Y - \mathbb{E}[Y]$, population whitening makes $\mathcal{T}_Y(f) = \|\mathbb{E}[Y_c f(X)^\top]\|_F^2$ the target variance captured by the best affine target predictor. Let $\lambda \geq 0$ weight the target term.

Proposition 2 (Target supervision reshapes representation selection)**.** *For white $Z = f(X)$ and a square-integrable target Y, separately optimizing the shared linear predictor and the affine target-prediction head implies that minimizing their combined weighted risk is equivalent to maximizing:*

$$\underbrace{\mathcal{C}(f) - \delta^2(f)}_{\text{predictable and reusable}} + \underbrace{\lambda \mathcal{T}_Y(f)}_{\text{readable specified target}} \tag{10}$$

over the same feasible representation class. For two feasible representations, a predictive disadvantage $\Delta_{\text{pred}} > 0$ is outweighed by a readable-target advantage $\Delta_{\text{task}} > 0$ whenever $\lambda \Delta_{\text{task}} > \Delta_{\text{pred}}$.

Appendix A.3 gives the decomposition and factor-bound extension. While this proves that target specification redirects capacity allocation along the representation frontier, translating this geometric shift into guaranteed downstream control utility requires specific operator-alignment conditions, which we establish next.

Write $A(f) = \mathcal{C}(f) - \delta^2(f)$ and let f_λ maximize $A(f) + \lambda \mathcal{T}_Y(f)$ over a fixed feasible class. In a finite observable space, Y induces $S_Y = G_Y^* G_Y \succeq 0$, where $G_Y h = \mathbb{E}[Y_c h(X)]$, and $\mathcal{T}_Y(f) = \operatorname{tr}(P_f S_Y)$ for the rank-d representation projection P_f.

Proposition 3 (Target-driven reallocation and operator-aligned score geometry)**.** *For $0 \leq \lambda_1 < \lambda_2$, let $A_i = A(f_{\lambda_i})$ and $T_i = \mathcal{T}_Y(f_{\lambda_i})$. Then*

$$T_2 \geq T_1, \qquad A_2 \leq A_1, \qquad \lambda_1(T_2 - T_1) \leq A_1 - A_2 \leq \lambda_2(T_2 - T_1). \tag{11}$$

If the optimized prediction reward on white rank-d projections has the quadratic form $A(P) = \operatorname{tr}(PQ)$, then every global optimum projects onto a leading d-dimensional eigenspace of $Q + \lambda S_Y$. Consequently, if $S_Y = aW$ for $a > 0$ and $W \succeq 0$, the overlap $\operatorname{tr}(P_\lambda W)$ is nondecreasing with λ.

To establish the downstream score alignment, assume a standardized endpoint–goal residual $U \sim \mathcal{N}(0, \sigma^2 I)$, define the representation-induced score $D_\lambda = U^\top P_\lambda U$, and let the declared task cost be $C_W = U^\top W U$ for $0 \neq W \succeq 0$. Although $D_\lambda \sim \sigma^2 \chi^2_d$ for every λ,

$$\operatorname{Corr}(D_\lambda, C_W) = \frac{\operatorname{tr}(P_\lambda W)}{\sqrt{d \operatorname{tr}(W^2)}}. \tag{12}$$

Consequently, the operator-matched condition $S_Y = aW$ guarantees that increasing the target weight mathematically aligns the native score with the true task cost, even as its marginal distribution remains fixed. We detail the proof and non-isotropic extensions in Appendix A.3.2. This establishes a structural boundary: semantic overlap alone is insufficient to guarantee improved downstream score alignment or closed-loop control utility without explicit operator alignment.

4 A State-Target Intervention at Two Training Interfaces

To empirically investigate the practical implications of our selection theory, we instantiate the external target Y using normalized simulator states paired with offline training images. We employ full-state supervision as a robust empirical intervention, utilizing the complete state as a comprehensive, task-relevant target space to guide representation selection.

We use an affine state head $h_\phi(z) = Wz + b$ to map latents to normalized simulator states for two distinct supervisory objectives. The *encoding objective* measures whether the specified state is linearly decodable from the observed latent, whereas the *prediction objective* evaluates whether this state information is preserved following an action-conditioned latent transition. Given the encoded latent $z_t = f_\theta(X_t)$, the predicted latent $\hat{z}_{t+1}$, and the normalized state $\bar{s}_t \in \mathbb{R}^{d_s}$, the corresponding losses are defined as:

$$\mathcal{L}_E = \mathbb{E}\, d_s^{-1} \|h_\phi(z_t) - \bar{s}_t\|^2, \tag{13}$$

$$\mathcal{L}_P = \mathbb{E}\, d_s^{-1} \|h_\phi(\hat{z}_{t+1}) - \bar{s}_{t+1}\|^2. \tag{14}$$

We parameterize the visual encoder with θ, the action-conditioned predictor with ψ, and the state head with ϕ. To decouple their optimization, we structure the gradient updates as follows:

$$\begin{aligned} G_\theta &= \nabla_\theta[\mathcal{J}_{\text{pred}} + \beta_E \mathcal{L}_E + \beta_P \mathcal{L}_P], \\ G_\psi &= \nabla_\psi[\mathcal{J}_{\text{pred}} + \beta_P \mathcal{L}_P], \\ G_\phi &= \beta_E \nabla_\phi \mathcal{L}_E. \end{aligned} \tag{15}$$

$\mathcal{L}_E$ updates the visual encoder and readout to ensure the declared state is recoverable from observations, serving as the empirical analogue of the target-weighted selection term studied in Propositions 2 and 3. However, this encoding constraint alone does not require the action-conditioned predictor to preserve these readable coordinates. $\mathcal{L}_P$ penalizes one-step prediction errors within the specified state space. We implement a decoupled gradient routing where $\mathcal{L}_P$ updates only the upstream encoding and prediction modules, while the state head is optimized exclusively by $\mathcal{L}_E$. This encourages the state head to dynamically define the target coordinates for the prediction path without its parameters being altered by the prediction loss.

To systematically decompose this intervention, we evaluate three training regimes configured via the encoding and prediction penalties: the pure native objective, encoded-state training, and joint supervision. The nonzero penalty weight came from an earlier configuration and held constant across conditions without task-specific tuning. The contrast between the native objective and encoded-state training bridges to our theoretical analysis. The further contrast with joint supervision evaluates the impact of enforcing this semantic constraint on the transition dynamics.

5 Experiments

The experiments first test factor selection under controlled conditions, then examine relative factor energies in the native architecture with zero actions and with logged physical actions. A unified four-task case study tests whether one specified-target intervention changes native closed-loop control.

5.1 Experimental setup

Unified comparison protocol. We follow the LeWorldModel architecture and base training setup (Maes et al., 2026). To ensure reproducibility, we define a standardized protocol for all evaluations. For the closed-loop evaluation, we compare the native objective (O), encoded-state supervision (E), and joint supervision (J) across four continuous control tasks: Cube, PushT, TwoRoom, and DMControl Reacher. Retrained from matched initializations under a single frozen protocol, the three conditions differ in their penalty weights: $(\beta_E, \beta_P) = (0, 0)$, $(3, 0)$, and $(3, 3)$, respectively. Episodes are partitioned into mutually exclusive 80/10/10 train/validation/test sets before constructing temporal windows, with all normalizers fitted exclusively on training data.

For a paired comparison, all three conditions face the exact same 100 held-out test roots per task and training seed, with identical initial conditions, goals, and CEM random streams. Deployment relies entirely on the unchanged native latent score, without access to the simulator state. Appendix B.4 details the complete hyperparameters, datasets, and seed-level checks.

5.2 Factor selection and target-weighted reallocation

We establish the baseline representation preference using a controlled image dataset with disentangled background and positional factors. Under exact population whitening and independent factor transitions, the temporal predictability gap dictates capacity allocation. When the background is highly persistent, it holds a predictive advantage and consequently monopolizes almost all latent energy, aligning with Theorem 1. Reversing this temporal advantage predictably transfers that capacity dominance to the position factor, with the full persistence comparison in Appendix B.1.

Having isolated this predictability bias, we fix the high-background-persistence environment to test whether explicit target supervision redirects capacity toward less predictable task variables. We jointly optimize the encoder via a closed-form shared linear predictor and an affine target head weighted by λ, detailing the exact optimization setup in Appendix B.1. Figure 2 demonstrates that increasing this supervision weight systematically reallocates latent capacity. By an intermediate weight of

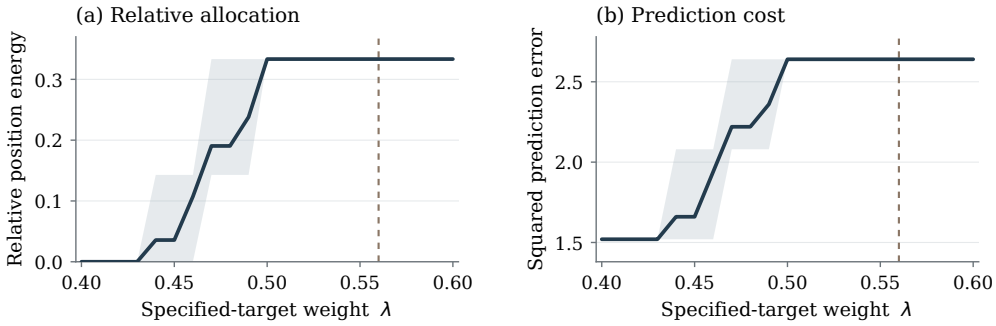


Figure 2: Target-weighted reallocation under exact whitening. (a) Position energy relative to scene energy. (b) Optimized squared one-step prediction error. The curves show four-seed means and shading shows seed ranges. The dashed line marks the analytic comparison $\lambda^* = 0.56$.

$\lambda = 0.50$, the representation successfully retains the target coordinates across all random seeds while predictably sacrificing pure prediction accuracy. This confirms that target specification counters predictability bias before the native tests (Section 5.3; task images in Appendix Fig. 5).

5.3 Predictability-driven selection in the native architecture

We transition from a zero-action baseline which confirms that background persistence increases background-relative embedding variation (Appendix B.2) to action-conditioned prediction. This tests whether the selection preference persists under physical trajectories. In TwoRoom, six branches share four-frame physical trajectories and cross background persistence $p_B \in \{0.1, 0.9\}$ with native prediction, true-label supervision, or trajectory-permuted random-label supervision. Appendix Fig. 4 shows the paired renderings.

To characterize how encoder variation depends on visual factors, we evaluate its output z_{bs} on the complete grid of 96 positions and 256 backgrounds, with grand mean $\bar{z}$. Averaging over positions gives a mean response for each background; E_B is the average squared distance of these responses from $\bar{z}$. Averaging over backgrounds analogously gives the position component E_S. We compute E_{BS} from the residual of each output after removing both mean effects. Total energy is the average squared distance of all outputs from $\bar{z}$ and equals $E_B + E_S + E_{BS}$ on this balanced grid. The ratio $R = \log[(E_B + \varepsilon)/(E_S + E_{BS} + \varepsilon)]$ summarizes the background main effect relative to position and interaction, describing representation geometry rather than state readability.

Training	Persistence	E_B	E_S	E_{BS}	R
Native Objective	Low	74.58	15.78	22.48	0.677
	High	122.05	8.82	21.77	1.392
True-label	Low	64.58	13.46	19.45	0.687
	High	78.08	12.48	20.15	0.887
Random-label	Low	73.11	8.78	16.82	1.057
	High	87.35	8.98	18.18	1.178

Table 1: Factor energies of the native TwoRoom encoder on the held-out grid after 500 matched updates, averaged across four seeds. R is computed for each seed before averaging. Larger values indicate a stronger background main effect relative to position and interaction. Seed-level values and paired intervals appear in Appendix Tables 5 and 6.

As demonstrated in Table 1, the native objective exhibits a severe predictability bias: raising background persistence drastically increases background energy E_B while visibly suppressing state energy E_S. Introducing true-label supervision effectively counteracts this shift, preserving higher absolute state energy and heavily restricting background expansion at high persistence. However, this realloca-

tion is not exclusively semantic. The random-label control also attenuates the persistence response, and offline probes do not immediately resolve a state-readability advantage for true labels. These energy shifts therefore imply that predictability and target specification alter the underlying embedding geometry. Complete paired contrasts and planning diagnostics are deferred to Appendix B.3.

5.4 A UNIFIED SPECIFICATION CASE STUDY

To investigate the practical downstream implications of our selection theory, we use full simulator-state supervision to evaluate native control. While our native checkpoints rely on SIGReg regularization rather than satisfying the exact-whitening theoretical premise, the underlying capacity bottleneck and selection mechanism motivate this simulated control case study. We compare the native objective, encoded-state training, and joint supervision on the four tasks illustrated in Appendix Fig. 5.

	Closed-loop success rate (% ↑)		
Task	Native objective	Encoded-state supervision	Joint supervision
Cube	66.50	75.00	**75.25**
PushT	83.75	87.50	**92.25**
TwoRoom	81.75	86.75	**91.00**
Reacher	85.50	85.75	**96.00**

Table 2: Closed-loop success under the unchanged native latent-goal planner. The columns compare prediction-only training, encoded-state supervision, and joint supervision from encoded observations and predicted latents. Each rate averages four training seeds evaluated on the same 100 held-out roots. Bold indicates the highest mean success rate for each task. Paired differences, 95% intervals, and seed-level results appear in Appendix B.4.

As demonstrated in Table 2, full joint supervision universally improves closed-loop success across all four tasks and training seeds. However, paired contrast analysis reveals that the specific interface driving these gains is task-dependent. For Cube, the performance improvement is almost supported by the encoding interface. Conversely, PushT, TwoRoom, and Reacher see gains from encoded-state supervision alone; their improvements manifest through the addition of the prediction-interface constraint. These results confirm that explicit target specification enhances downstream closed-loop outcomes, though the effective architectural pathway varies by environment.

5.5 ABLATIONS AND MECHANISM DISCUSSION

To unpack the mechanisms, Table 3 brings together three targeted measurements under the unified protocol: the effect of supervision scope on native success, the recoverability of task coordinates from predicted latents, and the physical outcome of committed elite-mean actions.

Supervision scope and task-coordinate prediction. Task-specific supervision J_{task} achieves nearly identical native success to full-state supervision J_{full} on Cube, PushT, and TwoRoom. However, on Reacher, J_{full} achieves 96.00% success while J_{task} drops to 82.75%. Table 3 reveals that J_{task} achieves lower task-coordinate prediction error than the baseline O across all tasks, including Reacher. This discrepancy—where superior prediction of supervised task coordinates fails to preserve the full-state control gains—demonstrates that task-coordinate readability alone may be insufficient for native control, separating pure prediction accuracy from actual downstream utility. Complete full-state probe results are deferred to Appendix B.5.

Committed actions and replanning. Evaluating the physical outcomes of committed elite-mean actions reveals that the timing of performance gains varies heavily by task. For TwoRoom and PushT, the joint supervision J advantage manifests predominantly during the first plan within the initial 25 actions. Conversely, Cube and Reacher rely heavily on the second planning stage for their improvements. Reacher, in particular, exhibits slightly worse first-plan success and higher physical cost than the baseline O, but subsequently rescues a substantially larger fraction of failed roots during replanning, indicating that the intervention alters sequential recovery behavior rather than merely improving initial zero-shot execution.

(a) Success by target scope (%↑)

Task	O	J_{full}	J_{task}
Cube	66.50	**75.25**	74.75
PushT	83.75	**92.25**	**92.25**
TwoRoom	81.75	91.00	**91.50**
Reacher	85.50	**96.00**	82.75

(b) Task-coordinate prediction error ↓

Task	O	J_{full}	J_{task}
Cube	0.111	0.076	**0.027**
PushT	0.385	0.243	**0.213**
TwoRoom	0.168	**0.145**	0.147
Reacher	0.638	0.657	**0.617**

(c1) Mean cost: first plan ↓

Task	O	E	J
Cube	2.001	1.558	1.370
PushT	1.078	1.035	0.893
TwoRoom	1.730	1.416	1.247
Reacher	1.817	1.821	2.117

(c2) Mean cost: second plan ↓

Task	O	E	J
Cube	4.795	4.911	2.633
PushT	3.243	2.485	2.211
TwoRoom	4.621	4.034	3.939
Reacher	1.110	1.129	0.859

Table 3: Downstream mechanism diagnostics and ablations; all values are four-seed means. (a) Native success on 100 held-out roots per seed. (b) Task-coordinate normalized root-mean-squared prediction error at the 25-action prediction horizon. (c1,c2) Executed physical cost of the committed mean of the final top-30 CEM candidates. Panel (c1) includes all 100 test initializations per seed. Panel (c2) reports second-stage costs only for episodes that remain unsuccessful after the first committed plan.

These empirical diagnostics contextualize our theoretical claims. While Propositions 2 and 3 establish how target specification reallocates latent capacity and mathematically aligns score geometry under idealized conditions, the four-task evaluation demonstrates that this upstream geometric shift interacts with native deployment in environment-specific ways. Downstream utility depends inherently on how the modified latent geometry interacts with task-specific search bottlenecks.

6 Limitations

Theoretically, our formal selection mechanism and score-alignment proofs rely on structural assumptions, such as exact population whitening and specific operator matching. Because native checkpoints exhibit anisotropy and empirical full-state supervision does not strictly certify this mathematical operator condition, these theorems establish a controlled foundational mechanism rather than a universal guarantee for unconstrained world models. Empirically, because random-label supervision partially attenuates predictability bias, the precise semantic mediation of the observed control gains remains an open question. Our extensive diagnostics do not identify a single causal mediator that governs the transfer from upstream representation selection to downstream closed-loop success, dictating the need for environment-aware analysis in future world model architectures.

7 Conclusion

When the prediction target is jointly learned, self-prediction structurally biases capacity toward predictable content, inherently risking task misalignment within non-collapsed representations. We formalize this predictability bias, mathematically proving that explicit target specification redirects capacity allocation and establishes a rigorous condition for aligning the native planning score with true task costs. Empirically, our controlled interventions isolate this predictability-driven factor selection, confirming its persistence under real action-conditioned trajectories. Finally, our unified closed-loop evaluation across four continuous control tasks serves as a decisive downstream probe. While the specific behavioral and replanning manifestations vary by environment, this improvement in native success is consistent with our central thesis: the joint prediction objective acts as a profound semantic filter. Overcoming predictability bias at the representation can improve downstream utility, cementing target specification as a critical architectural mechanism in world model design.

AI USE STATEMENT

We used generative AI tools to identify related work, assist with research ideation and execution, including mathematical reasoning, implementation checks, and interpretation of results and draft and polish parts of the manuscript. The authors take responsibility for the final content, citations, analyses, and results.

REPRODUCIBILITY STATEMENT

Appendix B records the controlled processes, model updates, metrics, seeds, planner budget, primary control results, and scope restrictions. Appendices A–A.4 give the assumptions and proofs.

ETHICS STATEMENT

The experiments use simulated environments and existing offline trajectories. They do not establish safe deployment on physical robots. Privileged simulator state is used only where explicitly declared for training or diagnostic evaluation.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6bdde0373d53d4a501249547084bed43-Abstract-Conference.html.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Assran_Self-Supervised_Learning_From_Images_With_a_Joint-Embedding_Predictive_Architecture_CVPR_2023_paper.html.

Mido 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.

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.

Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLOS ONE*, 11(2):e0150171, 2016. doi: 10.1371/journal.pone.0150171.

Amir-Massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494. PMLR, 2017. URL https://proceedings.mlr.press/v54/farahmand17a.html.

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3480–3491, 2021. URL https://proceedings.mlr.press/v139/fu21b.html.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

David Klindt, Yann LeCun, and Randall Balestriero. When does LeJEPA learn a world model? *arXiv preprint arXiv:2605.26379*, 2026. URL https://arxiv.org/abs/2605.26379.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770. PMLR, 2020. URL https://proceedings.mlr.press/v120/lambert20a.html.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0.

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. URL https://arxiv.org/abs/2603.19312.

Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9(1):5, 2018. doi: 10.1038/s41467-017-02388-1.

Suraj Nair, Silvio Savarese, and Chelsea Finn. Goal-aware prediction: Learning to model what matters. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7207–7219, 2020. URL https://proceedings.mlr.press/v119/nair20a.html.

Frank Noé and Feliks Nüske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Modeling & Simulation*, 11(2):635–655, 2013. doi: 10.1137/110858616.

Samuel E Otto and Clarence W Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019. doi: 10.1137/18M1177846.

Zhao-Han Peng, Shaohui Li, Zhi Li, Shulan Ruan, Yu Liu, and You He. From observations to events: Event-aware world models for reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fc09b26b85ab3abb2832bd555a2e4215-Abstract-Conference.html.

Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Information Science and Statistics. Springer New York, 2004. doi: 10.1007/978-1-4757-4321-0.

Pablo Ruiz-Morales, Dries Vanoost, Davy Pissoort, and Mathias Verbeke. Koopman invariants as drivers of emergent time-series clustering in joint-embedding predictive architectures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25176–25183, 2026. doi: 10.1609/aaai.v40i30.39708. Issue 30.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

Claas A Voelcker, Anastasiia Pedan, Arash Ahmadian, Romina Abachi, Igor Gilitschenski, and Amir-Massoud Farahmand. Calibrated value-aware model learning with probabilistic environment models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 61745–61768, 2025. URL https://proceedings.mlr.press/v267/voelcker25a.html.

Laurenz Wiskott and Terrence J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002. doi: 10.1162/089976602317318938.

Hao Wu and Frank Noé. Variational approach for learning Markov processes from time series data. *Journal of Nonlinear Science*, 30:23–66, 2020. doi: 10.1007/s00332-019-09567-y.

Andrii Zahorodnii. Improving world models using deep supervision with linear probes. In *ICLR 2025 Workshop on World Models*, 2025. URL https://openreview.net/forum?id=kSAJdxOUZX.

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, 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

A Assumptions and Proofs

All function-space claims use real $L^2(\mu)$. The population normalization described in the main text is

$$\mathbb{E}f(X) = 0, \qquad \mathbb{E}[f(X)f(X)^\top] = I_d. \tag{16}$$

The stationary conditional-expectation operator is a contraction: $\|\mathcal{K}h\|_2 \le \|h\|_2$. Under this normalization, an encoder $f = (f_1, \dots, f_d)$ defines the isometry $V : \mathbb{R}^d \to L^2_0(\mu)$ by $Va = \sum_i a_i f_i$. For a vector of functions, $\|\mathcal{K}f\|^2_{\mathrm{HS}} = \sum_i \|\mathcal{K}f_i\|^2_2$.

Theorem 1 compares models within one declared feasible class. Its L^* is an infimum over exactly that encoder and predictor class. The reference-factor frame V_B must be observable from the input and realizable by the encoder, and its linear competitor must belong to the predictor class. If the infimum is not attained, Equation (9) applies to near-optimal sequences. Singular-value degeneracy requires fixing a top frame; different frames can have different closure costs ζ.

Population whitening in Equation (16) is an analysis device. SIGReg encourages a Gaussian-like spread through finite batches but does not imply exact population covariance, an unrestricted function class, or the factor decomposition. The native-model experiment measures these discrepancies rather than treating regularization as proof of the assumptions.

A.1 Prediction-risk and projection results

A.1.1 Isotropy as a fixed-direction minimax reference

The main-text remark uses the following elementary comparison, separately from the prediction-risk analysis.

Lemma 1 (Task-agnostic isotropy and its task-specific limit)**.** *Let $S \in \mathbb{R}^m$ be centered with $\mathrm{Cov}(S) = I_m$, let $Z = LS$ with $L \in \mathbb{R}^{m\times m}$, and fix $\mathrm{tr}(L^\top L) = c > 0$. Then*

$$\max_{L:\,\|L\|_F^2 = c} \; \min_{\|u\|=1} \|Lu\|^2 = c/m, \tag{17}$$

and $L^\top L = (c/m)I_m$ characterizes all maximizers. For a specified unit direction w, under the same trace constraint, $\max_L \|Lw\|^2 = c$, attained at $L^\top L = cww^\top$. If $m > 1$, isotropy is strictly suboptimal for this directional objective.

Write $G = L^\top L \succeq 0$ and let its eigenvalues be $\eta_1, \dots, \eta_m$. Since $\mathrm{Cov}(S) = I_m$, the latent covariance is $LL^\top$, with the same spectrum and trace as G. Rayleigh's principle gives

$$\min_{\|u\|=1} \|Lu\|^2 = \lambda_{\min}(G) \le \frac{1}{m}\sum_{i=1}^m \eta_i = \frac{c}{m}.$$

Equality holds if and only if all eigenvalues equal c/m, so the optimal metric and covariance are both isotropic; L itself is not unique. Taking $c = m$ gives white output. The conclusion needs only second moments, not Gaussianity.

For a fixed unit w, $w^\top G w \le \mathrm{tr}\, G = c$, with equality at $G = cww^\top$. Even if singular codes are excluded and $m > 1$, the full-rank metric

$$G_a = (c - (m-1)a)ww^\top + a(I_m - ww^\top), \qquad 0 < a < c/m,$$

has trace c and $w^\top G_a w = c - (m-1)a > c/m$. Thus the task-specific improvement need not rely on collapse. It does change the covariance shape: this comparison is in the trace-constrained class, not among codes already required to be white.

This geometric result does not identify latent covariance with the sensitivity of an arbitrary nonlinear encoder to physical perturbations. Nor does whitening choose the encoded variables: for independent standardized scalars B, S, the codes $f_B(B,S) = B$ and $f_S(B,S) = S$ both have unit variance, but their optimal squared-error state-prediction risks for target S are respectively 1 and 0. The first code really omits this independent target; no such target-loss conclusion follows from an energy ratio alone.

A.1.2 FINITE-SPACE CAPACITY BOUND AND POSITIVE-GAP CONSTRUCTION

Precise definitions. Let $V : \mathbb{R}^d \to L^2_0(\mu)$ have columns f_j, so population whitening makes V an isometry. Write $\Pi = VV^*$ for projection onto their span and $A = V^*\mathcal{K}V$ for the best shared linear update. The content and closure quantities used in the main text satisfy

$$\mathcal{C}(V) = \|\mathcal{K}V\|^2_{\mathrm{HS}} = \|A\|^2_F + \delta^2(V), \qquad \delta^2(V) = \|(I - \Pi)\mathcal{K}V\|^2_{\mathrm{HS}}. \tag{18}$$

We use $\mathcal{C}(f)$ and $\mathcal{C}(V)$ interchangeably, and likewise for δ^2. On the declared n-dimensional invariant space with singular values $\sigma_1 \geq \cdots \geq \sigma_n$, the exact frontier definitions are

$$S_d = \sum_{i=1}^{d} \sigma_i^2, \qquad C_d(\varepsilon) = \max_{\delta^2(V) \leq \varepsilon} \mathcal{C}(V), \qquad \Pi_d(\varepsilon) = S_d - C_d(\varepsilon). \tag{19}$$

All maxima range over white d-frames in this space. The nonempty feasible-set condition is required. The additional shared-capacity conclusion, with $Q_d(\varepsilon) := \max_{\delta^2(V) \leq \varepsilon} \|V^*\mathcal{K}V\|^2_F$, is

$$\max\{0, C_d(\varepsilon) - \varepsilon\} \leq Q_d(\varepsilon) \leq C_d(\varepsilon). \tag{20}$$

Use orthonormal coordinates in the declared invariant function space, so that $\mathcal{K}$ is an n-dimensional contraction. The white frames form the compact Stiefel manifold. Since $\mathcal{C}$ and δ^2 are continuous, the assumed nonempty closure-constrained set is compact and both maxima exist.

Let e_i be right singular vectors and let $a_i = \|V^*e_i\|^2$. Orthogonality gives $0 \leq a_i \leq 1$ and $\sum_i a_i = d$. Consequently

$$\mathcal{C}(V) = \mathrm{tr}(V^*\mathcal{K}^*\mathcal{K}V) = \sum_{i=1}^{n} \sigma_i^2 a_i \leq \sum_{i=1}^{d} \sigma_i^2 = S_d.$$

The unconstrained spectral maximum is attained by a top right-singular frame, which need not obey the closure budget. This proves $\Pi_d \geq 0$ and the content bounds. For every feasible frame, orthogonal projection gives $\|V^*\mathcal{K}V\|^2_F = \mathcal{C}(V) - \delta^2(V) \leq C_d$. Evaluating this identity at a content-maximizing frame gives $Q_d \geq C_d - \varepsilon$; nonnegativity gives the other lower bound. It does not assert that the maximizers of Q_d and C_d coincide.

Relative to all n singular directions, the missing predictable content is

$$\sum_{i=1}^{n} \sigma_i^2 - C_d(\varepsilon) = \sum_{i>d} \sigma_i^2 + \Pi_d(\varepsilon).$$

The first term is the finite-dimensional truncation; the second is the extra shared-closure restriction. Neither term specifies the semantics of what is omitted.

For a concrete stationary realization, take independent standard Gaussian ξ_t and $X_t = (\xi_t, \xi_{t-1})$. The orthonormal coordinate functions $h_1(X_t) = \xi_t$ and $h_2(X_t) = \xi_{t-1}$ span an invariant space with $\mathcal{K}h_1 = 0$, $\mathcal{K}h_2 = h_1$. Every real unit frame is $v = (a, b)^\top$, $a^2 + b^2 = 1$. Put $x = b^2$. Direct computation gives

$$\mathcal{C}(v) = x, \quad v^\top \mathcal{K} v = ab, \quad \delta^2(v) = x - a^2b^2 = x^2.$$

Maximizing $x \in [0, 1]$ under $x^2 \leq \varepsilon$ gives $C_1(\varepsilon) = \min\{\sqrt{\varepsilon}, 1\}$. Here $S_1 = 1$, so this positive-gap example follows despite exact whitening. Its shared capacity is $\sqrt{\varepsilon} - \varepsilon$ for $0 \leq \varepsilon \leq 1/4$ and $1/4$ for $\varepsilon \geq 1/4$. By contrast, a self-adjoint restriction has an invariant frame of its top absolute eigenvalues, attaining S_d with $\delta = 0$ and hence $\Pi_d = 0$ for every $\varepsilon \geq 0$.

These are finite-space results, not an assumption that arbitrary nonlinear encoders have a mutual-information budget. A restricted neural encoder class may have an additional realizability deficit. The positive-gap example establishes a possibility, not generic positivity or task loss in the measured checkpoints.

A.1.3 RISK DECOMPOSITION AND SHARED PROJECTION

The one-step risk in the uncontrolled or fixed-action analysis is

$$L(f, g) = \mathbb{E}\|g(f(X_t)) - f(X_{t+1})\|^2. \tag{21}$$

The two residual terms in Equation (5) are exactly

$$R_{\text{miss}}(f) = \|\mathcal{K}f - \mathbb{E}[\mathcal{K}f \mid f]\|_{\text{HS}}^2,$$
$$R_{\text{fit}}(f,g) = \mathbb{E}\|g(f(X_t)) - \mathbb{E}[\mathcal{K}f(X_t) \mid f(X_t)]\|^2. \tag{22}$$

Thus the full conditional-expectation identity is

$$L(f,g) = \mathbb{E}\|f(X_{t+1})\|^2 - \|\mathcal{K}f\|_{\text{HS}}^2 + \|\mathcal{K}f - \mathbb{E}[\mathcal{K}f \mid f]\|_{\text{HS}}^2 + \mathbb{E}\|g(f(X_t)) - \mathbb{E}[\mathcal{K}f(X_t) \mid f(X_t)]\|^2. \tag{23}$$

The following lemma proves this identity and its compact main-text form.

Lemma 2 (Conditional-mean risk decomposition). *Let $Y \in L^2$, let X be the available input, let $Z = T(X)$, and write $m_X = \mathbb{E}[Y \mid X]$ and $m_Z = \mathbb{E}[Y \mid Z]$. For any $g(Z) \in L^2$,*

$$\mathbb{E}\|Y - g(Z)\|^2 = \mathbb{E}\|Y - m_X\|^2 + \mathbb{E}\|m_X - m_Z\|^2 + \mathbb{E}\|m_Z - g(Z)\|^2. \tag{24}$$

Proof. Write $Y - g = (Y - m_X) + (m_X - m_Z) + (m_Z - g)$. The first summand is orthogonal to every X-measurable function, and the second is orthogonal to every Z-measurable function because $m_Z = \mathbb{E}[m_X \mid Z]$. Expanding the squared norm proves the identity. □

For $Y = f(X_{t+1})$, $X = X_t$, and $Z = f(X_t)$, we have $m_X = \mathcal{K}f$, which proves the general identity in Equations (5) and (23). Under population whitening, $\mathbb{E}\|Y\|^2 = d$, so the innovation term specializes to

$$\mathbb{E}\|Y - m_X\|^2 = d - \|\mathcal{K}f\|_{\text{HS}}^2, \tag{25}$$

which is the form used by the finite-dimensional results.

Lemma 3 (Content and shared closure). *For an isometry V and $A = V^*\mathcal{K}V$,*

$$\mathcal{K}V = VA + (I - VV^*)\mathcal{K}V \tag{26}$$

is an orthogonal decomposition. Consequently, $\|\mathcal{K}V\|_{\text{HS}}^2 = \|A\|_F^2 + \delta^2(V)$, and A minimizes $\|\mathcal{K}V - VM\|_{\text{HS}}^2$ over matrices M.

Proof. VV^* is the orthogonal projection onto $\operatorname{ran} V$. Hence the two terms in Equation (26) are orthogonal, and $\|VA\|_{\text{HS}} = \|A\|_F$. Orthogonal projection is the least-squares minimizer. Combining this approximation error with Equation (25) gives Equation (6). □

A.2 Proof of the factor-selection theorem

Formal comparison class and gap. Here B denotes the observable reference factor, not a spatial background partition. The proof uses its conditional-expectation subspace; nuisance status depends on the specified task. Let $P_B h = \mathbb{E}[h(X) \mid B]$ and $Q_B = I - P_B$. The declared class is $\mathcal{H}_{\text{adm}} = \mathcal{N} \oplus \mathcal{D}$, with $\mathcal{N} \subset L_0^2(B)$ and $\mathcal{D} \subset \ker P_B$. The separation condition used in Theorem 1 is

$$\mathcal{K}P_B = P_B\mathcal{K}. \tag{27}$$

Let $\lambda_1 \geq \cdots \geq \lambda_d$ be the squared top d singular values of $\mathcal{K}|_{\mathcal{N}}$. Their right singular frame V_B must be observable and realizable by the encoder class, and its shared linear update must be in the predictor class. Define

$$M = \sum_{i=1}^{d} \lambda_i, \qquad \rho_D = \|\mathcal{K}|_{\mathcal{D}}\|_{\text{op}}, \qquad \gamma = \lambda_d - \rho_D^2, \qquad \zeta = \|(I - V_B V_B^*)\mathcal{K}V_B\|_{\text{HS}}^2. \tag{28}$$

These are respectively reference-factor content, residual predictive norm, the squared-strength gap, and reference-factor closure cost. The main result requires $\gamma > 0$ and compares $L(f,g) \leq L^* + \xi$ to the infimum over this same feasible model class.

Let $f^B = P_B f$ and $f^D = Q_B f$. Since P_B is an orthogonal projection and f is white,

$$\|f^B\|_{\text{HS}}^2 = d - v(f), \qquad \|f^D\|_{\text{HS}}^2 = v(f). \tag{29}$$

Commutation in Equation (27) makes the two subspaces reducing for $\mathcal{K}$, so $\mathcal{K}f^B$ and $\mathcal{K}f^D$ are orthogonal. The definition of ρ_D gives

$$\|\mathcal{K}f^D\|_{\mathrm{HS}}^2 \le \rho_D^2 v(f). \tag{30}$$

Let (e_i) be right singular functions of $\mathcal{K}|_{\mathcal{N}}$ and let their squared singular values be $\lambda_1 \ge \lambda_2 \ge \cdots$. Define occupancy weights $w_i = \|V^* e_i\|^2$. Because $VV^* \preceq I$, $0 \le w_i \le 1$; completeness on $\mathcal{N}$ and Equation (29) give $\sum_i w_i = d - v(f)$. Set $a = \lambda_d$. Then

$$\begin{aligned}
\|\mathcal{K}f^B\|_{\mathrm{HS}}^2 &= \sum_i \lambda_i w_i \\
&= M - \sum_{i \le d} \lambda_i (1 - w_i) + \sum_{i > d} \lambda_i w_i \\
&\le M - a \left[\sum_{i \le d} (1 - w_i) - \sum_{i > d} w_i \right] = M - a v(f).
\end{aligned} \tag{31}$$

The bracket equals $d - \sum_i w_i = v(f)$. Combining Equations (30) and (31) yields the content bound; Equation (25) then gives the risk bound for every predictor:

$$\|\mathcal{K}f\|_{\mathrm{HS}}^2 \le M - \gamma v(f), \qquad L(f, g) \ge d - M + \gamma v(f). \tag{32}$$

For the realizable reference-factor frame V_B, Lemma 3 gives shared linear risk

$$L_{\mathrm{lin}}^*(V_B) = d - M + \zeta. \tag{33}$$

The assumed predictor class contains this competitor, so $L^* \le d - M + \zeta$. If $L(f, g) \le L^* + \xi$, then $d - M + \gamma v(f) \le d - M + \zeta + \xi$, proving Equation (9).

For the optimal shared linear predictor of f, Lemma 3 and the content upper bound give

$$L_{\mathrm{lin}}^*(f) = d - \|\mathcal{K}f\|_{\mathrm{HS}}^2 + \delta^2(f) \ge d - M + \gamma v(f) + \delta^2(f). \tag{34}$$

Comparing to the same reference-factor competitor proves the strengthened claim $\gamma v(f) + \delta^2(f) \le \zeta + \xi$.

A useful sufficient process. For independent refresh kernels $K_B = p_B I + (1 - p_B)\Pi_B$ and $K_S = p_S I + (1 - p_S)\Pi_S$, every centered B-only function has singular value p_B, while the operator norm on centered state and interaction functions is $\max\{p_S, p_B p_S\} = p_S$. Thus $\gamma = p_B^2 - p_S^2$ and a B-only frame is invariant, giving $\zeta = 0$. This is the process used in Section B.1.

When the theorem cannot be invoked. For a deterministic measure-preserving full state, $\|\mathcal{K}h\| = \|h\|$ and the unrestricted dynamic norm can equal one. A history window containing overlapping frames also has exactly preserved coordinate functions. A constant reference factor has a zero-dimensional centered function space. These cases require a predeclared restricted function class or a closure analysis; an empirical compressed operator norm cannot replace the true ρ_D without an additional assumption.

Auxiliary target-risk decomposition. Equation (43) follows from Lemma 2 and minimization of the final orthogonal term over $\mathcal{H}$. This identity specifies the training target; it is not a control-improvement guarantee.

A.3 General target weighting and score geometry

This section proves Propositions 2 and 3, then records what a specified target constrains without treating downstream control as a consequence of target readability.

A.3.1 Proof of Proposition 2

Let $Z = f(X)$ have zero mean and covariance I_d, let $Y \in L^2$, and set $Y_c = Y - \mathbb{E}Y$ and $M = \mathbb{E}[Y_c Z^\top]$. Minimizing over the intercept gives $b^* = \mathbb{E}Y$. Completing the square gives

$$\begin{aligned}\mathbb{E}\|WZ - Y_c\|^2 &= \mathbb{E}\|Y_c\|^2 + \operatorname{tr}(WW^\top) - 2\operatorname{tr}(WM^\top) \\ &= \mathbb{E}\|Y_c\|^2 - \|M\|_F^2 + \|W - M\|_F^2. \end{aligned} \tag{35}$$

Hence $W^* = M$ and the exact affine-regression identity is

$$\inf_{W,b} \mathbb{E}\|WZ + b - Y\|^2 = \mathbb{E}\|Y_c\|^2 - \mathcal{T}_Y(f). \tag{36}$$

Combining with Equation (6) yields, for separately optimized latent prediction and target regression,

$$L^*_{\text{lin}}(f) + \lambda L^*_Y(f) = d + \lambda\mathbb{E}\|Y_c\|^2 - [\mathcal{C}(f) - \delta^2(f) + \lambda\mathcal{T}_Y(f)], \tag{37}$$

which proves Equation (10). A fixed dimension normalization of the state loss can be absorbed into λ; task-coordinate scaling must remain fixed.

For two feasible representations with predictive-risk difference $\Delta_{\text{pred}} = L^*_{\text{lin}}(f_S) - L^*_{\text{lin}}(f_B) > 0$ and task-energy difference $\Delta_{\text{task}} = \mathcal{T}_Y(f_S) - \mathcal{T}_Y(f_B) > 0$, their joint-risk difference is $\Delta_{\text{pred}} - \lambda\Delta_{\text{task}}$. Thus supervision can reverse their preference. This comparison alone neither excludes a better third representation nor determines the native ranking of planning candidates.

A.3.2 Proof and scope of Proposition 3

Let $f_i = f_{\lambda_i}$, $A_i = A(f_i)$, and $T_i = \mathcal{T}_Y(f_i)$. Optimality at the two weights gives

$$\begin{aligned} A_1 + \lambda_1 T_1 &\geq A_2 + \lambda_1 T_2, \\ A_2 + \lambda_2 T_2 &\geq A_1 + \lambda_2 T_1. \end{aligned} \tag{38}$$

Adding the inequalities shows $(\lambda_2 - \lambda_1)(T_2 - T_1) \geq 0$, hence $T_2 \geq T_1$. Rearranging each one gives Equation (11); because $\lambda_1 \geq 0$, it also gives $A_2 \leq A_1$. Thus stronger supervision cannot reduce the supervised energy of a global optimum, but it can trade away the pure prediction reward. Equivalently, $F(\lambda) = \max_f [A(f) + \lambda\mathcal{T}_Y(f)]$ is convex and T_λ is a subgradient; at a unique differentiable optimum, $F'(\lambda) = T_\lambda$.

For completeness, define $G_Y h = \mathbb{E}[Y_c h(X)]$ on the declared finite observable space. Then $S_Y = G_Y^* G_Y \succeq 0$, and for the isometric frame V with projection $P = VV^*$,

$$\mathcal{T}_Y(V) = \|G_Y V\|^2_{\text{HS}} = \operatorname{tr}(V^* S_Y V) = \operatorname{tr}(P S_Y). \tag{39}$$

If $A(P) = \operatorname{tr}(PQ)$, the joint reward is $\operatorname{tr}[P(Q + \lambda S_Y)]$. Ky Fan's variational principle makes every maximizing rank-d projection a leading d-dimensional spectral projection of $Q + \lambda S_Y$. When Q and S_Y are simultaneously diagonal with eigenvalues q_i, s_i, respectively, the selected directions are the d largest $q_i + \lambda s_i$. A direction j with $s_j > s_i$ overtakes direction i at $\lambda = (q_i - q_j)/(s_j - s_i)$; exact ties make the optimum nonunique.

For the score statement, let $R \sim \mathcal{N}(0, \sigma^2 I_n)$. Any rank-$d$ projection can be written $P = VV^\top$ with $V^\top V = I_d$, so $V^\top R \sim \mathcal{N}(0, \sigma^2 I_d)$ and $D_P = R^\top P R \sim \sigma^2 \chi^2_d$. For symmetric matrices B, C and centered isotropic Gaussian R, Isserlis' identity gives

$$\operatorname{Cov}(R^\top B R, R^\top C R) = 2\sigma^4 \operatorname{tr}(BC). \tag{40}$$

Taking $B = P$ and $C = W$, and using $P^2 = P$, yields $\operatorname{Var}(D_P) = 2\sigma^4 d$, $\operatorname{Var}(C_W) = 2\sigma^4 \operatorname{tr}(W^2)$, and Equation (12). If $S_Y = aW$ with $a > 0$, the comparative-statics result makes $\operatorname{tr}(P_\lambda W) = a^{-1} T_\lambda$ nondecreasing.

For a non-isotropic Gaussian residual $R \sim \mathcal{N}(\mu, \Omega)$, the same representation score is a generalized noncentral chi-square variable with

$$\begin{aligned} \mathbb{E}D_P &= \operatorname{tr}(P\Omega) + \mu^\top P \mu, \\ \operatorname{Cov}(D_P, C_W) &= 2\operatorname{tr}(P\Omega W \Omega) + 4\mu^\top P \Omega W \mu. \end{aligned} \tag{41}$$

Thus alignment outside the isotropic specialization depends on the residual law as well as the subspace overlap. Likewise, if S_Y is not a positive multiple of W, supervision can increase $\operatorname{tr}(P_\lambda S_Y)$ while decreasing $\operatorname{tr}(P_\lambda W)$; no task benefit follows without an alignment condition.

Target advantage modifies the factor-selection bound. Under the hypotheses of Theorem 1, let S_B be the content of the realizable top reference-factor frame f_B, and let $\delta^2(f_B) = \zeta$. The proof of that theorem gives $\mathcal{C}(f) \le S_B - \gamma v(f)$. If the separately optimized joint risk of f_λ is within ξ of its infimum on the same feasible class, comparison with f_B yields

$$\mathcal{C}(f_\lambda) - \delta^2(f_\lambda) + \lambda \mathcal{T}_Y(f_\lambda) \ge S_B - \zeta + \lambda \mathcal{T}_Y(f_B) - \xi.$$

Combining the two inequalities gives

$$\gamma v(f_\lambda) + \delta^2(f_\lambda) \le \zeta + \xi + \lambda[\mathcal{T}_Y(f_\lambda) - \mathcal{T}_Y(f_B)]. \tag{42}$$

At $\lambda = 0$ this recovers the stronger linear-predictor selection bound. An external target can compensate for a predictive disadvantage through a readable-energy advantage. The inequality does not guarantee that increasing λ increases v in actual training, and its ξ refers to the joint population objective, not a measured neural-optimization slack.

Relation to the spectral frontier. The finite-space analysis in Appendix A.1 defines the content–closure frontier and its spectral tax. Target specification changes the objective on the same feasible class, not its dimension or frontier. Neither a positive spectral tax nor its reduction by itself implies native planning or control utility.

A.3.3 What a specified target constrains

Let $Z = f(X)$, $m_X = \mathbb{E}[Y \mid X]$, and $m_Z = \mathbb{E}[Y \mid Z]$. For a declared prediction class $\mathcal{H}$, squared target prediction satisfies

$$\inf_{h \in \mathcal{H}} \mathbb{E}\|Y - h(Z)\|^2 = \underbrace{\mathbb{E}\|Y - m_X\|^2}_{R_{\mathrm{irr}}} + \underbrace{\mathbb{E}\|m_X - m_Z\|^2}_{D_Y(f)} + \underbrace{\inf_{h \in \mathcal{H}} \mathbb{E}\|m_Z - h(Z)\|^2}_{A_{\mathcal{H}}(f)}. \tag{43}$$

The auxiliary loss penalizes both loss of target-relevant conditional means and failure to expose retained information to the chosen readout class. With an affine head, the last term measures affine accessibility; Proposition 2 specializes this risk to a task-energy reward under whitening. Neither D_Y nor $A_{\mathcal{H}}$ is a fixed fitting constant.

A.4 Supplementary rollout error decomposition

This supplementary identity separates local prediction residuals from changes caused by recursively constructed contexts. It is not offered as a unified explanation of the observed control gains. Long-horizon world-model prediction is also studied through stochastic dynamics, value targets, richer generative models, and event-aware representations (Hafner et al., 2019; Hansen et al., 2024; Alonso et al., 2024; Peng et al., 2026).

From one-step prediction to imagined trajectories. Planning also feeds predictions back into the model. Let c_t and $\hat{c}_t$ be the true and recursively constructed latent histories under the same actions, and set $e_t = z_t - \hat{z}_t$.

Lemma 4 (Local and propagated rollout errors)**.** *For any fixed encoder and predictor, define the true-context residual $r_{t+1} = z_{t+1} - g(c_t, a_t)$ and the context-induced change $\Delta_{t+1} = g(c_t, a_t) - g(\hat{c}_t, a_t)$. Then, pathwise,*

$$e_{t+1} = r_{t+1} + \Delta_{t+1}, \tag{44}$$

$$\|e_{t+1}\|^2 = \|r_{t+1}\|^2 + \|\Delta_{t+1}\|^2 + 2\langle r_{t+1}, \Delta_{t+1} \rangle. \tag{45}$$

The proof below adds and subtracts the true-context prediction. Multi-step error combines local prediction error with feedback from earlier errors; the cross term can amplify or cancel them. Thus one-step accuracy alone does not determine the terminal predictions used for planning, nor must error grow monotonically with horizon.

Proof of Lemma 4. With the same action a_t, substitute $\hat{z}_{t+1} = g(\hat{c}_t, a_t)$ into $e_{t+1} = z_{t+1} - \hat{z}_{t+1}$ and add and subtract $g(c_t, a_t)$. This gives Equation (44); expanding its squared norm gives Equation (45). The identity holds for realized stochastic trajectories and arbitrary nonlinear predictors, without whitening, independence, or a zero cross term. The history $\hat{c}_t$ contains the same initial observed latents as c_t, with later entries replaced by recursive predictions. □

If g is locally L_t-Lipschitz in its history, then $\|e_{t+1}\| \le \|r_{t+1}\| + L_t\|c_t - \hat{c}_t\|$. For a stacked history of length k, the squared context norm equals the sum of the corresponding k squared latent errors (initial observed entries have zero error). Average one-step MSE does not bound this feedback uniformly.

For a fixed goal latent z_g, the native terminal score changes by

$$\|\hat{z}_H - z_g\|^2 - \|z_H - z_g\|^2 = \|e_H\|^2 - 2\langle z_H - z_g, e_H\rangle. \tag{46}$$

This connects rollout error to the deployed score, but its direction relative to the goal matters. It neither equates latent score with physical task cost nor gives a monotone relation between prediction MSE and control success.

In the stationary linear shared-frame setting with $A = V^*\mathcal{K}V$,

$$e_H^{(2)} := \mathbb{E}\|f(X_{t+H}) - (A^\top)^H f(X_t)\|^2 = d - \|\mathcal{K}^H V\|_{\mathrm{HS}}^2 + \|\mathcal{K}^H V - VA^H\|_{\mathrm{HS}}^2, \tag{47}$$

and

$$\mathcal{K}^H V - VA^H = \sum_{j=0}^{H-1} \mathcal{K}^{H-1-j}(\mathcal{K}V - VA)A^j. \tag{48}$$

The first follows by applying the one-step projection identity to the H-step target; the second follows by induction. Because $\|A\|_{\mathrm{op}} \le 1$, a worst-case envelope is linear in $H\delta$, but our registered synthetic studies find such norm envelopes too loose to guide model or horizon selection.

B Experimental Protocols and Seed-Level Results

This section reports the protocols that support the paper's stated claims.

B.1 Controlled CNN: persistence, supervision, and pixel response

A finite visual process with matched training. We cross 16 textured backgrounds with 16 locations of a red disk, giving 256 distinct 64×64 RGB images. At each transition, background and position independently retain their identities with probabilities p_B and p_S, respectively, and otherwise refresh from their uniform marginals. Thus p_B denotes background *persistence*; a background redraw is attempted with probability $1 - p_B$ and can return the same identity. Changing either transition probability leaves the single-frame image distribution fixed. We hold $p_S = 0.5$ and train an eight-dimensional CNN at $p_B \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, with no target loss ($\lambda = 0$) or a position-readout loss ($\lambda = 0.6$). Each of the 40 runs receives 1,000 full-distribution Adam updates; the four seeds $1, 7, 42, 3072$ use matched initial weights across all ten conditions. This five-level grid is a post-hoc sensitivity and visualization analysis of the same finite process.

The pixel encoder has three GELU convolutional layers (channels $3 \to 16 \to 32 \to 32$, kernels $5, 3, 3$, stride two), a 64-unit GELU layer, and an eight-unit output. Each update encodes all 256 images, centers and Cholesky-whitens the outputs, then solves the shared one-step linear predictor and affine readout of two centered, unit-variance position coordinates in closed form. The optimized objective is prediction risk plus λ times position-readout risk; gradients pass through whitening and both analytic solutions to the CNN. Scene and position identities specify the controlled transition and permit post-training factor decomposition, but are not supplied to the pixel encoder. The same finite image bank is used to train and measure this mechanism, so the experiment does not test image generalization.

Where supervision changes the selected factors. For the factorial latent array z_{bs}, averaging over the other factor and subtracting the grand mean gives orthogonal background, position, and interaction main effects with $E_B + E_S + E_{BS} = 8$ under exact whitening. At $p_B = 0.1$ and 0.3, both training conditions allocate almost all eight units to position; supervision still raises the linearly readable fraction of the two target coordinates from approximately 0.64 to 1. At $p_B = 0.7$ and 0.9, the unsupervised models allocate almost all eight units to background, whereas supervision yields $(E_B, E_S) \simeq (6, 2)$ in every seed. The $p_B = 0.5$ row lies between these regimes: mean background energy changes from 0.132 to 0.342, with seed-dependent variation, so it does not support a monotonic allocation claim. Interaction energy is below 1.1×10^{-4} in every cell. At

$p_B = 0.9$, the change from $\lambda = 0$ to 0.6 also raises optimized prediction risk from 1.52 to 2.64, exposing the prediction cost of retaining target coordinates.

Pixel-level visualization. Figure 3 holds one rendered pair fixed across all trained conditions for each intervention: either move the disk from position 5 to 10 on background 2, or replace background 2 by 5 at position 5. For each pair, $D = \frac{1}{2}\|z(x_1) - z(x_0)\|_2^2$. Signed pixel contributions integrate the encoder Jacobian along the straight path between the two images using 32 Gauss–Legendre nodes and the saved whitening statistics. The maps average contributions over four seeds before plotting; their pixel sum equals the corresponding mean pair distance up to numerical quadrature error. At $p_B = 0.7$ and 0.9, supervision raises this fixed foreground-pair distance from below 5.1×10^{-5} to about 0.80, while reducing the background-pair distance from about 10 to 6.5–6.8. Since only modified pixels can contribute, a hotspot's location alone does not establish model attention. The curve instead decomposes all 256 images and must not be read as the distance of the displayed pair.

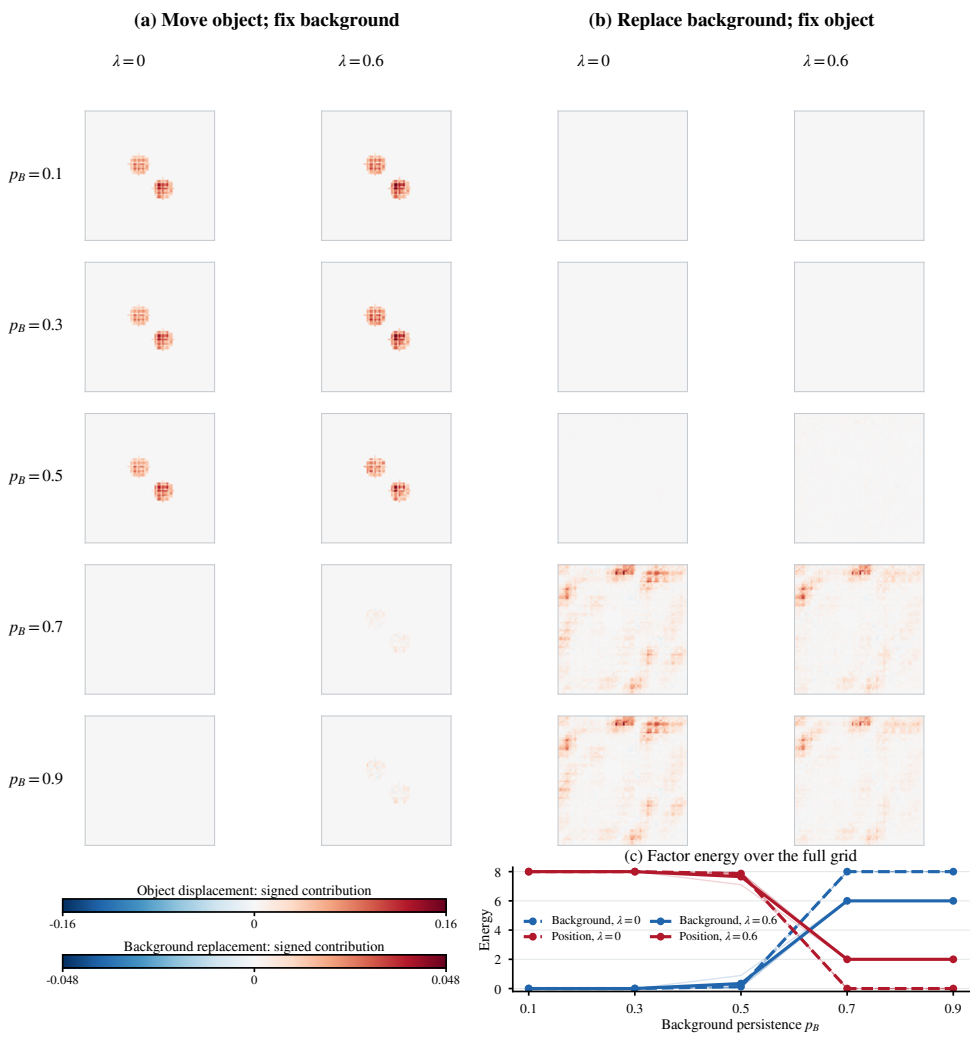


Figure 3: Four-seed counterfactual response of the exact-whitening CNN. Rows vary background persistence p_B; within each intervention, columns compare no position supervision ($\lambda = 0$) and position supervision ($\lambda = 0.6$). Panels (a) and (b) show mean signed pixel contributions to one fixed pair's encoded distance, with a shared color scale within each intervention. Panel (c) shows background (blue) and position (red) main-effect energies over the full factorial grid; dashed lines denote $\lambda = 0$, solid lines $\lambda = 0.6$, and faint lines individual seeds. At high persistence, supervision restores a position response and reduces background allocation.

Relation to the target-weight sweep. The fixed-$p_B = 0.9$ sweep in Figure 2 asks a complementary question: how the selected factors and optimized prediction error change as λ varies. The original four-seed settings are $\lambda \in \{0, 0.2, 0.4, 0.55, 0.6, 0.8, 1, 2\}$; a post-hoc independently trained follow-up covers 0.40–0.60 in increments of 0.01. Repeated settings reproduce the original cells. All seeds have negligible target readability through $\lambda = 0.43$, acquire intermediate readable directions at 0.44–0.49, and retain both target directions by 0.50. The optimized prediction risk rises from approximately 1.52 to 2.64. The analytic comparison $\lambda^* = p_B^2 - p_S^2 = 0.56$ ranks the two pure factor representations; it is not an observed training threshold. Indeed, the learned target-readable solution at $\lambda = 0.50$ has joint risk about 2.64, above the scene competitor's $1.52 + 2(0.50) = 2.52$. This separates a finite-network optimization path from the conditional population selection result.

All 40 new cells completed. The maximum covariance-whitening error is 4.76×10^{-15}, the maximum factor-energy decomposition error is 2.66×10^{-15}, and the maximum absolute pixel-attribution closure error is 3.83×10^{-6}. At $p_B = 0.9$ and $\lambda = 0$, optimized prediction risk is within 6.5×10^{-5} of the finite-space optimum. Stationarity, factor separation, sufficient background rank, and zero closure slack hold for the specified finite process. These checks support the controlled selection mechanism; the native control checkpoints are evaluated separately below.

B.2 Zero-action tests in the native architecture

The exact-whitening experiment makes the selection variables identifiable, but it leaves open whether the same qualitative effect appears in the native 192-dimensional architecture. We therefore next remove action variation while retaining the LeWorldModel encoder and predictor. This zero-action intervention is deliberately intermediate: it transfers the factor test to the deployed architecture without yet mixing temporal selection with action conditioning or closed-loop search.

Does persistence change native representation geometry? Matched TwoRoom branches start from the same native-objective epoch-10 checkpoint and receive identical agent-position sequences and texture marginals. Only background retention changes from $p_B = 0.1$ to 0.9, while position retention remains 0.5. After 500 matched warm-start updates, the high- and low-persistence branches have mean background/residual energies $(E_B, v) = (132.92, 24.53)$ and $(87.64, 23.63)$. Greater persistence therefore adds background-relative energy, but does not uniformly suppress the residual $v = E_S + E_{BS}$. This observation motivates a relative factor-preference statistic; it does not support interpreting the effect as a count of displaced dimensions or general information loss.

Does a specified target redirect that preference? We cross the same persistence intervention with the prediction-only objective (O) and joint state supervision (J), holding the image marginal, state stream, texture bank, initialization, optimizer, trainable backbone modules, and 500-update budget fixed. A high-persistence random-label branch preserves target scale while breaking image–target correspondence. An independent grid of 96 previously unused positions and 256 new textures measures

$$R = \log \frac{E_B + \varepsilon}{E_S + E_{BS} + \varepsilon}. \tag{49}$$

Paired contrast in R	Mean	95% t interval
O: high minus low	$+0.3681$	$[+0.3251, +0.4111]$
J: high minus low	$+0.1727$	$[+0.1020, +0.2433]$
Attenuation of the high–low shift	$+0.1954$	$[+0.0845, +0.3064]$
High persistence: J minus O	-0.2724	$[-0.4741, -0.0708]$
High persistence: true- minus random-label J	-0.1489	$[-0.2933, -0.0045]$

Table 4: Zero-action native-architecture factor test on an independent state–texture grid. Higher R denotes more background energy relative to state and interaction energy. Every displayed contrast has the shown sign in all four seeds. The grid supplies new measurements of the same trained models, not independent training replications.

The independent grid confirms relative reallocation rather than global latent rescaling: at high persistence, $J - O$ background energy is -29.329 and state energy is $+1.557$, with both directions shared by all four seeds. The stronger accessibility claim does not follow. The high–low change in the full residual v is -0.6074 with interval $[-1.9518, 0.7370]$, and the preregistered multilayer perceptron (MLP) attenuation holds in only three seeds. Moreover, the zero-action design contains no action–state coupling, and its random-label branch at high persistence does not by itself identify a semantic difference in the full persistence response. We therefore use this experiment as native-architecture evidence for relative factor selection and target-dependent reallocation, not as evidence of downstream control mediation.

B.3 ACTION-CONDITIONED FACTOR SELECTION

Zero action isolates factor selection, but cannot show that it survives the action-conditioned prediction problem used by the model. We therefore restore real dynamics and logged actions while retaining the same controlled persistence contrast. For each seed $1, 7, 42, 3072$, matched TwoRoom branches start from the same native-objective checkpoint and receive the same four-frame physical trajectories, three nonzero logged actions, batch order, and texture marginal. Low and high branches differ only in background persistence, $p_B = 0.1$ versus 0.9, while state persistence remains 0.5. The J branches supervise true agent coordinates; random-label branches permute complete coordinate trajectories across episodes, preserving label scale and temporal continuity while breaking image–state correspondence. Every branch runs 500 updates with the predictor conditioned on the logged actions. Both encoded- and predicted-state losses have weight 3. The predicted-state loss uses the same detached state-head weights as the unified case study: it updates the encoder and predictor, whereas only the encoded-state loss updates the state head. The action encoder remains fixed, so this experiment tests selection under action-conditioned prediction, not co-adaptation of the action representation. The four J branches were rerun with this routing; the O branches have no state loss and retain their original checkpoints. All factor, probe, and planning diagnostics below use the rerun checkpoints.

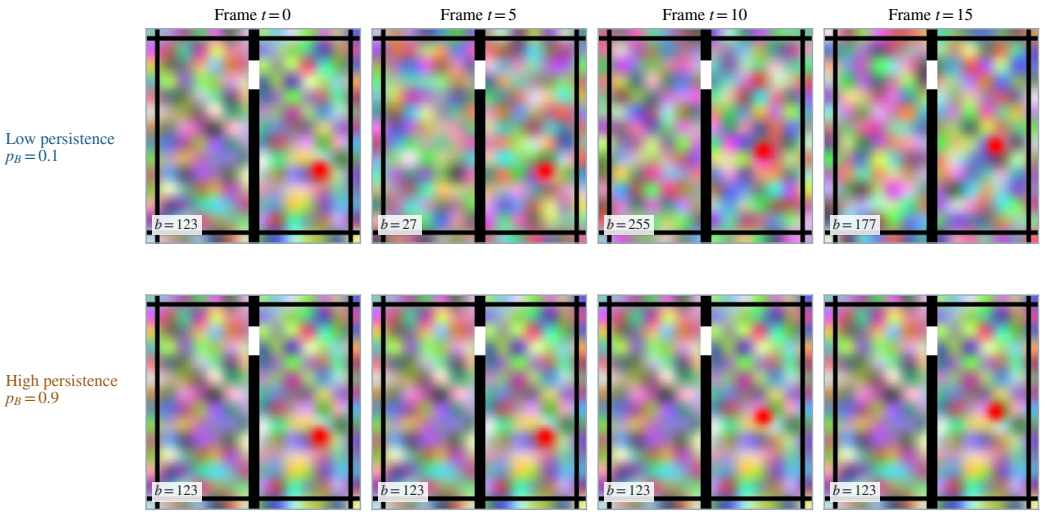


Figure 4: One paired TwoRoom training window from the action-conditioned factor intervention. Columns show observations at physical times $0, 5, 10, 15$ relative to the window start; both rows use the same agent positions and logged actions. The initial background identity and redraw proposals are paired. Only the probability of retaining the previous background changes: $p_B = 0.1$ above and $p_B = 0.9$ below. Labels b identify the rendered texture in this illustrative window; a redraw can return the same identity. Within each persistence setting, the three training regimes receive the corresponding images and differ in their supervision targets.

On 96 held-out positions crossed with all 256 backgrounds, define

$$\begin{aligned} R &= \log \frac{E_B + \varepsilon}{E_S + E_{BS} + \varepsilon}, & G_R &= R_{O,h} - R_{O,l}, \\ A_R &= G_R - (R_{J,h} - R_{J,l}), & S_R &= R_{J_{\mathrm{rand}},h} - R_{J,h}, \\ Q_R &= (R_{J_{\mathrm{rand}},h} - R_{J_{\mathrm{rand}},l}) - (R_{J,h} - R_{J,l}). \end{aligned} \tag{50}$$

The first three contrasts were specified before the random-label low-persistence branch was added; Q_R is post hoc.

Seed	G_R	A_R	S_R	Q_R
1	0.7731	0.5079	0.3114	-0.1279
7	0.5051	0.4053	0.1248	-0.0033
42	0.8905	0.6466	0.4058	-0.0836
3072	0.6884	0.4985	0.3228	-0.1008
Mean	0.7143	0.5146	0.2912	-0.0789
Paired 95% t interval	$[0.4562, 0.9723]$	$[0.3564, 0.6728]$	$[0.1025, 0.4799]$	$[-0.1642, 0.0064]$

Table 5: Seed-level action-conditioned factor contrasts. Positive G_R denotes stronger background preference under high persistence. The post-hoc Q_R interval includes zero and does not establish a difference in persistence response specific to true labels.

The positive G_R supports predictability-driven factor selection with real actions and physical trajectories. The random-label high–low response is $+0.1208$ $[+0.0669, +0.1746]$, compared with $G_R = +0.7143$ under O; random labels therefore also attenuate the response. Although A_R and S_R are positive, Q_R does not exclude zero. The result supports action-conditioned factor selection, while it does not establish attenuation specific to true labels or mediation of control outcomes.

Condition	Seed	E_B	E_S	E_{BS}	E_{total}	R	State probe
O_l	1	71.61	14.21	23.90	109.73	0.631	0.0205
O_l	7	81.43	12.37	16.43	110.23	1.039	0.0157
O_l	42	59.59	21.44	28.66	109.69	0.174	0.0201
O_l	3072	85.67	15.11	20.94	121.72	0.866	0.0225
O_h	1	121.11	6.69	23.06	150.87	1.404	0.0206
O_h	7	126.24	7.69	19.25	153.18	1.545	0.0173
O_h	42	110.11	12.05	25.95	148.11	1.064	0.0200
O_h	3072	130.72	8.84	18.80	158.35	1.554	0.0218
J_l	1	61.43	13.32	21.76	96.51	0.560	0.0199
J_l	7	72.23	10.20	14.41	96.84	1.077	0.0149
J_l	42	54.46	17.75	24.32	96.54	0.258	0.0194
J_l	3072	70.20	12.59	17.30	100.10	0.854	0.0228
J_h	1	75.41	11.14	21.89	108.43	0.826	0.0223
J_h	7	83.47	9.84	15.90	109.20	1.176	0.0161
J_h	42	69.95	17.16	25.18	112.29	0.502	0.0206
J_h	3072	83.50	11.79	17.62	112.91	1.044	0.0237
$J_{\text{rand},l}$	1	71.02	7.36	18.77	97.15	1.000	0.0193
$J_{\text{rand},l}$	7	76.29	8.23	14.63	99.15	1.205	0.0159
$J_{\text{rand},l}$	42	66.49	11.63	19.85	97.98	0.748	0.0173
$J_{\text{rand},l}$	3072	78.66	7.90	14.03	100.59	1.277	0.0203
$J_{\text{rand},h}$	1	84.58	7.09	20.04	111.71	1.137	0.0185
$J_{\text{rand},h}$	7	88.71	8.02	16.12	112.85	1.301	0.0167
$J_{\text{rand},h}$	42	83.74	12.44	21.34	117.52	0.908	0.0176
$J_{\text{rand},h}$	3072	92.38	8.36	15.20	115.94	1.366	0.0199

Table 6: Absolute held-out factor energies and observed-latent state readability after 500 updates. The state-probe column reports normalized root-mean-square error (NRMSE) on the test episodes from a newly fitted multilayer perceptron, using disjoint training, validation, and test episodes; lower is better. Subscripts l and h denote low and high background persistence, and rand denotes random labels. $E_{\text{total}} = E_B + E_S + E_{BS}$.

Compared with random labels, true labels increase E_S by 4.6825 $[1.6247, 7.7403]$ at low persistence and 3.5033 $[1.5281, 5.4786]$ at high persistence. The corresponding true-minus-random E_B differences are -8.5288 $[-13.8351, -3.2225]$ and -9.2722 $[-14.8469, -3.6974]$. These changes cannot be explained by a common rescaling of all energies. For each checkpoint, we remove the training state head, freeze the world model, and fit fresh ridge-regression and MLP probes from observed 192-dimensional latents to two-dimensional agent-position coordinates, using 180 training, 60 validation, and 60 test episodes with eight windows per episode. The probe training episodes come from the world-model training partition; test episodes come from its held-out test partition. In this 500-update true-versus-random comparison, the MLP probe does not resolve better observed-latent state readability for true labels: true-minus-random test error is $+0.0010$ $[-0.0015, 0.0035]$ at low persistence and $+0.0025$ $[-0.0009, 0.0059]$ at high persistence. The ridge-regression intervals likewise span zero. This contrast differs from the ten-epoch J_{task}-versus-O predicted-latent NRMSE comparison in Table 9. In fixed-batch gradient checks at steps $0, 50, 100, 250, 500$, the predicted-state

loss has zero state-head gradient and nonzero encoder and predictor gradients in all 16 rerun training runs. The true- and random-label loss and gradient trajectories differ, so this control does not hold optimization difficulty fixed.

We also score 300 candidate plans for each of the same 100 fixed initial simulator states and goals, and repeat native closed-loop evaluation using these checkpoints. Across four seeds, the mean true physical cost of the model-selected top 30 candidates is 47.46 for J_l, 48.31 for J_h, 46.43 for $J_{\mathrm{rand},l}$, and 48.16 for $J_{\mathrm{rand},h}$; lower is better. Native closed-loop success is 24.75%, 24.50%, 33.00%, and 27.00%, respectively, versus 28.25% and 31.75% for O_l and O_h. The true-supervision attenuation contrast in success is -3.75 percentage points with 95% interval $[-22.99, 15.49]$. All 24 evaluated runs pass replay checks comparing uninstrumented and recorded runs, as well as matched-input checks. These diagnostics do not show that the observed factor reallocation improves candidate selection or closed-loop control.

B.4 Unified state-target case study

Tasks, targets, and splits. The datasets are `ogbench/cube_single_expert`, `pusht_expert_train`, `tworoom`, and `dmc/reacher_random`. Full-state targets have dimensions 28, 7, 2, and 6, respectively. Episodes are partitioned into mutually exclusive 80% training, 10% validation, and 10% test partitions before window construction, and action and state normalizers use training episodes only. Success uses Cube block-position error ≤ 0.04 m; PushT joint pusher/block position error < 20 and shortest periodic block-angle error $< \pi/9$; TwoRoom position error < 16; and Reacher maximum raw joint-angle error < 0.05 rad.

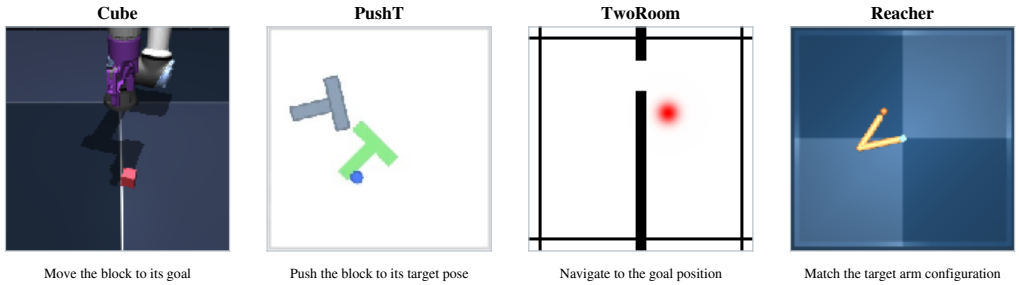


Figure 5: Representative stored observations from the four datasets in the unified case study. The text below each image describes the task goal; the images provide visual context for the environments and are not outcome comparisons.

Training and evaluation. The prediction-only (O), encoded-state-supervision (E), and joint-supervision (J) conditions use $(\beta_E, \beta_P) = (0,0), (3,0), (3,3)$ with the parameter routing in Equation (15). Within each task, all arms share data, initialization seed, architecture, optimizer, and ten-epoch budget. Images are 224×224, latent dimension is 192, batch size is 128, and AdamW uses learning rate 5×10^{-5} and weight decay 10^{-3}. Training uses three-frame history, mixed precision, and gradient clipping at one.

Each checkpoint is evaluated on the same 100 held-out initial simulator states and goals per seed, with matched complete states and initial-condition-specific random streams. The native cross-entropy method (CEM) planner uses 300 candidates, 30 updates, 30 elites, and five model steps of five primitive actions, with at most two planning calls. Deployment uses only the native latent goal-image score. All 48 training runs and 48 evaluation runs satisfy the frozen manifest, checkpoint, initial-state, candidate-stream, submitted-action, and trajectory-replay checks.

Native latent covariance audit. We froze all 48 accepted epoch-10 checkpoints and encoded one sampled frame from each of 768 distinct training episodes and 768 distinct held-out test episodes per task. The same sampled frames were used for all three arms and four seeds; the training image transform and the model's native `encode()['emb']` output were retained without post-hoc

whitening. Each sample covariance was centered within its partition and divided by 768. Table 7 reports four-seed means on test episodes. Training-episode estimates agree in direction: across all training conditions, $\mathrm{tr}(\hat{\Sigma})/192$ ranges from 0.967 to 1.036, $\|\hat{\Sigma} - I\|_F/\sqrt{192}$ from 1.195 to 2.231, and participation ratio from 33.8 to 79.0.

Task	Training regime	$\mathrm{tr}(\hat{\Sigma})/192$	$\|\hat{\Sigma} - I\|_F/\sqrt{192}$	Participation ratio
Cube	Prediction only	0.998	1.241	75.4
	Encoded-state supervision	0.985	1.518	56.9
	Joint state supervision	0.990	1.850	42.7
PushT	Prediction only	0.990	1.225	75.9
	Encoded-state supervision	0.983	1.529	56.2
	Joint state supervision	0.991	1.786	45.2
TwoRoom	Prediction only	1.014	1.825	45.3
	Encoded-state supervision	1.021	1.893	43.3
	Joint state supervision	1.028	2.211	34.2
Reacher	Prediction only	0.985	1.576	53.9
	Encoded-state supervision	0.987	1.609	52.5
	Joint state supervision	0.986	1.906	40.5

Table 7: Native latent covariance on held-out episodes, four-seed means. The participation ratio is $(\mathrm{tr}\,\hat{\Sigma})^2/\,\mathrm{tr}(\hat{\Sigma}^2)$; it is not algebraic rank or a count of semantic factors. All arms have latent dimension 192.

Finite sampling itself perturbs a white covariance estimate: 64 synthetic draws of 768 independent standard-normal vectors in 192 dimensions give $\|\hat{\Sigma} - I\|_F/\sqrt{192}$ between 0.495 and 0.508 and participation ratio between 152.6 and 154.0. This reference is a scale check, not a formal test for the image distribution. The observed deviation is much larger and remains similar across train and test episodes. A trace near 192 does not imply isotropy; the participation ratio alone does not establish information collapse or quantify task information. This audit does not measure the theorem's factor separation, predictive gap, realizability, or closure cost. Post-hoc whitening would not make the original training objective satisfy population whitening.

Task	Training condition	1	7	42	3072
Cube	O	66	68	65	67
Cube	E	75	75	73	77
Cube	J	82	74	74	71
PushT	O	83	84	87	81
PushT	E	86	90	87	87
PushT	J	90	92	95	92
TwoRoom	O	86	82	81	78
TwoRoom	E	85	88	87	87
TwoRoom	J	89	92	91	92
Reacher	O	92	80	86	84
Reacher	E	90	88	84	81
Reacher	J	97	92	97	98

Table 8: Unified-protocol success percentage under the native cross-entropy-method planner for every training seed. O, E, and J denote prediction-only training, encoded-state supervision, and joint supervision.

The paired $E - O$ intervals are $[6.45, 10.55]$, $[-0.82, 8.32]$, $[-1.75, 11.75]$, and $[-8.01, 8.51]$ percentage points in task order. The paired $J - E$ intervals are $[-8.31, 8.81]$, $[0.77, 8.73]$, $[3.45, 5.05]$, and $[0.94, 19.56]$; the paired $J - O$ intervals are $[0.39, 17.11]$, $[5.74, 11.26]$, $[1.97, 16.53]$, and $[4.34, 16.66]$. These are conditional intervention contrasts, not evidence of universal necessity or synergy.

Success-coordinate target sufficiency. An auxiliary arm J_{task} changes only the supervised columns: Cube block position, PushT pusher and block positions plus raw block angle, TwoRoom

agent position, and Reacher's two raw joint angles. Reacher uses raw angles rather than sine–cosine features; PushT uses raw angle for training but the shortest periodic difference for success. J_{task} otherwise inherits the unified training and native-deployment protocol. All 16 training runs, 16 closed-loop evaluations, and 48 probe evaluations pass the declared checks. Post-hoc ridge-regression probes are fitted on observed latents from disjoint training episodes and evaluated on predicted latents from common logged-action test windows; they are neither the training state head nor the deployed score.

Task	Δ NRMSE H1	Δ NRMSE H5	$J_{\text{task}} - O$ success	$J_{\text{task}} - J_{\text{full}}$ success
Cube	-0.0928 $[-0.1099, -0.0757]$	-0.0838 $[-0.0955, -0.0722]$	$+8.25$ $[+2.10, +14.40]$	-0.50 $[-13.78, +12.78]$
PushT	-0.2645 $[-0.3273, -0.2017]$	-0.1720 $[-0.2211, -0.1230]$	$+8.50$ $[+4.71, +12.29]$	0.00 $[-3.44, +3.44]$
TwoRoom	-0.0728 $[-0.0838, -0.0619]$	-0.0209 $[-0.0430, +0.0011]$	$+9.75$ $[+4.66, +14.84]$	$+0.50$ $[-5.09, +6.09]$
Reacher	-0.0254 $[-0.0299, -0.0209]$	-0.0214 $[-0.0237, -0.0191]$	-2.75 $[-12.59, +7.09]$	-13.25 $[-20.64, -5.86]$

Table 9: Success-coordinate target check. NRMSE denotes normalized root-mean-square error; H1 and H5 denote one- and five-step prediction horizons. Probe entries are paired differences between task-coordinate supervision (J_{task}) and prediction-only training (O); lower is better. J_{full} denotes full-state supervision. Success entries are percentage-point differences under the unchanged native planner. Brackets give paired 95% t intervals over four seeds.

Target-coordinate probe error decreases for every task, horizon, and seed, although the TwoRoom five-step interval crosses zero. J_{task} retains supported control gains on Cube, PushT, and TwoRoom, but not on Reacher, where success is 82.75% versus 85.50% for O and 96.00% for J_{full}. Thus success-coordinate readability may be insufficient for native control. This result neither shows that task-specific supervision is generally harmful nor contradicts Proposition 3: semantic coordinate overlap does not certify $S_Y = aW$, and native success is not the score-correlation quantity in Equation (12).

B.5 Unified ablation measurements

Common-window state probes. We freeze the unified epoch-10 $O/E/J$ checkpoints and use the same logged-action windows for every arm of a task. The 180 training, 60 validation, and 60 test episodes are disjoint; each contributes eight windows, giving 2,400 windows and 480 test windows per task. A separate affine ridge head is fitted from observed latents to the complete simulator state: Cube has 28 coordinates, PushT 7, TwoRoom 2, and Reacher 6. Ridge strength is chosen only on validation episodes. At test time, the same head is applied to encoded future observations and to five-step action-conditioned predicted latents. NRMSE divides each coordinate's residual by its training-set standard deviation, then takes the root mean square over windows and coordinates. The three error components in Table 10 are not additive.

Task	Arm	Total H1	Encoding H5	Transport H5	Total H5
Cube	O	0.7949	0.7899	0.0639	0.7877
	E	0.5617	0.5906	0.2368	0.5403
	J	0.5329	0.5842	0.3375	0.5620
PushT	O	0.6255	0.5970	0.1726	0.5943
	E	0.2304	0.4012	0.3546	0.2599
	J	0.1362	0.4100	0.4456	0.2064
TwoRoom	O	0.1091	0.0700	0.1705	0.1680
	E	0.0468	0.0055	0.1491	0.1491
	J	0.0380	0.0057	0.1453	0.1453
Reacher	O	0.7767	0.7755	0.1120	0.7752
	E	0.7708	0.7661	0.1030	0.7679
	J	0.6577	0.7653	0.3233	0.6668

Table 10: Complete-state post-hoc NRMSE on common held-out logged-action windows, four-seed means. Encoding compares the fitted head on future observed latents with the state; transport compares its outputs on predicted and observed future latents; total compares the predicted-latent output with the state. Lower is better. The head is diagnostic and is never used by the native planner.

The same matched protocol also probes success-related coordinates. The E results are newly evaluated; O and full-state J reuse the identical-window probes from the target-sufficiency study.

Table 11 reports absolute values, complementing paired differences in Table 9. The task-coordinate and complete-state NRMSEs average over different coordinate sets.

Task	O H1	E H1	J H1	O H5	E H5	J H5
Cube	0.1107	0.0446	0.0541	0.1108	0.0498	0.0759
PushT	0.3940	0.1429	0.1576	0.3853	0.2201	0.2431
TwoRoom	0.1091	0.0468	0.0380	0.1680	0.1491	0.1453
Reacher	0.6302	0.6244	0.6363	0.6383	0.6265	0.6572

Table 11: Post-hoc total NRMSE for success-related coordinates on the same disjoint logged-action windows, four-seed means. These heads are fitted after world-model training and are not the auxiliary heads used during training.

Committed elite-mean actions in both planning stages. The native CEM uses 300 candidates, 30 updates, and the mean of its final top 30 candidates as the committed 25-action plan. The recorded mean-action consistency error is zero. The first stage runs on all 100 test roots. Only roots not yet successful enter the second solve, at states produced by their own first plan. Table 12 gives absolute physical costs and success rates from executed trajectories, not individual-candidate replay. Second-stage cost is the unweighted mean of four seed-level means on the arm-specific active roots; its active count varies by arm and it cannot be read as a paired causal contrast.

Task	Arm	Cost 1	Cost 2 active	Active 2	Success 25 (%)	Rescue (/100)	Success 50 (%)
Cube	O	2.001	4.795	35.50	64.50	2.00	66.50
	E	1.558	4.911	26.25	73.75	1.25	75.00
	J	1.370	2.633	37.25	62.75	12.50	75.25
PushT	O	1.078	3.243	23.75	76.25	7.50	83.75
	E	1.035	2.485	16.75	83.25	4.25	87.50
	J	0.893	2.211	10.50	89.50	2.75	92.25
TwoRoom	O	1.730	4.621	22.75	77.25	4.50	81.75
	E	1.416	4.034	17.50	82.50	4.25	86.75
	J	1.247	3.939	11.75	88.25	2.75	91.00
Reacher	O	1.817	1.110	55.50	44.50	41.00	85.50
	E	1.821	1.129	54.25	45.75	40.00	85.75
	J	2.117	0.859	58.25	41.75	54.25	96.00

Table 12: Native closed-loop execution of committed elite-mean plans, four-seed means. Cost 1 uses all roots; cost 2 uses each arm's first-stage failures. Cube and PushT cost is the best normalized target distance reached within that stage; TwoRoom and Reacher cost is the stage endpoint distance. Success is the environment's termination event. The difference between first- and second-stage costs also reflects the change in evaluated roots.

Table 13 conditions instead on each seed's initial roots where both O and J fail the first stage. The second-stage physical states still differ because the first actions differ. Thus the contrasts retain paired initial-root identity without isolating a planner-only effect.

Task	$J - O$ second-stage cost	$J - O$ second-stage success (pp)
Cube	-2.179 $[-2.654, -1.705]$	$+24.95$ $[+11.86, +38.03]$
PushT	$+0.636$ $[-0.532, +1.803]$	-9.17 $[-26.15, +7.82]$
TwoRoom	-1.340 $[-2.464, -0.216]$	$+8.06$ $[-8.74, +24.85]$
Reacher	-0.253 $[-0.416, -0.089]$	$+19.10$ $[+6.89, +31.30]$

Table 13: Paired common-first-stage-failure contrasts under the unified protocol. Brackets give four-seed 95% t intervals. Lower cost and higher success favor J. Shared initial roots do not imply shared states at the second solve.

Changing the downstream candidate score. As a supplementary intervention, we keep each unified-protocol world-model checkpoint and closed-loop setup fixed and change only the CEM candidate score. Let $D_L(U)$ be the native terminal latent cost for an action sequence U, $\hat{z}_H(U)$ its predicted terminal latent, and z_g the encoded goal image. For each checkpoint, an affine task-

coordinate state head with weight matrix W defines

$$D_S(U) = \mathrm{mean}_j \left[\left(\frac{(W(\hat{z}_H(U) - z_g))_j}{\sigma_j} \right)^2 \right], \qquad D_{\mathrm{mix}}(U) = \frac{D_L(U)}{m_L} + \frac{D_S(U)}{m_S}.$$

Here σ_j is the training-set standard deviation of state coordinate j; the affine intercept cancels in the difference. The O and J arms use their existing, identically specified post-hoc task-coordinate heads, and we fit the same head for E. Both normalized terms have fixed unit weight. For each checkpoint, m_L and m_S are the respective medians of positive finite costs on 300 initial CEM candidates from each of eight independent validation episode roots. We freeze these scales before evaluation; validation and test roots do not overlap. The planner still observes image latents rather than simulator states.

We compare native and mixed scoring on the same 100 held-out initial roots per task, arm, and seed, using four seeds. Checkpoints, goals, first-round candidate streams, CEM settings (300 candidates, 30 updates, 30 elites), 25-action commitments, and the limit of two planning rounds are matched. We audited root states, images, candidate hashes, and success events across all 48 paired cells. Table 14 reports four-seed mean success rates and paired mixed-minus-native differences; rescue/loss counts pool the 400 paired roots per row. Later candidates and physical states may diverge after different first-round actions.

Task	Arm	Native (%)	Mixed (%)	Δ success (pp) [95% interval]	Rescued/Lost
Cube	O	66.50	75.75	$+9.25$ $[+5.27, +13.23]$	51/14
	E	75.00	78.00	$+3.00$ $[+0.75, +5.25]$	38/26
	J	75.25	76.75	$+1.50$ $[-5.56, +8.56]$	30/24
PushT	O	83.75	82.75	-1.00 $[-4.90, +2.90]$	33/37
	E	87.50	85.50	-2.00 $[-7.03, +3.03]$	20/28
	J	92.25	90.50	-1.75 $[-3.27, -0.23]$	8/15
TwoRoom	O	81.75	99.25	$+17.50$ $[+11.62, +23.38]$	71/1
	E	86.75	99.00	$+12.25$ $[+10.25, +14.25]$	49/0
	J	91.00	99.25	$+8.25$ $[+6.25, +10.25]$	33/0
Reacher	O	85.50	79.75	-5.75 $[-15.59, +4.09]$	42/65
	E	85.75	84.00	-1.75 $[-15.89, +12.39]$	44/51
	J	96.00	94.50	-1.50 $[-3.55, +0.55]$	13/19

Table 14: Supplementary downstream-score intervention under the unified closed-loop protocol. Each difference is computed on paired roots within a seed; brackets are 95% t intervals over four seeds, without a multiple-comparison adjustment. Rescued/lost counts are initial roots that succeed only with mixed/native scoring, respectively.

Mixed scoring improves success in all three TwoRoom arms and in Cube O and E, but does not show a consistent benefit across tasks; PushT J decreases in this comparison. This intervention establishes that changing the downstream candidate score can change closed-loop outcomes. It does not identify the source of the native $O/E/J$ differences or provide evidence that the state head mediates J's native gains.

C Tracing the Downstream Control Path

The control comparison shows that target supervision changes success, but success alone does not show where the change enters planning. We examine when the difference appears, how the planner selects actions, what those actions achieve, and what the model observes before planning again. These diagnostics were collected separately. The checkpoints and initial conditions are matched within each test, but the results cannot be combined into a causal decomposition of the unified control comparison. In particular, the target-supervised and reference checkpoints used in the later diagnostics are not interchangeable with the three arms of that comparison.

C.1 Timing and reached states

A stage monitor compares prediction-only training with joint supervision after the first 25 actions and after a possible second plan. On TwoRoom, the joint model's 11.75 percentage point lead is already

present after the first plan and remains the same at the end. Cube moves from a 2.25 point deficit after the first plan to a 9.25 point gain after replanning. PushT's early 6.75 point lead narrows to 1.75 points, whereas Reacher's 3.25 point lead grows to 9 points. The tasks therefore differ in when the final success gap emerges.

We also examine initial conditions on which both models fail after the first plan. On Cube, joint supervision leaves the system closer to the goal but moving faster. On Reacher, it leaves the arm farther from the goal but usually moving more slowly; more of these cases succeed after the second plan. These reached states describe the trajectories produced by each model. Because the first actions differ, even the shared initial conditions do not provide a common state at the second planning call. Conditioning on both models' first-plan failures is also a choice made after those actions, so these observations do not identify a state variable that causes the later gain.

C.2 Candidate selection and committed actions

A better planning score would rank physically better actions more highly when the candidate actions are held fixed. The fixed-candidate test shows a resolved ranking change on Cube and PushT, but the directions are opposite; the other two tasks show no resolved change. A second test fixes the initial state, goal, and candidate pool before comparing scores. It likewise finds no consistent advantage for the target-supervised model. Its Reacher choices have higher physical cost across the tested contexts, while the Cube result depends on context. These tests do not support improved ranking of a common set of actions as a shared explanation for the control gains.

The planner's search also changes which actions enter its final candidate pool. To examine the last selection within that pool, we compare the action formed from the 30 highest-scored candidates with one formed from 30 randomly selected candidates in the same pool. The difference between these selection rules is not resolved for any task; this is an absence of detected interaction, not evidence that the rules are equivalent. In a separate Cube replay of the second plan, individual selected endpoints, their average physical endpoint, and the executed mean action all improve under the target-supervised model. Random selection from its final pool improves similarly. The physical improvement is observable, but the replay does not isolate the final ranking decision, especially because the models reach different states before that plan.

Checks on the learned representation do not close this gap. In action-conditioned TwoRoom, linear and nonlinear probes give inconsistent evidence that the change in latent factor energies improves state-coordinate prediction. A short target-label intervention also fails to improve the physical cost of selected candidates and worsens it on Reacher. This short training intervention does not replace the full-training control comparison.

C.3 Physical outcomes and replanning

After committed actions are executed, physical target distance at 50 actions favors the target-supervised model on Cube and TwoRoom. The corresponding comparisons on PushT and Reacher do not resolve an improvement, so endpoint distance alone does not explain the Reacher success gain. Reacher instead shows a difference after the next observation becomes available. Updating the observation before the second plan improves later success in both the target-supervised and reference models. Swapping the velocities of the reached states does not reproduce a reliable success change. The lower velocity observed in many jointly supervised Reacher trajectories is therefore insufficient on its own to explain their later successes.

The measured effects occur at different stages and through different physical outcomes across tasks. The probe, fixed-candidate, final-pool, endpoint, and velocity tests each constrain a possible explanation, but none establishes a shared mediator of the four-task success gains. They also do not rule out an unmeasured common mediator.