A LINEAR MAP GOES A LONG WAY: LEARNED COSTS FOR PLANNING WITH FROZEN WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Planning with a visual world model requires a cost that compares predicted outcomes with a goal. Learned costs outperform raw latent distance, but it remains unclear how much of this benefit requires a flexible scoring network. We keep the encoder and dynamics fixed and train three costs to predict the elapsed time between logged observations under identical conditions: a linear map, a shared nonlinear map, and the joint scorer of prior work. In TwoRoom navigation, the linear cost raises planning success from 18% to 96%, and neither flexible cost adds a clear gain. The result holds for a second family of world models, and across rooms the linear cost also outperforms a cost built from decoded positions. Whitening the latent space or training the same map on shuffled targets recovers almost none of the improvement, indicating that the temporal targets are essential to it. In PushT block pushing, the flexible costs fit elapsed time better, even on episodes excluded from cost training, yet the joint scorer plans worse than the linear cost. The linear cost raises success from 36% to 44%, while the joint scorer reaches only 29%, and this gap persists after tuning each family and after symmetrizing the joint scorer. A comparably trained linear cost is therefore the baseline that learned planning costs should beat, and temporal loss alone is not a reliable criterion for choosing among them.

1 INTRODUCTION

Visual world models support planning from images by predicting the outcomes of candidate actions (Maes et al., 2026b; Zhou et al., 2025). Choosing among those predictions also requires a cost that compares each outcome with the goal. Squared latent distance is a convenient choice, but predictive accuracy and usefulness for control are distinct objectives (Lambert et al., 2020). Learned costs can improve planning while leaving the predictive model fixed (Li et al., 2026a; Singh, 2026). We ask how much of that improvement requires a flexible scoring network.

Raw distance weights latent directions according to the geometry already present in the representation, whether or not those directions are useful for reaching a goal. In LeWorldModel (LeWM) (Maes et al., 2026b), for instance, the rowspace of a linear position probe accounts for less than 1% of the squared latent distance between predicted terminal states and the goal, yet carries most of the signal that separates good candidate plans from poor ones (Li et al., 2026a). A learned cost can change this weighting without retraining the predictive model. A linear map applies one matrix to both representations before measuring their distance, while a shared nonlinear map allows the geometry to vary with location. A joint scorer processes the predicted outcome and the goal together and need not define a distance at all. Comparing these constructions measures how much additional scoring flexibility helps beyond a trained linear geometry. Figure 1 illustrates the linear case in two dimensions, where a map learned from elapsed time down-weights a direction that changes without bringing the agent closer to the goal.

We train all three costs to predict the elapsed time between logged observations, with the encoder and dynamics fixed and with the same training pairs, objective, budget and model selection rule. These shared conditions remove the most direct sources of confounding, although architecture, parameterization and optimization remain intertwined. In PushT, we additionally give every family the same optimizer search, selected by planning success on separate development tasks.

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.

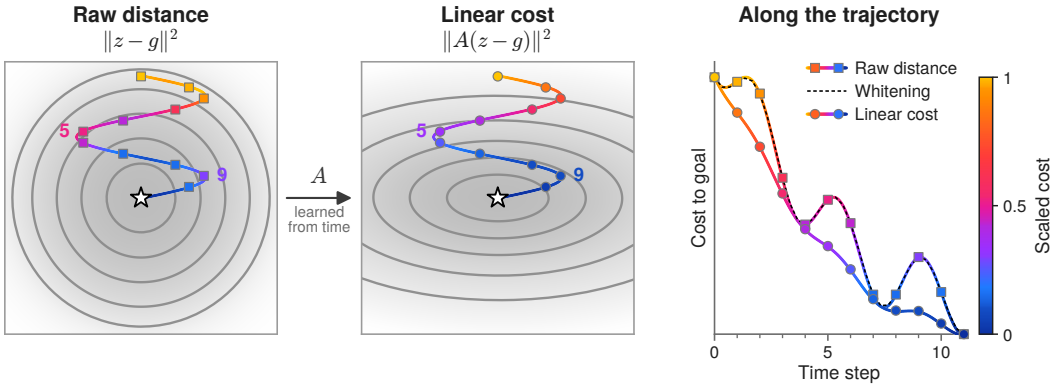

Figure 1: A linear map learned from elapsed time turns a misleading distance into a cost that falls as the agent approaches the goal. In this two dimensional latent space, one direction tracks progress toward the goal and the other changes without bringing the agent closer. The learned map A down-weights the second direction and stretches the circular level sets of raw distance into ellipses. Along the plotted trajectory, raw and whitened distance rise with each sideways swing, as at steps 5 and 9, while the linear cost decreases at every step.

Our main finding is that, in navigation, a linear reweighting of the frozen latent space suffices to recover the geometry that planning requires. A single matrix trained on elapsed time raises TwoRoom success with LeWM from 18% to 96%, and neither the nonlinear cost nor the joint scorer of prior work shows a clear advantage over it. The gain carries over to two separately trained models from PLDM (Planning with a Latent Dynamics Model) (Sobal et al., 2025). Covariance whitening and training on shuffled targets recover almost none of this gain, indicating that the temporal targets are essential to it. Across rooms, the linear cost also outperforms a cost built from a linear position probe, and over a broader set of goals it more often orders observations by their distance through the doorway when that order disagrees with straight line proximity. Figure 2 summarizes the main comparisons.

A second finding is that temporal fit and planning utility can come apart. In PushT, the linear cost improves over raw distance while the joint scorer falls below it, even though both flexible costs predict elapsed time more accurately than the linear cost, including on episodes excluded from cost training and configuration selection. The joint deficit persists when every cost starts each task from the same state, after an equal optimizer search, and after averaging the joint scorer over both endpoint orders. A comparably trained linear cost is therefore the baseline that learned planning costs should beat, and temporal loss alone is not a reliable criterion for choosing among them.

2 Related Work

Planning with visual world models. PlaNet (Hafner et al., 2019) plans from pixels through learned latent dynamics. DINO-WM (Zhou et al., 2025) predicts over frozen DINOv2 features, whereas PLDM (Sobal et al., 2025) and LeWorldModel (Maes et al., 2026b) learn representations jointly with dynamics. LeWM uses the SIGReg regularizer introduced in LeJEPA (Balestriero & LeCun, 2025). Methods including SD-JEPA (Thil et al., 2026), RC-aux (Li et al., 2026b) and Temporal Straightening (Wang et al., 2026) modify representation or dynamics training. We study a different intervention: changing the planning cost while keeping the predictive representation fixed.

Learning geometric distances. Mahalanobis metric learning expresses distance through a learned linear transformation (Weinberger & Saul, 2009; Jain et al., 2012; Kulis, 2013). Contrastive supervision extends this construction to shared nonlinear embeddings (Hadsell et al., 2006). VIP (Ma et al., 2023) learns video representations whose embedding distances define goal-conditioned values and support visual reward construction. These approaches motivate our geometric costs. We apply the maps to an already trained predictive representation and compare their planning utility.

Temporal distances for goal reaching. Trajectory data connect learned distances to behavior. Semi-parametric topological memory (SPTM) (Savinov et al., 2018) classifies temporal proximity to support graph-based navigation. Dynamical Distance Learning (Hartikainen et al., 2020) estimates policy-dependent temporal distance, while LEXA (Mendonca et al., 2021) learns temporal distances from imagined rollouts for goal reaching. Contrastive RL (Eysenbach et al., 2022) and Hilbert foundation policies (Park et al., 2024) connect learned embeddings with goal-conditioned values. Quasimetric methods (Wang & Isola, 2022; Wang et al., 2023) and Metric Residual Networks (Liu et al., 2023) introduce directed-distance structure, while contrastive successor features provide metric structure for temporal distances (Myers et al., 2024). We compare cost constructions under symmetric temporal-gap regression on fixed representations.

Learning costs for frozen world models. The closest work replaces the cost used to score frozen predictions. Trajectory Reachability Metrics (TRM) (Li et al., 2026a) learns a joint endpoint cost from logged temporal separations and improves planning with frozen LeWM and PLDM models. Singh (2026) reports saturation and reversal of squared latent distance at large separations in LeWM, and studies position-based and temporal alternatives. Our linear map induces a full positive semidefinite quadratic cost, including interactions between latent coordinates. We compare it with a shared nonlinear map and the TRM endpoint architecture under a common temporal training procedure. The contribution is the measurement of additional scoring flexibility beyond a comparably trained linear baseline.

3 Comparing Planning Costs on Frozen World Models

3.1 Planning with Frozen Predictions

We learn the planning cost while keeping the predictive model fixed. Let E denote the image encoder and projector, with $z_t = E(o_t)$ and $g = E(o_g)$ in $\mathbb{R}^d$. From the observation history and candidate action sequence $\mathbf{a}_t$, the frozen dynamics predict a terminal representation $\hat{z}_{t+H}(\mathbf{a}_t)$. The planner scores the sequence by

$$J_c(\mathbf{a}_t) = c(\hat{z}_{t+H}(\mathbf{a}_t), g)\,. \tag{1}$$

The cross-entropy method (CEM) (Rubinstein, 1999) updates its proposals using the lowest-cost candidates. The controller executes the final elite-mean plan as described in Section 4.2. Learned transformations act only in the cost computation. Dynamics remain in the original latent coordinates, and all parameters stay fixed during evaluation.

3.2 Linear, Nonlinear and Joint Costs

We write normalized raw distance as $c_I(z, g) = \|z - g\|_2^2/d$. The native PushT evaluator uses the unnormalized sum $d\, c_I(z, g)$. A shared map $f : \mathbb{R}^d \to \mathbb{R}^d$ transforms the two representations separately before comparing them:

$$c_f(z, g) = \frac{1}{d}\|f(z) - f(g)\|_2^2. \tag{2}$$

The direct linear cost (D), nonlinear cost (N), and factorized linear control (F) use

$$\begin{aligned} f_D(z) &= Az, \\ f_N(z) &= W_2\,\mathrm{GELU}(W_1 z + b_1), \\ f_F(z) &= W_2(W_1 z + b_1). \end{aligned} \tag{3}$$

The factorized control removes the nonlinear activation and is evaluated on the second navigation task set. Its bias cancels between endpoints. Both linear constructions therefore induce global positive semidefinite quadratic costs, while the nonlinear map allows local weighting to vary. These costs are symmetric and zero at identical representations. The squared-distance form does not guarantee the triangle inequality.

Joint scorer. The joint scorer (J) uses TRM's endpoint architecture (Li et al., 2026a):

$$\psi(z, g) = [z, g, z - g, |z - g|], \qquad c_J(z, g) = s\, h_\phi(\psi(z, g)). \tag{4}$$

Here $|z - g|$ is the elementwise absolute difference, and h_ϕ has two SiLU hidden layers and a Softplus output. It processes the ordered endpoints jointly without enforcing symmetry. A separate PushT comparison evaluates $c_{J,\mathrm{sym}}(z, g) = [c_J(z, g) + c_J(g, z)]/2$ using the original joint heads without retraining.

Decoded-position reference. For navigation, an affine decoder $\widehat{x} = Bz + b$ defines the cost $c_{\mathrm{XY}}(z, g) = \|B(z - g)\|_2^2$, denoted P_{XY}. The intercept cancels. The decoder is trained with position labels and then frozen, providing a separate reference for the temporally trained costs.

Linear controls. A covariance-whitening cost uses

$$c_W(z, g) = \frac{1}{d}(z - g)^\top (\Sigma + \lambda I)^{-1}(z - g), \qquad \lambda = 10^{-3}\frac{\mathrm{tr}(\Sigma)}{d}, \tag{5}$$

where Σ is estimated from unique training-pair endpoints. A second control trains the direct linear map after one random permutation of the temporal training targets, while using real validation targets to select the epoch. Appendix A.5 gives both procedures.

3.3 Common Training and Model Selection

The temporal costs predict the separation of two observations within a logged episode: $y_{ij} = |t_i - t_j|/s$, with $s = 224$ for the main TwoRoom and PushT comparisons. The target measures separation along the logged trajectory rather than minimum travel time. Its symmetry leaves directed reachability outside the comparison. For training pairs $\mathcal{P}$, we minimize

$$\mathcal{L}_{\mathrm{reg}} = \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \ell_\beta(\widetilde{c}(z_i, z_j) - y_{ij}), \qquad \beta = 1, \tag{6}$$

where ℓ_β is Smooth-L1 loss. For shared maps, $\widetilde{c} = c_f$: squared mapped distance is fitted directly to normalized temporal separation. For the joint scorer, $\widetilde{c} = h_\phi \circ \psi$, with the factor s applied only at deployment. Training uses two encoded observations, whereas planning compares a predicted endpoint with an encoded goal.

Training and validation pairs come from the same pool of source episodes, so validation loss measures fit within that pool. Both endpoint orders are sampled with equal probability. Episodes used for planning tasks are excluded from both. Appendix A.2 gives the full sampler.

Within each comparison, the temporal costs share training pairs, minibatch order, objective, optimizer settings and budget. The direct map starts at $A = W_2W_1$, the product of the nonlinear map's initial weights, and the factorized control copies those weights, so the two linear constructions share their initial distance function. The joint scorer has a separate initialization. Every run completes the full training budget. We select the earliest epoch with minimum mean loss over all validation pairs.

The PushT tuning study gives each family the same six learning-rate and weight-decay configurations. Development planning success selects one configuration per family. Temporal pair-validation loss still selects the epoch within each run. This tests a finite search rather than individually optimized architectural ceilings. Appendix A gives dimensions, sampling and training details, and Appendix A.4 examines an alternative training readout for the linear cost.

4 Experimental Setup

4.1 Environments, Models and Task Sets

Navigation. TwoRoom (Sobal et al., 2025) connects two rooms through a doorway. We evaluate three frozen LeWorldModel models. The main navigation evaluation uses 200 tasks, split equally between goals in the start room and goals in the other room, drawn from 1,000 episodes excluded from cost training. Tasks across rooms require passing through the doorway and place the goal region inside the destination room. The temporal costs are trained without these episodes, each model receives its own position probe, and the tasks and analysis were fixed before any planning outcome. Each temporal cost has three training runs per model and is evaluated with three planner seeds. A second set of 200 tasks, almost all within one room, adds the factorized control and two PLDM models trained without its evaluation episodes.

Block pushing. PushT uses one released LeWM model and 200 evaluation episodes excluded from cost training. Each temporal cost has three training runs and two planner seeds. Because the environment's reset is unseeded, a follow-up evaluation fixes one starting state per task and shares it across costs. For each cost family, planning success on 100 separate development tasks selects one of six learning-rate and weight-decay configurations, and we call the resulting cost the tuned cost. A later diagnostic measures temporal loss on episodes excluded from cost training and configuration selection. The linear controls use the balanced navigation tasks and the fixed-state PushT tasks. Whitening is estimated from the cost-training encodings, and the shuffled-target and comparison linear costs each have one training run per model. Appendix B gives the full coverage of every study.

4.2 Planning and Statistical Evaluation

We use `stable-worldmodel` (Maes et al., 2026a) with each model's native observation history and action representation, and identical planner settings across costs. CEM uses 300 candidates, 30 iterations and 30 elites. Each plan spans 25 actions and is executed in full before replanning, within a 50-action budget. TwoRoom counts a success when position error is strictly below 16 pixels at any evaluated step, and the rollout continues after a hit. PushT stops at the first step with combined agent and block position-error norm below 20 pixels and symmetry-adjusted angular error below 20 degrees.

We average training runs and planner seeds within each task, weight tasks and the three LeWM navigation models equally, and report the PLDM models separately. Differences are in percentage points (points). Intervals are 95% paired bootstrap intervals over tasks, or over source episodes for losses, unadjusted for multiple comparisons. Appendix B.4 gives details and the primary comparison of each study.

5 Results

5.1 A Linear Cost Captures Most of the Navigation Improvement

On balanced TwoRoom tasks, the linear cost raises success from 18% with raw distance to 96% (Table 1). Neither the nonlinear cost nor the joint scorer has a clear overall advantage over it. The joint scorer adds a small gain for goals in the start room, while the comparison across rooms is inconclusive. Because success approaches its ceiling, an exploratory analysis also examines the trajectories. The joint scorer has a later first success within the action budget, with failures counted at the end of the budget, and ends about 2 pixels farther from the goal, with no clear difference in final success (Appendix C.1).

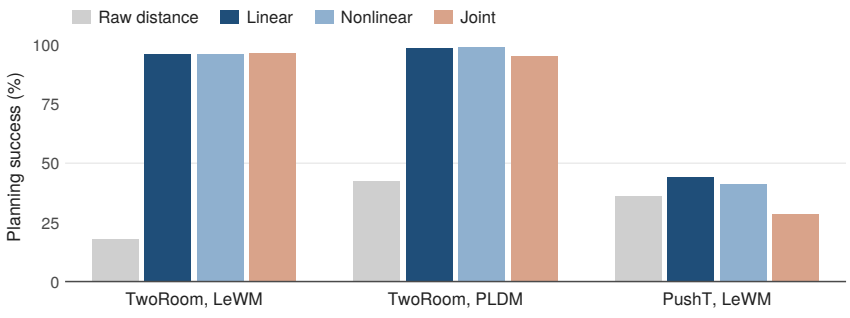

Figure 2: A trained linear cost captures most of the improvement over raw distance, and the more flexible costs add no consistent gain. In PushT, where many tasks remain unsolved, the joint scorer falls well below the linear cost. Bars show success on 200 tasks per setting, averaged over training runs and planner seeds (Table 1).

The linear improvement repeats with PLDM on the second task set. Both separately trained models exceed 98% success with the linear cost, against about 42% with raw distance. The nonlinear and joint costs were evaluated on one of these models and show no clear advantage there (Appendix D).

Table 1: The linear cost captures most of the navigation improvement and improves PushT planning. Each setting uses 200 tasks. Balanced TwoRoom averages three LeWM models with three training runs per temporal cost. PushT uses one model with three training runs, and the primary PLDM model uses one training run per cost. PushT rates use the original evaluation. The lower panel reports selected comparisons in points with marginal, unadjusted 95% paired intervals. The tuned PushT comparison starts every cost from the same state for each task.

Setting	Planner seeds	Raw	Linear	Nonlinear	Joint	Position
Balanced TwoRoom, LeWM	3	18.1	96.3	96.1	96.6	91.4
TwoRoom, PLDM primary model	1	42.5	98.5	99.0	95.5	–
PushT, LeWM	2	36.3	44.3	41.1	28.6	–

Selected comparisons

Setting	Comparison	Difference	95% interval
Balanced TwoRoom	Joint − linear	$+0.28$	$[-0.69, 1.26]$
Balanced TwoRoom, across rooms	Linear − position	$+10.3$	$[5.9, 15.0]$
TwoRoom, PLDM primary model	Linear − raw	$+56.0$	$[49.0, 63.0]$
PushT	Joint − linear	-15.67	$[-20.00, -11.50]$
PushT, tuned, same starts	Joint − linear	-12.33	$[-16.92, -7.92]$

On the second task set, with goals almost entirely in the start room, flexibility helps modestly. With a single planner seed, the nonlinear cost exceeds both linear constructions by one to two points. The joint scorer also gains about one point at that seed, but its advantage is no longer clear once three planner seeds are averaged (Appendix C).

Temporal training outperforms the linear controls. Whitening the latent space by its covariance reweights directions without any temporal signal. Training the same linear map on shuffled targets keeps the optimization but breaks the link between each pair and its elapsed time. On the balanced tasks, with one training run per learned linear cost and one planner seed, the trained linear cost reaches 97% success, against 20% for whitening and 11% for the shuffled control, which falls below raw distance. Whitening helps within a room and hurts across rooms, leaving it close to raw distance overall (Appendix C.5).

5.2 Better Temporal Fit Does Not Guarantee Better Planning

PushT leaves many tasks unsolved and separates temporal prediction from planning. Under the common training procedure, the linear cost reaches 44% success, against 36% for raw distance and 29% for the joint scorer (Table 1). The nonlinear cost trails the linear cost by about 3 points, an inconclusive difference before tuning. Yet both flexible costs achieve lower temporal loss on validation pairs from the training episodes (Table 15).

The loss advantage extends to separate episodes. Lower validation loss could reflect memorization of the training episodes. We therefore evaluate temporal loss on episodes excluded from cost training and configuration selection. On these separate episodes, the nonlinear and joint costs again have lower loss than the linear cost in every training run, before and after tuning (Appendix E.4). Across all eighteen optimizer configurations, every nonlinear and joint configuration has lower loss on these episodes than every linear configuration, while every linear configuration has higher development success (Figure 3).

The joint deficit survives controlled evaluation and tuning. Starting every cost from the same initial state for each task leaves the joint deficit essentially unchanged. Tuning improves the joint scorer, yet it still trails the tuned linear cost by about 12 points. The tuned nonlinear cost, which is symmetric like the linear cost, also plans worse by about 4 points. The mismatch between loss and planning is therefore not unique to the asymmetric joint scorer. Averaging the untuned joint scorer over both endpoint orders improves its success by about 8 points and leaves it about 7 points below the linear cost.

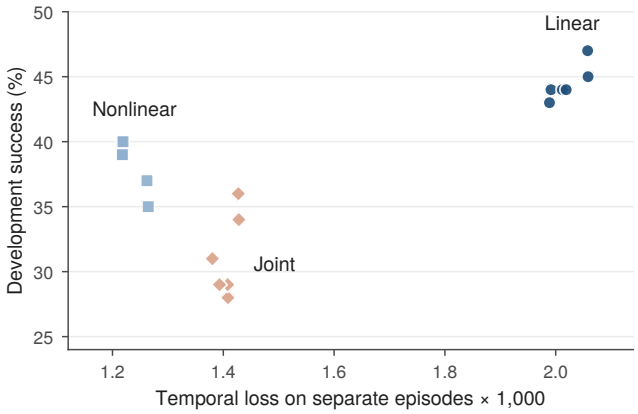

Figure 3: Lower temporal loss does not guarantee better planning. Each point is one optimizer configuration, trained once and evaluated with one planner seed on the same 100 development tasks. Loss uses episodes excluded from cost training and configuration selection and was measured after configurations were chosen. Table 14 gives the values.

The controls behave as in navigation. From the same starting states, with one training run per learned linear cost and two planner seeds, whitening and the shuffled control both plan worse than the trained linear cost. Whitening does not clearly differ from raw distance, and the shuffled control falls below it (Appendix E.5). Appendix E gives all PushT protocols and results.

5.3 Beyond Decoded Euclidean Position

The decoded-position cost asks whether Euclidean proximity in decoded coordinates explains the linear cost's success. Across rooms, the linear cost exceeds it by 10.3 points $[5.9, 15.0]$ and reaches the goal's room in 98% of runs, against 87% for the position cost (Appendix C.1).

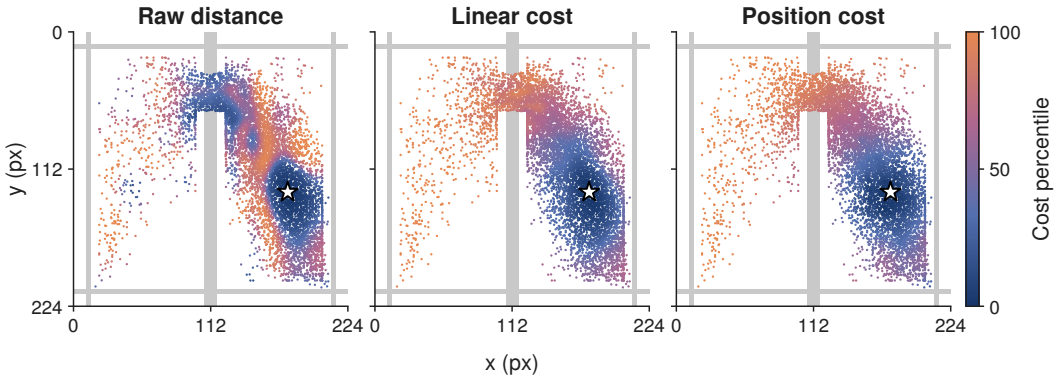

Figure 4: Three costs rank observations relative to one goal (star). Each point is the agent's position in one of 8,000 observations excluded from cost training, colored by its cost to the goal. Gray bars are walls, with the doorway in the divider. Raw latent distance assigns low cost to observations across the wall near the doorway, while the linear and decoded-position costs keep low cost in the goal's room (Appendix F.1).

To see how the two costs order observations, we evaluate 399 goals from the navigation pool across three LeWM models. For each goal, we select pairs of observations for which Euclidean proximity and a doorway distance estimate disagree, and measure how often each cost orders them like the doorway estimate. The linear cost agrees more often than decoded position for 84–95% of goals, depending on the model, while the position cost does better for some goals (Figure 4 and Appendix F.2). The learned linear maps stay numerically full rank but concentrate their weight. Raw distance weights every direction equally, so any two directions receive about one percent of its

weight. Every linear map from the second task set instead gives the two directions that decode position roughly a tenth to a fifth of its weight, far more than random directions of the same dimension (Appendix F.3). Temporal training thus concentrates weight on structure that is already present in the frozen latent space.

6 DISCUSSION

The results distinguish accurate temporal prediction from useful planning decisions. Temporal regression averages errors over logged observation pairs, and elapsed time under the data-collection policy is only a proxy for progress toward the goal. The planner instead ranks predicted outcomes for a single goal. On PushT, these criteria separate. Across gaps of up to 100 steps, the nonlinear cost orders logged observations by elapsed time more accurately than the linear cost, but orders them by their error to the goal state less accurately (Table 2). More accurate temporal prediction can therefore accompany poorer ordering by task state.

Among the learned costs, correlation with state error across all gaps orders the three families in the same way as planning success, both before and after tuning. This correspondence is incomplete. Raw distance has slightly higher concordance with state error within 25 logged steps than the linear cost despite planning worse. Tuning the joint scorer improves its correlation with state error and its planning success while increasing its selection regret on the logged banks (Appendix E.6). These diagnostics rank encoded observations from logged trajectories, whereas the planner ranks predicted outcomes generated by its own action search, and how the costs order those predictions remains a distinct question. Temporal accuracy, ordering by task state and online planning success should therefore be evaluated separately. In navigation, where planning is near ceiling, loss on separate episodes instead favors the linear cost in most runs (Table 6).

Table 2: Temporal ordering, ordering by task state and planning success separate in PushT. Each cost scores the 100 observations preceding each of 500 goals from separate episodes. Concordance uses observations within 25 steps of the goal, and regret measures how much farther from the goal state the lowest-cost observation is than the closest one. Success comes from the separate 200-task evaluation with the same starting state for every cost. Appendix E.6 gives definitions and intervals.

	Logged observations				Planning
Cost	Time τ_b	State τ_b	State concordance	Regret	Success (%)
Raw distance	0.533	0.526	0.906	0.0056	35.8
Linear	0.834	0.629	0.898	0.0056	44.7
Linear, tuned	0.831	0.631	0.901	0.0060	46.1
Nonlinear	0.869	0.604	0.892	0.0064	41.3
Nonlinear, tuned	0.869	0.604	0.893	0.0065	42.0
Joint	0.823	0.569	0.874	0.0275	29.1
Joint, tuned	0.829	0.581	0.887	0.0484	33.8

The linear result is also a statement about the frozen representation. A full quadratic cost reweights latent directions and their interactions without changing the predictive model. In navigation, this reweighting recovers most of the improvement and outperforms decoded Euclidean position across rooms, while the tested covariance whitening and shuffled-target alternatives leave substantially lower success. The frozen representation therefore supports a useful planning geometry, which temporal training of a linear cost recovers in these tasks. The linear cost also has practical structural advantages. It uses about one seventh as many parameters as the joint scorer, is symmetric by construction, and is represented by a single matrix whose spectrum can be inspected directly.

Extending the comparison. The evidence covers two model families in navigation and one released model in PushT. Crossing model families with environments, for instance a frozen DINO-WM on PushT, would show how far the pattern extends. Broader optimizer searches, further unsupervised baselines and directed reachability targets are natural extensions. Appendix B details the coverage and statistical conditioning of every study.

7 CONCLUSION

A linear map trained on temporal separation is a strong planning cost for the evaluated frozen world models, and the tested whitening and shuffled target controls do not recover its performance. More flexible scorers help in some comparisons, yet on PushT their lower temporal loss, including on episodes excluded from cost training and configuration selection, does not translate into better planning. Learned costs should therefore be compared with a linear alternative given comparable supervision, training resources and model selection opportunities.

AI USE STATEMENT

Generative AI tools were used for language revision, code generation and modification, figure preparation, and extraction of results from saved artifacts. All AI-assisted outputs were reviewed by the authors, who retained responsibility for experimental decisions, reported results, and the final manuscript. Generative AI was not used for research conceptualization, key contributions, literature discovery, or substantive scientific content generation.

ETHICS STATEMENT

The experiments evaluate goal-conditioned planning in simulated navigation and block-pushing environments. The reported success rates do not establish safety or reliability for deployment on physical robots. Such deployment would require separate assessment of model errors, distribution shift, and interactions with people and the physical environment.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the cost constructions, training and evaluation. Appendices A and B document the architectures, initialization, temporal-pair sampling, task construction, preprocessing and statistical analysis. Appendices C, D and E report the navigation and block-pushing comparisons, including the development search and controlled evaluations. Appendix F defines the ordering and representation diagnostics, and Appendix G gives the earlier quadratic comparison.

REFERENCES

Randall Balestriero and Yann LeCun. LeJEPA: Provable and Scalable Self-Supervised Learning Without the Heuristics. *arXiv preprint arXiv:2511.08544v3*, 2025.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive Learning as Goal-Conditioned Reinforcement Learning. *Advances in Neural Information Processing Systems*, 2022.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality Reduction by Learning an Invariant Mapping. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. *International Conference on Machine Learning*, 2019.

Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical Distance Learning for Semi-Supervised and Unsupervised Skill Discovery. *International Conference on Learning Representations*, 2020.

Prateek Jain, Brian Kulis, Jason V. Davis, and Inderjit S. Dhillon. Metric and Kernel Learning Using a Linear Transformation. *Journal of Machine Learning Research*, 2012.

Brian Kulis. Metric Learning: A Survey. *Foundations and Trends in Machine Learning*, 2013.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective Mismatch in Model-based Reinforcement Learning. *Conference on Learning for Dynamics and Control*, 2020.

Liangyu Li, Shengzhi Wang, Libin Qiu, Mingliang Xiong, and Qingwen Liu. World Model Control by Trajectory Reachability Metrics. *arXiv preprint*, 2026a.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Predictive but Not Plannable: RC-aux for Latent World Models. *arXiv preprint*, 2026b.

Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric Residual Network for Sample Efficient Goal-Conditioned Reinforcement Learning. *AAAI Conference on Artificial Intelligence*, 2023.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. *International Conference on Learning Representations*, 2023.

Lucas Maes, Quentin Le Lidec, Luiz Facury, Nassim Massaudi, Ayush Chaurasia, Francesco Capuano, Richard Gao, Taj Gillin, Dan Haramati, Damien Scieur, Yann LeCun, and Randall Balestriero. stable-worldmodel: A Platform for Reproducible World Modeling Research and Evaluation. *arXiv preprint*, 2026a.

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*, 2026b.

Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and Achieving Goals via World Models. *Advances in Neural Information Processing Systems*, 2021.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making. *International Conference on Machine Learning*, 2024.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation Policies with Hilbert Representations. *International Conference on Machine Learning*, 2024.

Reuven Y. Rubinstein. The Cross-Entropy Method for Combinatorial and Continuous Optimization. *Methodology and Computing in Applied Probability*, 1999.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-Parametric Topological Memory for Navigation. *International Conference on Learning Representations*, 2018.

Joyjeet Singh. The Objective Is the Bottleneck: Latent World Models Encode What Their Planners Cannot Use. *arXiv preprint*, 2026.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from Reward-Free Offline Data: A Case for Planning with Latent Dynamics Models. *Advances in Neural Information Processing Systems*, 2025.

Lucas Thil, Jesse Read, Rim Kaddah, and Guillaume Doquet. Subspace-Decomposed JEPAs: Disentangling Progression and Content in Latent World Models. *arXiv preprint*, 2026.

Tongzhou Wang and Phillip Isola. On the Learning and Learnability of Quasimetrics. *International Conference on Learning Representations*, 2022.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal Goal-Reaching Reinforcement Learning via Quasimetric Learning. *International Conference on Machine Learning*, 2023.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim G. J. Rudner, Yann LeCun, and Mengye Ren. Temporal Straightening for Latent Planning. *International Conference on Machine Learning*, 2026.

Kilian Q. Weinberger and Lawrence K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, 2009.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features Enable Zero-shot Planning. *International Conference on Machine Learning*, 2025.

A Cost Definitions and Training

Notation and task names. D denotes the direct linear cost, F the factorized linear control, N the shared nonlinear cost, J the joint scorer, and P_{XY} the decoded-position cost. The original tasks are the previously evaluated navigation set, called the second task set in the main text. The balanced tasks use refitted costs and equal numbers of tasks within and across rooms. The earlier opposite-room tasks are a separate pair set from previously evaluated episodes. They are not the across-room stratum of the balanced study. In the main text, *tuned* refers to the development-selected PushT configurations, labeled *selected* in the appendix tables.

A.1 Cost architectures

The original LeWM navigation comparison uses $d = 192$. The maps in Equation 3 have $A \in \mathbb{R}^{192\times 192}$, $W_1 \in \mathbb{R}^{256\times 192}$, $b_1 \in \mathbb{R}^{256}$ and $W_2 \in \mathbb{R}^{192\times 256}$, with no output bias. D has 36,864 parameters; N and F each have 98,560. All three train and deploy $\|f(z) - f(g)\|_2^2/192$. For D this gives $(z-g)^\top A^\top A(z-g)/192$. F implements the same quadratic family through W_2W_1, since its bias cancels between endpoints. Neither linear construction imposes a trace constraint or uses a learned scalar calibration in the main comparison.

Joint scorer. J uses $[z, g, z-g, |z-g|]$ as its input, where the absolute value is elementwise. Its network has dimensions $768 \to 256 \to 256 \to 1$, SiLU hidden activations and a Softplus output, with 262,913 parameters. It regresses the unscaled output onto $|\Delta t|/224$ and deploys $224h_\phi$. The original heads use ordered endpoints. The separate deployment-only symmetrization of those heads is described in Appendix E.2.

Position reference. The original navigation comparison uses one affine position decoder $\widehat{x} = Bz + b$ per world model, fitted by unregularized float64 least squares on centered encodings and pixel coordinates, with relative singular-value cutoff 10^{-12}. It uses the available cached observations from the cost-training episodes. Planning casts the stored coefficients to float32 and uses $\|B(z - g)\|_2^2$, without refitting, whitening or scalar calibration. The intercept cancels. Probe coverage and accuracy are reported in Appendix F.3.

A.2 Temporal pairs and common training

For each original LeWM model, all four temporal costs use the same 100,000 training and 10,000 pair-validation identities, encoded once, with target $|\Delta t|/224$. The common eligible pool contains 9,600 episodes. The 400 episodes supplying the original planning evaluations are excluded from both training and pair validation.

The sampler uses NumPy `default_rng` with seeds $w+1$ for training and $w+2$ for pair validation, where w is the world-model seed. Each draw selects an eligible episode uniformly. For its length L, it selects a gap k uniformly from $1, \ldots, L-1$ and a start uniformly from $0, \ldots, L-k-1$, then reverses the endpoint order with probability $1/2$. There is no additional gap cap. Sampling is with replacement, allowing repeated and reversed pairs. Training and validation use independent draws from the same episode pool and can share observations or complete pairs. Validation therefore measures fit within that pool rather than performance on separate source episodes.

Initialization and minibatches. World-model seeds and cost-training seeds each range over $\{42, 2026, 3072\}$. N uses the PyTorch `Linear` initialization on CPU under the training seed. F receives copies of those tensors, and D starts at $A = W_2W_1$. N and F share initial parameters but generally different functions. F and D share their initial endpoint-distance function. J uses a separate CPU generator with the training seed and shares neither parameters nor an initial function with the maps. At zero-based epoch e, a CPU generator seeded with the training seed plus e produces the minibatch permutation shared by all four costs.

Optimization and selection. Each run completes 40 epochs of AdamW with learning rate 10^{-3}, weight decay 10^{-4}, moments $(0.9, 0.999)$ and $\epsilon = 10^{-8}$. Training uses minibatch-mean Smooth-L1 loss with $\beta = 1$ and batch size 1,024. No scheduler, early stopping or gradient clipping is

used. Validation sums all 10,000 example losses and divides by 10,000. The earliest strict minimum over epochs 1–40 selects the model; epoch zero is ineligible. Selected costs are frozen before their planning evaluations. The original fits use float32 on A100 GPUs with PyTorch 2.6.0, matrix-multiplication TF32 disabled and cuDNN TF32 enabled. Appendix B.3 reports the numerical-configuration checks and their coverage.

The balanced navigation, PLDM and original PushT comparisons retain this objective and optimization budget, with their episode pools, repetition coverage and any sampler differences specified in their respective appendices. For parameter training, the PushT search varies learning rate and weight decay. Development planning success selects among the same six configurations for each family; within each run, temporal validation loss still selects the epoch (Appendix E.2).

A.3 Selected epochs and training histories

Table 3 gives all 36 selected original-navigation costs. Joint heads select epochs 6–12, while two direct linear runs select epoch 40.

Table 3: Selected epochs and temporal validation losses for all 36 original LeWM navigation fits. Each entry is epoch / loss, with sample-weighted Smooth-L1 loss multiplied by 10^3. D, F, N and J share the validation pairs within each world model. Validation pairs come from the cost-training episode pool. The earliest minimum validation loss over the completed 40 epochs selects each model.

World-model seed	Training seed	D	F	N	J
42	42	38 / 2.569	27 / 2.594	26 / 2.568	7 / 2.579
42	2026	40 / 2.569	39 / 2.592	29 / 2.555	12 / 2.601
42	3072	32 / 2.554	32 / 2.577	39 / 2.566	11 / 2.591
2026	42	32 / 2.662	37 / 2.686	34 / 2.633	9 / 2.700
2026	2026	27 / 2.678	28 / 2.691	28 / 2.647	7 / 2.653
2026	3072	36 / 2.663	36 / 2.679	36 / 2.636	6 / 2.683
3072	42	36 / 2.684	32 / 2.697	36 / 2.684	8 / 2.682
3072	2026	38 / 2.685	31 / 2.706	10 / 2.671	6 / 2.690
3072	3072	40 / 2.689	39 / 2.701	15 / 2.673	9 / 2.699

A.4 Calibration-link sensitivity within D

A separate sensitivity check keeps the direct linear parameterization but changes its training readout. For $q(z, g) = \|Az - Ag\|_2^2$, two learned scalars a, b define

$$\widetilde{c}_{\text{link}}(z, g) = \frac{\text{softplus}(a)\sqrt{q(z, g) + 10^{-8}} + b}{224}.$$

This prediction is regressed onto $|\Delta t|/224$. One run per world model uses training seed 42 and starts from the saved $A = W_2 W_1$ initialization, with the other settings above unchanged. All three runs select epoch 40. Planning still uses $q/192$, evaluated on the original 200 tasks with planner seed 42.

Table 4: Changing the direct linear cost's training readout does not establish a planning difference in this sensitivity check. Counts are successes on the original 200 tasks per model, using training seed 42 and planner seed 42. Differences are D-link minus original D, in points with marginal 95% paired intervals. The final row uses equal world-model weights.

World model	Original D	D-link	Difference (points)	95% interval
42	187/200	190/200	$+1.50$	$[-1.50, 5.00]$
2026	197/200	196/200	-0.50	$[-3.00, 1.50]$
3072	198/200	199/200	$+0.50$	$[-1.00, 2.00]$
Equal-weight	582/600	585/600	$+0.50$	$[-0.83, 1.67]$

Table 4 uses the original training-seed-42 D instance for each world model as comparator, with pooled success of 97.00%. It is distinct from the nine-fit aggregate. The interval includes zero for this change within the linear cost. Appendix G describes the earlier full-SPD construction.

A.5 Covariance whitening and shuffled training targets

These later controls use the balanced navigation models and the released PushT model. Their definitions were fixed after the earlier results and before their own planning outcomes. The predictive models, planning tasks and success criteria are unchanged.

Covariance construction. For each model, we take the unique observation endpoints in its 100,000 training pairs, deduplicated by source observation identity and weighted equally. Navigation observations belong to the 8,600 balanced cost-training episodes. PushT observations belong to its cost-training pool.

From the stored float32 encodings, float64 accumulation gives

$$\mu = \frac{1}{n}\sum_i z_i, \qquad \Sigma = \frac{1}{n}\sum_i (z_i - \mu)(z_i - \mu)^\top.$$

With the fixed additive ridge in Equation 5, an eigendecomposition $\Sigma = Q\,\mathrm{diag}(\sigma_i)Q^\top$ gives

$$W = Q\,\mathrm{diag}\Big((\sigma_i + \lambda)^{-1/2}\Big)\,Q^\top.$$

Planning casts W to float32 and uses $\|W(z-g)\|_2^2/d$, with $d = 192$. The predictor continues in its original coordinates. There is no eigenvalue truncation or outcome-based choice of the ridge.

Shuffled-target training. The shuffled linear cost, denoted D_π, uses the original training pairs, saved training-seed-42 initialization and minibatch permutations for the corresponding direct linear cost. One permutation of the 100,000 target indices, drawn with `default_rng(20260926)`, is applied to each model's target vector. It preserves the target multiset. All four runs complete the original 40-epoch AdamW schedule with learning rate 10^{-3} and weight decay 10^{-4}. Real pair-validation targets select the earliest strict minimum.

Navigation results appear in Appendix C.5, and PushT results in Appendix E.5. For navigation, the comparator is the refitted linear cost from the balanced study at training seed 42. For PushT, it is the linear cost trained with the original optimizer setting at training seed 42 and evaluated under the same fixed resets. Each comparator uses the same planner coverage as its controls.

B Evaluation Protocols and Statistical Analysis

B.1 Study coverage and source episodes

Table 5 gives planning coverage. The separate-episode loss diagnostic is described in Appendix E.4. Costs share task identities within each comparison. Original LeWM cost training excludes 400 of the dataset's 10,000 episodes: 100 development, 100 Test and 200 original evaluation episodes, with one task per episode. The balanced study reserves another 1,000 episodes from the remaining 9,600 and refits its costs on 8,600. Throughout the paper, excluded episodes are excluded from cost training, pair validation and probe fitting. Whether they were seen during world-model pretraining is unknown. PLDM training excludes the original evaluation episodes. Task construction and model-specific protocols are detailed in Appendices C.1, D and E.

Three-run entries use seeds 42, 2026 and 3072; two planner runs use 42 and 2026, and one-run entries use 42. The LeWM navigation models have seeds 42, 2026 and 3072. PLDM seeds 42 and 2026 are reported separately. P_{state} is the unscaled PushT state decoder. J_{sym} reuses the original joint heads without additional training.

B.2 TwoRoom task construction and preprocessing

The environment is `swm/TwoRoom-v1`; the dataset has 920,809 rows. The three LeWM models use retained epoch-10 checkpoints. Positions are pixel-centre coordinates with x rightward and

Table 5: Planning coverage. Training counts are per trained map or network and world model. Raw distance and whitening have no training repetitions, and decoded references use one probe per model. The fixed-reset PushT row covers original and selected D/N/J, raw distance and symmetrized original J. Development evaluates six configurations per family. In the linear-control rows, W is covariance whitening and D_π uses shuffled training targets. Their D comparators use training seed 42. Appendix G describes the earlier quadratic comparison.

Comparison	Costs	Tasks	Models	Training runs	Planner runs
Balanced TwoRoom	Raw, D/N/J, P_{XY}	200	3	3	3
Original TwoRoom	Raw, D/F/N/J, P_{XY}	200	3	3	1
Original, expanded	Raw, D/J, P_{XY}	200	3	3	3
Earlier opposite-room	Raw, D/F/N/J, P_{XY}	200	3	3	1
PLDM, primary	Raw, D/N/J	200	1	1	1
PLDM, second	Raw, D	200	1	1	1
PushT, original	Raw, D/N/J, P_{state}	200	1	3	2
PushT, fixed resets	Raw, D/N/J, J_{sym}	200	1	3	2
PushT, development	D/N/J	100	1	1	1
Balanced, linear controls	Raw, W, D/D_π, P_{XY}	200	3	1	1
PushT, linear controls	Raw, W, D/D_π	200	1	1	2

y downward. The physical divider is at $x = 112$, with doorway centre $(112, 49)$. Right-room membership is $x > 112$; equality belongs to the left room.

Test and original evaluation tasks were constructed sequentially, excluding previously used episodes. Eligible nominal 50-step pairs use the same episode, with the end-exclusive loader deploying start+49. Endpoints must straddle the data-inferred horizontal partition, be at least 35 pixels apart, have via-point distance at least 150 pixels and a nonnegative distance ratio, and each be at least 25 horizontal pixels from the inferred doorway. Eligible pairs were shuffled with `default_rng(42)`, stably sorted in descending order of via-point distance times $\log(1 + \text{minimum horizontal doorway distance})$, and the highest-ranked pair per episode was selected. These constructors use geometry rather than planning outcomes. This horizontal partition differs from the physical rooms: all 100 development tasks are within-room, while the original 200 tasks contain 194 within-room and six across-room tasks, three per direction.

Images use float32 scaling, ImageNet normalization and bilinear resizing to 224 pixels. Action and proprioceptive `StandardScaler` states were reconstructed with the producer's fitting operation on the full source dataset and then frozen. Consequently, cost-training exclusions do not apply to these normalization statistics. The policy inverse-transforms action blocks, and physical commands are clipped to $[-1, 1]$. These transforms are shared across costs.

B.3 Planning, success and numerical configuration

CEM uses 300 candidates, 30 iterations and 30 elites. A plan has five blocks of five physical actions; its final elite-mean sequence is executed in full within a 50-action budget, subject to success termination. Original LeWM navigation uses history size and frame skip one. Its costs are terminal-only, with auxiliary reachability, temporal-distance and action penalties disabled. Original 200-task LeWM cells use two row-sorted 100-task blocks, resetting the producer and CEM seed at each block. Native PLDM and PushT interfaces and fixed-reset handling are specified in their respective appendices.

Success. TwoRoom success is any evaluated position error strictly below 16 pixels, and rollouts continue after a hit. PushT requires the combined agent/block position-error norm below 20 pixels and the symmetry-adjusted angle error below 20°. It stops at the first success.

Numerical configuration. The first navigation comparisons use A100 float32 with cuDNN TF32 enabled, under which online encodings differ from the training cache by up to 0.023 in maximum absolute value. Costs computed from cached and online encodings nonetheless have Spearman

correlation of at least 0.9999 for all 39 tested heads. All later navigation and PushT runs use float32 with TF32 disabled and cuDNN deterministic mode enabled.

B.4 Paired analysis and study-specific priorities

We average training and planner repetitions within task, cost and model, then weight tasks and the three LeWM navigation models equally. Raw and decoded references have no training repetitions. PLDM models remain separate. Each task set has distinct source episodes; repeated runs are not additional resampling units. Paired 95% intervals use 10,000 joint task resamples with `default_rng(42)` and linear percentile endpoints. All costs, models and repetitions retain each sampled task's multiplicity. Overall comparisons resample the complete evaluation set, and balanced strata resample their respective 100 tasks. Contrasts are formed within each resample. The intervals are marginal and unadjusted. Models, trained costs and planner seeds are averaged rather than resampled. Tuning intervals additionally condition on the development set, selected configurations and fixed resets, excluding selection uncertainty. Intervals containing zero do not establish equivalence.

Each study designates its primary contrast before its outcomes, and its result table marks it. All other contrasts are secondary, and trajectory analyses are exploratory.

The first navigation comparisons reuse tasks evaluated in earlier experiments, and the earlier opposite-room set reuses their source episodes. Every later study was motivated by earlier results and fixed before its own outcomes. These include the balanced navigation study, the controlled-reset PushT follow-up and its configuration search, the separate-episode loss diagnostic and the linear controls. The PushT follow-up and the linear controls reuse previously evaluated final tasks, and configuration selection was committed before final evaluation on them.

C Complete TwoRoom Results

All rates use the averaging and paired task-bootstrap procedure in Appendix B.4. Differences are percentage points, with marginal, unadjusted 95% intervals. The balanced study uses new costs; the original and earlier opposite-room studies share their retained costs. These task sets are analyzed separately.

C.1 Balanced tasks and refitted costs

Partition and refitting. From the original 9,600 cost-training episodes, we reserve 1,000 uniformly without replacement using `default_rng(42)`. The remaining 8,600 supply the 100,000 training and 10,000 pair-validation examples for each frozen model, using the conditional episode/gap/start sampler of Appendix A.2. Sampler seeds are the world-model seed plus one and plus two. Another 10,000 pairs from reserved episodes, sampled with the model seed plus three, measure selected-model loss for reporting only. They do not select the model. Each world model receives three new D/N/J runs under the common 40-epoch procedure, with D initialized at the product of N's initial weight matrices and J initialized separately. All costs are frozen before task construction. The predictive models are unchanged.

A new affine probe per model uses centered float64 least squares on cached observations from the 8,600 episodes, with relative singular-value cutoff 10^{-12}. The 200 task-goal observations are excluded from its training and selection. On these goals, (R_x^2, R_y^2) equals $(0.9982, 0.9960)$, $(0.9969, 0.9971)$ and $(0.9988, 0.9960)$ for models 42, 2026 and 3072. The full affine prediction is evaluated without clipping or outcome filtering; planning uses $\|B(z-g)\|_2^2$.

Temporal loss on reserved episodes. Table 6 reports the in-pool validation and reserved-episode losses of the same 27 selected costs. Reserved pairs are used only after epoch selection. The linear cost has lower reserved loss than the nonlinear cost in eight of nine paired runs and lower than the joint scorer in seven of nine. The corresponding in-pool counts are three of nine and nine of nine.

Task construction. Goals are logged 49 steps after the start, with the loader also requiring a valid row at start plus 50. Both strata require a doorway distance estimate of at least 96 pixels. Across-

Table 6: Temporal losses on the training episode pool and reserved episodes for all 27 balanced navigation costs. Every loss is multiplied by 10^3. Each row compares the same D/N/J instances on the same pairs within that world model. Pair validation selects the epoch. Reserved losses are measured only after selection on episodes excluded from cost training and epoch selection.

		Pair validation			Reserved episodes		
Model	Training seed	D	N	J	D	N	J
42	42	2.558	2.551	2.602	2.781	2.824	2.764
42	2026	2.581	2.556	2.627	2.766	2.802	2.799
42	3072	2.560	2.570	2.600	2.782	2.796	2.785
2026	42	2.597	2.589	2.607	2.756	2.783	2.795
2026	2026	2.588	2.586	2.616	2.765	2.798	2.743
2026	3072	2.587	2.578	2.621	2.752	2.781	2.779
3072	42	2.626	2.625	2.637	2.704	2.716	2.736
3072	2026	2.621	2.621	2.634	2.721	2.713	2.729
3072	3072	2.626	2.636	2.638	2.695	2.719	2.703

room pairs additionally require different endpoint rooms and $|x_g - 112| \geq 16$, placing the success region in the destination room. Among reserved episodes, 568 have an eligible across-room window and 296 a within-room window. Using `default_rng(2026)`, we select 100 eligible across-room episodes uniformly without replacement and one eligible window uniformly within each. Of the 267 remaining eligible within-room episodes, another 100 are sampled by the same rule. All 200 source episodes are distinct. Task identities and analysis were fixed before planning outcomes.

Evaluation. The evaluation crosses three world models, three training seeds for each learned cost and three planner seeds, and rollouts continue after a hit. Crossing requires a post-action position in the goal's room after starting in the other room, using $x > 112$ without interpolation. It is separate from the strict 16-pixel any-hit criterion. Table 7 reports the pooled rates and all six declared contrasts. Overall J−D is primary; the other contrasts are secondary. Overall intervals resample 200 tasks, and stratum intervals resample their respective 100 tasks.

Table 7: Balanced TwoRoom results averaged over all three planner seeds. The upper panel gives success and crossing percentages; across-room and within-room strata contain 100 tasks each. Crossing is visiting the destination room, separately from goal success. The lower panel contains all six declared contrasts in points with marginal 95% paired intervals. Only overall J−D is primary.

Cost	All 200	Across rooms	Within room	Crossing
Raw	18.11	30.33	5.89	60.33
D	96.28	95.15	97.41	98.37
N	96.07	94.67	97.48	98.04
J	96.56	94.22	98.89	98.41
P_{XY}	91.39	84.89	97.89	87.22

Contrast	Tasks	Difference	95% interval
$J - D$	All 200 (primary)	$+0.28$	$[-0.69, 1.26]$
$J - D$	Across rooms	-0.93	$[-2.22, 0.41]$
$J - D$	Within room	$+1.48$	$[0.19, 3.04]$
$D - P_{\mathrm{XY}}$	Across rooms	$+10.26$	$[5.89, 15.04]$
$N - D$	All 200	-0.20	$[-0.96, 0.57]$
$D - \mathrm{raw}$	All 200	$+78.17$	$[73.70, 82.44]$

Trajectory analysis. After the planning results were known, we compared capped first-hit step (the first post-action step with error below 16 pixels, or 51 without a hit), final error and final success at action 50, using all executions and the analysis above. Relative to the linear cost, the joint scorer has a higher capped first-hit step ($+0.80$ $[0.41, 1.18]$) and ends farther from the goal ($+2.08$ pixels

$[1.43, 2.77]$), with no clear difference in final success (-0.09 points $[-1.09, 0.93]$). The nonlinear and linear costs do not clearly differ on any of the three measures. Across rooms, the linear cost ends closer to the goal than the position cost (-6.93 pixels $[-8.98, -5.13]$).

C.2 Original single-planner comparisons

Table 8 preserves the original comparisons at planner seed 42. Each temporal cost uses three training runs per model; raw and decoded position have one instance. N−F is primary in the N/F/D studies. The joint/position extension has J−D and $D - P_{\mathrm{XY}}$ on the original tasks as primaries. All other entries are secondary. On the original tasks, N exceeds F in every paired training run. The model-specific J−D differences are $+3.67$, -0.33 and $+0.17$ points for models 42, 2026 and 3072.

Original task composition. There are 194 within-room and six across-room tasks. Across-room successes are 40/54 for N and 37/54 for each linear form; raw succeeds on none of its 18 executions. Within-room counts are 1720/1746, 1685/1746 and 1697/1746 for N, F and D.

Table 8: Original and earlier opposite-room comparisons, each on 200 tasks at planner seed 42. Rates average training runs within task/model and then weight models equally. Contrast cells give points and marginal 95% paired intervals. † marks N−F primaries; ‡ marks the original-task joint/position primaries. All other displayed contrasts are secondary.

Cost	Original (%)	Earlier opposite-room (%)
Raw	18.00	51.33
F	95.67	97.50
D	96.33	97.33
N	97.78	97.50
J	97.50	97.22
P_{XY}	94.67	95.00

Contrast	Original	Earlier opposite-room
$N - F$	$+2.11\ [1.17, 3.22]^{\dagger}$	$0.00\ [-0.89, 0.89]^{\dagger}$
$N - D$	$+1.44\ [0.50, 2.50]$	$+0.17\ [-0.72, 1.00]$
$F - D$	$-0.67\ [-1.39, 0.06]$	$+0.17\ [-0.72, 1.00]$
$J - D$	$+1.17\ [0.11, 2.33]^{\ddagger}$	$-0.11\ [-1.28, 1.06]$
$J - N$	$-0.28\ [-1.22, 0.67]$	$-0.28\ [-1.28, 0.78]$
$J - F$	$+1.83\ [0.67, 3.11]$	$-0.28\ [-1.22, 0.67]$
$D - P_{\mathrm{XY}}$	$+1.67\ [-0.50, 3.89]^{\ddagger}$	$+2.33\ [0.44, 4.50]$
$P_{\mathrm{XY}} - \mathrm{raw}$	$+76.67\ [71.83, 81.33]$	$+43.67\ [37.33, 50.00]$

C.3 Planner replication on the original tasks

The replication adds planner seeds 2026 and 3072 to the retained seed-42 outcomes for raw, D, J and decoded position, using the same models, trained costs and 200 tasks. N and F have no additional planner evaluation, and all other settings are unchanged. The replication was specified before its own outcomes.

Table 9 retains each layer and all six pooled contrasts. Overall J−D is primary; model-specific J−D, D−position and D−raw are secondary. The pooled interval includes zero, while the model-42 interval remains positive.

Table 9: Original-task planner replication. Upper-panel rates are percentages; the last column gives J−D in points with its 95% paired interval. The last row averages all three planner seeds. The lower panel contains the remaining five declared pooled contrasts, all secondary, with the same units and interval convention.

Planner seed	Raw	D	J	P_{XY}	$J - D$
42	18.00	96.33	97.50	94.67	$+1.17\ [0.11, 2.33]$
2026	18.17	96.44	97.11	93.50	$+0.67\ [-0.50, 1.83]$
3072	17.00	96.39	97.17	94.50	$+0.78\ [-0.33, 1.94]$
Three-seed mean	17.72	96.39	97.26	94.22	$+0.87\ [-0.07, 1.93]$

Pooled comparison	Difference	95% interval
$J - D$, model 42	$+2.78$	$[0.89, 5.00]$
$J - D$, model 2026	-0.33	$[-1.89, 1.33]$
$J - D$, model 3072	$+0.17$	$[-0.78, 1.44]$
$D - P_{\mathrm{XY}}$	$+2.17$	$[0.26, 4.11]$
$D - \mathrm{raw}$	$+78.67$	$[74.20, 83.02]$

C.4 Earlier opposite-room tasks

After the original results, 200 new pairs were fixed from 234 of the excluded 400 episodes with unused eligible windows. Source episodes had been evaluated previously. Starts and deployed start+49 goals lie across the physical divider: 38 left-to-right and 162 right-to-left. No start is inside the goal disk; the minimum separation is 23.9 pixels. The primary N−F and other pooled comparisons appear in Table 8. The remaining N/F/D-study secondary contrast is D−raw, $+46.00$ points $[39.78, 52.22]$.

For 107 goals, the 16-pixel success disk extends into the start room. For the remaining 93, a hit requires a goal-room position at that step. Table 10 retains the exploratory split, including the earlier-recipe references.

Table 10: Exploratory divider-distance split of the earlier opposite-room tasks. Entries are successes divided by repeated evaluations, not distinct tasks. F/D/N have nine runs per task; raw and each earlier-recipe reference have three. For the 93 farther goals, a hit places the agent in the goal's room.

Cost	Near divider: 107 tasks $\lvert x_g - 112\rvert < 16$	Farther goals: 93 tasks $\lvert x_g - 112\rvert \geq 16$
Raw	194/321	114/279
F	957/963	798/837
D	955/963	797/837
N	956/963	799/837
Earlier full SPD	316/321	264/279
Earlier TRM	320/321	261/279

C.5 Covariance and shuffled-target controls

The two controls of Appendix A.5 use all 200 balanced tasks, all three world models and planner seed 42. Both learned linear costs use training seed 42. The real-target linear, raw and position comparators use the corresponding existing cells with the same model, planner, task and hardware coverage.

Table 11 reports all rates and thirteen declared contrasts, analyzed as in Appendix B.4, with each stratum resampling its 100 tasks.

Whitening improves over raw distance within a room and reduces success across rooms. Its overall raw-distance comparison remains inconclusive. Both whitening and shuffled-target training perform substantially below the temporal linear cost.

Table 11: Linear controls on balanced TwoRoom with reduced repetition coverage. Rates are percentages using three world models, planner seed 42 and training seed 42 for both learned linear costs. Each stratum contains 100 tasks. The lower panel gives all thirteen contrasts in points with marginal 95% paired intervals. Overall W−D and $D_\pi - D$ are principal comparisons in this late exploratory study. All other contrasts are secondary. These rates are separate from the broader repetition averages in Table 1.

Cost	All 200	Across rooms	Within room
Raw	19.00	31.33	6.67
Whitening W	20.33	20.00	20.67
Shuffled linear D_π	11.17	12.00	10.33
Temporal linear D	97.00	96.33	97.67
Position P_{XY}	90.67	83.67	97.67

Tasks	Contrast	Difference	95% interval
All 200	$W - D$	-76.67	$[-80.67, -72.33]$
	$W - \mathrm{raw}$	$+1.33$	$[-4.17, 6.67]$
	$D_\pi - D$	-85.83	$[-88.83, -82.67]$
	$D_\pi - \mathrm{raw}$	-7.83	$[-13.33, -2.50]$
Across rooms	$W - D$	-76.33	$[-81.67, -70.33]$
	$W - \mathrm{raw}$	-11.33	$[-18.67, -4.00]$
	$W - P_{\mathrm{XY}}$	-63.67	$[-70.67, -56.33]$
	$D_\pi - D$	-84.33	$[-88.67, -79.67]$
	$D_\pi - \mathrm{raw}$	-19.33	$[-28.00, -11.00]$
Within room	$W - D$	-77.00	$[-83.00, -70.67]$
	$W - \mathrm{raw}$	$+14.00$	$[7.33, 21.00]$
	$D_\pi - D$	-87.33	$[-91.33, -83.00]$
	$D_\pi - \mathrm{raw}$	$+3.67$	$[-2.00, 9.00]$

D PLDM NAVIGATION EVALUATION

World-model training. Two PLDM models were trained with the official `stable-worldmodel` script (Sobal et al., 2025; Maes et al., 2026a) on the 9,600 TwoRoom episodes excluding those supplying the original planning evaluations. Training seeds are 42 and 2026, with seed 42 designated primary. Both complete 100 epochs using bfloat16 automatic mixed precision with float32 model parameters. Both runs resumed once from their epoch-95 checkpoints with model, optimizer and scheduler state restored and random number generators reinitialized. Downstream encoding, cost training and planning use float32 without autocast.

Frozen interface and costs. Each model's encoder, projector and predictor are frozen, with 192-dimensional endpoint representations and a native history of up to three real observations, starting from one. All costs share this interface and blocks of five physical actions. Each model encodes the original 100,000 training and 10,000 pair-validation identities. D is trained once per model at training seed 42; the primary model also receives one N and one J. They use the Smooth-L1/AdamW procedure of Appendix A.2, with the same pair order within a model and target $|\Delta t|/224$. The earliest strict minimum of sample-weighted validation loss over all 40 epochs selects the cost. Primary D/N/J select epochs 26/26/6; secondary D selects epoch 26. J trains its unscaled output and deploys $224h_\phi$. No factorized linear control or position-cost planning arm is evaluated.

Evaluation and analysis. The unchanged 200 original tasks use planner seed 42, full 25-action plans and a 50-action budget. Success is any evaluated position error strictly below 16 pixels, and execution continues after a hit. Each contrast uses 10,000 paired resamples of task/source-episode identities under Appendix B.4. The two models are reported separately. Intervals condition on each model, its selected costs and the evaluated planner seed.

Table 12: The linear improvement appears on both PLDM models. The upper panel gives successes out of the same 200 original TwoRoom tasks, with one training run per learned cost and planner seed 42. Dashes denote unevaluated arms. The lower panel reports all four contrasts in points with marginal 95% paired intervals. Seed-42 D−raw is primary; its N−D and J−D comparisons are secondary, and seed 2026 provides an additional D−raw check.

World-model seed	Raw	D	N	J
42 (primary)	85	197	198	191
2026	83	199	–	–

World-model seed	Contrast	Difference (points)	95% interval
42 (primary)	D − raw	$+56.0$	$[49.0, 63.0]$
42	N − D	$+0.5$	$[-1.0, 2.5]$
42	J − D	-3.0	$[-6.0, 0.0]$
2026	D − raw	$+58.0$	$[51.0, 65.0]$

Both primary-model N−D and J−D intervals include zero. For J−D, two tasks succeed only with J and eight only with D.

E PUSHT COMPARISONS AND ROBUSTNESS

E.1 ORIGINAL TASKS, TRAINING AND EVALUATION

Tasks and model. We use one released LeWM PushT model with a 192-dimensional representation. The partition contains 14,898 cost-training source episodes, 50 probe-development episodes and 3,737 evaluation episodes. Of the latter, 3,719 have a valid 51-observation window. Using `default_rng(2026)`, we sample 200 distinct eligible episodes and one valid start uniformly within each, without a difficulty filter. The goal is the observation 50 logged steps later. Costs are frozen before task construction.

Temporal costs and state reference. D/N/J each use training seeds 42, 2026 and 3072. They share 100,000 training pairs and 10,000 pair-validation examples, with sampler seeds 43 and 44. Sampling is with replacement: uniform eligible episode, conditional uniform gap from 1 to $\min(100, L-1)$, conditional uniform start, and equally likely endpoint orders. Both splits use the cost-training episode pool and may share observations or pairs. The target is $|\Delta t|/224$. All runs use the architectures, initialization and 40-epoch Smooth-L1/AdamW procedure of Appendix A, with learning rate 10^{-3} and weight decay 10^{-4}. The earliest strict minimum of sample-weighted validation loss selects the epoch. J trains its unscaled output and deploys $224h_\phi$.

The affine decoded-state reference P_{state} uses agent and block positions and the cosine and sine of block angle as six targets. It fits 205,659 unique cached training-episode observations by centered float64 least squares, with relative singular-value cutoff 10^{-12}, and deploys $\|B(z-g)\|_2^2$. Targets retain their original scales, mixing pixel coordinates with unit-scale angular features. This is an unscaled state reference, distinct from the navigation position cost.

Planning and replay. Each cost uses planner seeds 42 and 2026. The native interface has a history of three observations and five-action blocks. CEM uses 300 candidates, 30 iterations and 30 elites, executing full 25-action plans within a 50-action budget. Success requires the combined agent/block position-error norm strictly below 20 pixels and symmetry-adjusted block-angle error below $\pi/9$ radians. The initial state is not scored separately. The first successful step freezes that environment; only actual physical actions are logged. Runtime is A6000 float32 with matrix-multiplication and cuDNN TF32 disabled, cuDNN deterministic mode enabled and benchmarking disabled.

Logged-action replay reaches the goal criterion on 198 tasks. Exceptions 74 and 121 remain in all declared comparisons, and no task is replaced. Before planning, the same four contrasts on the 198 passing tasks were fixed as exploratory sensitivities. Rates average training and planner repetitions within task, then tasks equally. Raw and the state reference have no training repetitions. Intervals follow Appendix B.4; the sensitivity separately resamples its 198 tasks. Table 13 preserves the original rates and all contrasts.

Table 13: Original PushT results. Rates are percentages averaged over three training runs per temporal cost and the indicated planner runs. Contrasts give points with marginal 95% paired intervals. J−D on all 200 tasks is primary; the other three comparisons are secondary. The 198-task replay sensitivity is exploratory. These outcomes are separate from the controlled-reset evaluations below.

Cost	Planner 42	Planner 2026	Pooled
Raw	36.00	36.50	36.25
D	44.83	43.67	44.25
N	42.17	40.00	41.08
J	27.67	29.50	28.58
P_{state}	13.00	16.50	14.75

Contrast	All 200 tasks	198-task sensitivity
$J - D$	-15.67 $[-20.00, -11.50]$	-15.74 $[-20.20, -11.45]$
$D -$ raw	$+8.00$ $[2.50, 13.67]$	$+8.25$ $[2.69, 13.64]$
$N - D$	-3.17 $[-7.08, 0.75]$	-3.20 $[-7.15, 0.76]$
$D - P_{\text{state}}$	$+29.50$ $[23.83, 35.25]$	$+29.71$ $[23.91, 35.35]$

Run-to-run variation. One later D rerun, training seed 42 and planner seed 42, does not reproduce the original rollouts exactly. Available paired first-plan action comparisons agree; this plan uses the logged start observation. Initial block states differ on 56 tasks, later actions on 38 of them, and five success flags change. Unseeded provisional resets and incompletely specified physical state are a plausible explanation. This variation motivated the fixed-reset evaluation below, which the main text uses for all tuned comparisons.

E.2 CONTROLLED RESETS AND FINITE CONFIGURATION SEARCH

The follow-up was fixed before its own outcomes and reuses the same 200 final tasks, including the two replay exceptions. Its 100 development tasks come from 3,519 eligible source episodes disjoint from cost training, pair validation, probe fitting and the original evaluation, and all pass logged-action replay.

Fixed initialization. Each development and final task receives one fixed native reset seed, shared across costs, configurations and repetitions. Canonical keys are ordered by split, episode, start row and goal row. PCG64 seeded with 20260923 assigns 300 distinct integers from $[0, 2^{31} - 1)$ without replacement. Reset precedes the unchanged native start/goal setters. Logged policy inputs, physics, preprocessing and planning rules remain unchanged. Actual initial states agree for each task across the corresponding cells. Differences from nominal dataset states do not filter or repair tasks.

Two independent linear-cost runs with training and planner seed 42 agree on initial states, actions, step counts and outcomes across all 200 tasks; one contributes to the reported rates.

Search and selection. Each family receives the same six learning-rate/weight-decay settings (Table 14). All other training settings, including the pair arrays, target, optimizer moments, batch size and 40-epoch budget, remain fixed. Every development configuration uses training seed 42 and planner seed 42 on the same 100 tasks. Selection takes, in order, the largest integer success count, smallest unrounded selected validation loss, original configuration if still tied, then smaller learning rate and weight decay. All 18 cells complete before automatic selection is committed. Thus temporal loss selects an epoch, while development planning selects a configuration.

The selected settings are $(3 \times 10^{-4}, 10^{-4})$ for D, $(10^{-3}, 10^{-2})$ for N, and $(3 \times 10^{-4}, 10^{-2})$ for J. All are nonoriginal. Each selected family uses training seeds 42, 2026 and 3072. All nine final heads are frozen before selected final planning. Table 15 gives all original and selected epochs and losses. Both neural costs have lower selected temporal validation loss than D in each corresponding training run, under both configuration choices.

Table 14: Complete PushT development search and later temporal loss check. Every configuration uses training seed 42 and planner seed 42 on the same 100 development tasks. L_{pool} is pair-validation loss from the cost-training episode pool. L_{new} uses the common separate-episode sample in Appendix E.4. Both losses are multiplied by 10^3. The dagger marks the configuration chosen by development success, and the original setting is learning rate 10^{-3} and weight decay 10^{-4}. New losses were measured afterwards and did not change any selection.

Cost	Learning rate	Weight decay	Successes	Epoch	$10^3 L_{\mathrm{pool}}$	$10^3 L_{\mathrm{new}}$
D†	3×10^{-4}	10^{-4}	47	40	1.905	2.058
D	3×10^{-4}	10^{-2}	45	40	1.906	2.058
D	10^{-3}	10^{-4}	43	40	1.830	1.989
D	10^{-3}	10^{-2}	44	40	1.833	1.991
D	3×10^{-3}	10^{-4}	44	35	1.868	2.012
D	3×10^{-3}	10^{-2}	44	35	1.875	2.019
N	3×10^{-4}	10^{-4}	40	40	1.002	1.217
N	3×10^{-4}	10^{-2}	39	40	1.002	1.217
N	10^{-3}	10^{-4}	39	38	0.940	1.218
N†	10^{-3}	10^{-2}	40	38	0.941	1.219
N	3×10^{-3}	10^{-4}	37	31	1.036	1.262
N	3×10^{-3}	10^{-2}	35	31	1.037	1.265
J	3×10^{-4}	10^{-4}	34	39	1.211	1.428
J†	3×10^{-4}	10^{-2}	36	39	1.210	1.427
J	10^{-3}	10^{-4}	28	39	1.193	1.408
J	10^{-3}	10^{-2}	29	39	1.193	1.408
J	3×10^{-3}	10^{-4}	29	28	1.183	1.393
J	3×10^{-3}	10^{-2}	31	28	1.179	1.380

Table 15: All eighteen original and selected PushT costs. The epoch is chosen using L_{pool}, the sample-weighted Smooth-L1 loss on pair validation from the cost-training episode pool. L_{new} is measured later on 10,000 pairs from 3,226 separate episodes, with no reselection. Losses and interval endpoints are multiplied by 10^3. Intervals are marginal, unadjusted 95% episode-cluster bootstrap intervals. Every training run completes forty epochs.

Cost	Training seed	Epoch	$10^3 L_{\mathrm{pool}}$	$10^3 L_{\mathrm{new}}$	New-loss 95% interval
Original configurations					
D	42	40	1.830	1.989	$[1.850, 2.137]$
D	2026	38	1.826	1.989	$[1.847, 2.138]$
D	3072	40	1.823	1.978	$[1.835, 2.128]$
N	42	38	0.940	1.218	$[1.117, 1.326]$
N	2026	40	0.943	1.201	$[1.099, 1.309]$
N	3072	40	0.941	1.168	$[1.069, 1.274]$
J	42	39	1.193	1.408	$[1.305, 1.515]$
J	2026	34	1.180	1.411	$[1.311, 1.514]$
J	3072	34	1.180	1.411	$[1.309, 1.517]$
Development-selected configurations					
D	42	40	1.905	2.058	$[1.913, 2.209]$
D	2026	39	1.901	2.056	$[1.910, 2.208]$
D	3072	40	1.904	2.058	$[1.911, 2.210]$
N	42	38	0.941	1.219	$[1.118, 1.328]$
N	2026	40	0.944	1.202	$[1.100, 1.309]$
N	3072	40	0.943	1.169	$[1.071, 1.275]$
J	42	39	1.210	1.427	$[1.328, 1.528]$
J	2026	40	1.210	1.369	$[1.275, 1.466]$
J	3072	40	1.197	1.433	$[1.333, 1.535]$

E.3 Fixed-reset comparisons and symmetrization

Fresh controls evaluate the original D/N/J heads with three training seeds and two planner seeds, and raw distance with both planner seeds. Selected configurations use the same coverage. Outcomes from the original evaluation are analyzed separately from the controlled-reset follow-up.

Deployment-only symmetrization uses the three original J heads:

$$c_{J,\mathrm{sym}}(z, g) = \frac{c_J(z, g) + c_J(g, z)}{2}.$$

Both planner seeds evaluate all 200 tasks. The comparator is original J under the same fixed resets; original D is a secondary comparator. No retraining or combination with the selected configuration is evaluated. Tables 16 and 17 report all eight cost variants and ten declared pooled contrasts.

Analysis. Training and planner repetitions are averaged within task, followed by equal task weights. Paired intervals follow Appendix B.4 and condition on the selected configurations and on one fixed reset per task, so they exclude variation across resets.

Table 16: Success under controlled initialization, with marginal 95% task-bootstrap intervals. Original configurations use the retained heads in fresh evaluations under fixed resets. Selected configurations use the finite development search. Symmetrization uses the original J heads. All rows use the same 200 final tasks and fixed resets.

Configuration	Cost	Success (%)	95% interval
Original	Raw	35.75	$[29.75, 42.00]$
	D	44.67	$[38.83, 50.42]$
	N	41.25	$[35.92, 46.50]$
	J	29.08	$[24.17, 33.92]$
Selected	D	46.08	$[40.33, 51.83]$
	N	42.00	$[36.67, 47.17]$
	J	33.75	$[29.00, 38.50]$
Symmetrized original	J_{sym}	37.58	$[32.50, 42.67]$

Table 17: All ten declared contrasts in the controlled-reset follow-up, in points with marginal, unadjusted 95% intervals. D/N/J denote original configurations evaluated under the fixed resets. Stars denote development-selected configurations. J_{sym} uses the original joint heads. The tuning primary is $J^\star - D^\star$, and the symmetrization primary is $J_{\mathrm{sym}} - J$. All other contrasts are secondary.

Comparison	Contrast	Difference	95% interval
Original controls	$J - D$	-15.58	$[-20.00, -11.25]$
	$N - D$	-3.42	$[-7.33, 0.58]$
	$D - \mathrm{raw}$	$+8.92$	$[3.50, 14.58]$
Selected configurations	$J^\star - D^\star$	-12.33	$[-16.92, -7.92]$
	$N^\star - D^\star$	-4.08	$[-7.92, -0.17]$
	$D^\star - D$	$+1.42$	$[-1.42, 4.25]$
	$N^\star - N$	$+0.75$	$[-1.33, 2.83]$
	$J^\star - J$	$+4.67$	$[1.67, 7.75]$
Symmetrization	$J_{\mathrm{sym}} - J$	$+8.50$	$[5.75, 11.33]$
	$J_{\mathrm{sym}} - D$	-7.08	$[-11.33, -2.83]$

E.4 Temporal loss on separate source episodes

This diagnostic was defined after the earlier planning results and before its new losses. Starting from the 3,519 eligible episodes used to construct the development task set, we exclude its 100 configuration-selection episodes. The remaining 3,419 episodes are disjoint from cost training, pair validation, probe training and development, and the original planning evaluation.

Using `default_rng(20260925)`, we draw 10,000 pairs with the original PushT sampler: uniform eligible episode, uniform gap from one to $\min(100, L-1)$, uniform valid start conditional on that gap, and equally likely endpoint orders. Sampling is with replacement. The sample represents 3,226 episodes. The same pairs evaluate all original and selected D/N/J instances at training seeds 42, 2026 and 3072.

Loss is sample-weighted Smooth-L1 with $\beta = 1$ against $|\Delta t|/224$. D/N use their mapped squared distance divided by 192, and J uses its unscaled output. Recomputed validation losses match the saved values for all eighteen heads.

For uncertainty, we resample the 3,226 represented episodes with replacement, carrying all sampled pairs from each episode together. Each resampled mean is the total loss divided by the total pair count. The same 10,000 episode draws, generated with `default_rng(42)`, serve every head and contrast. Linear percentile endpoints give marginal, unadjusted 95% intervals. Main contrasts average the three training runs equally within pair. Individual-run contrasts are secondary. The intervals condition on the frozen models, costs and sampled pairs within episodes, and exclude uncertainty from repeating training or configuration selection.

Table 15 gives all eighteen losses. Table 18 gives the four main and twelve individual-run contrasts. Every nonlinear-minus-linear and joint-minus-linear interval is below zero for both configuration choices. These are encoded-observation losses and do not identify the cause of the planning differences.

The same pairs evaluate all eighteen development configurations (Table 14). Every nonlinear and joint configuration has lower loss than every linear configuration, and these measurements change no epoch or configuration selection.

Table 18: All sixteen separate-episode PushT loss contrasts. Differences and marginal 95% interval endpoints are multiplied by 10^3. Each three-run mean is a main diagnostic contrast. Individual training-seed comparisons are secondary. Negative values favor the nonlinear or joint cost. Episode resampling preserves pairing across all costs and training runs.

Configuration	Training seed	N − D	J − D
Original	Three-run mean	$-0.789\ [-0.875, -0.709]$	$-0.575\ [-0.679, -0.478]$
	42	$-0.771\ [-0.860, -0.687]$	$-0.580\ [-0.686, -0.477]$
	2026	$-0.788\ [-0.877, -0.703]$	$-0.578\ [-0.691, -0.473]$
	3072	$-0.810\ [-0.898, -0.727]$	$-0.567\ [-0.675, -0.466]$
Selected	Three-run mean	$-0.860\ [-0.949, -0.775]$	$-0.647\ [-0.751, -0.553]$
	42	$-0.839\ [-0.930, -0.752]$	$-0.631\ [-0.733, -0.535]$
	2026	$-0.854\ [-0.945, -0.767]$	$-0.687\ [-0.791, -0.590]$
	3072	$-0.888\ [-0.981, -0.801]$	$-0.625\ [-0.734, -0.525]$

E.5 Covariance and shuffled-target controls

The controls of Appendix A.5 use all 200 existing final tasks under the same fixed resets and planner seeds 42 and 2026. Both learned linear costs use training seed 42. Comparators use the corresponding original-configuration linear and raw cells from the controlled-reset study. The original evaluation with unseeded resets is analyzed separately.

Table 19 reports every declared comparison, analyzed as in Appendix B.4, with planner seeds averaged within task. No task is removed.

Both controls perform below the temporal linear cost. Whitening has no clear difference from raw distance, while shuffled-target training performs below raw. This comparison tests one covariance estimator and one training-target permutation with real-label epoch selection.

Table 19: PushT linear controls on the same 200 tasks and fixed resets. Rates average planner seeds 42 and 2026. Both learned linear costs use training seed 42. All four contrasts are shown in points with marginal 95% paired intervals. W−D and $D_\pi - D$ are principal comparisons in this late exploratory study, with comparisons against raw distance secondary. The 45.00% linear rate uses a single training run and is separate from the broader aggregates.

Cost	Success (%)
Raw	35.75
Whitening W	31.75
Shuffled linear D_π	26.50
Temporal linear D	45.00

Contrast	Difference	95% interval
$W - D$	-13.25	$[-18.75, -7.75]$
$W - \text{raw}$	-4.00	$[-8.75, 0.75]$
$D_\pi - D$	-18.50	$[-24.75, -12.50]$
$D_\pi - \text{raw}$	-9.25	$[-14.75, -4.00]$

E.6 PER-GOAL RANKING OF LOGGED OBSERVATIONS

This late exploratory diagnostic asks how each cost orders observations relative to one goal. Of the 3,419 separate episodes of Appendix E.4, 2,534 contain at least 101 observations. We sample 500 of them, draw one goal per episode uniformly from the valid indices, and score the 100 preceding observations against that goal, with the observation first and the goal second. All eighteen core heads and raw distance score the same 50,000 pairs. Intervals treat the 500 source episodes as independent.

We compare each cost's ordering with two references: elapsed time to the goal, and a success-normalized state error $R = \max(e_{\text{pos}}/20,\ e_{\text{angle}}/(\pi/9))$, where e_{pos} combines agent and block position errors in pixels and e_{angle} is the symmetry-adjusted block angle error. $R < 1$ matches the success criterion, and its ordering outside that region is a diagnostic choice. Only 14% of observations within 25 steps of the goal satisfy $R < 1$, so short temporal gaps are not necessarily close to the goal state. For each goal, cost and window we compute Kendall's τ_b, which accounts for ties in both variables. Within gaps of 1–25 steps, the primary measure is concordance: the fraction of correctly ordered candidate pairs among those with unequal reference values. A cost tie receives half credit and a reference tie is excluded, using stored values without rounding. Every head has valid metrics for all 500 goals. Differences are formed per training run and averaged within goal. Intervals use 10,000 paired resamples of the 500 goal banks and are marginal and unadjusted.

The nonlinear and joint costs have lower temporal loss over all 100 gaps under both configuration choices. Within gaps of 1–25 steps, all four loss-difference intervals include zero, so the short-gap ranking advantage of the linear cost does not coincide with a clear loss difference within that window. Across all gaps, and within the 26–50 and 51–100 windows, the nonlinear cost has higher temporal Kendall correlation than the linear cost. It has lower state-error correlation in every window, as does the joint scorer. These are secondary comparisons. An additional exploratory comparison gives a tuned-minus-untuned change in all-gap state-error correlation of $0.012\ [0.007, 0.017]$ for the joint scorer, $0.002\ [-0.001, 0.004]$ for the linear cost and $0.0000\ [-0.0001, 0.0001]$ for the nonlinear cost, using the same paired goal-bank resampling.

Raw distance has higher short-gap concordance than the linear cost against both references. Its mean state-error correlation is also higher in the 26–50 window, while falling below the linear cost over all gaps and in the 51–100 window. Over all gaps, the joint scorer's state-error correlation also exceeds that of raw distance despite its lower planning success. The ranking diagnostic therefore does not reproduce the complete ordering of online planning success.

The lowest-cost observation in each bank shows how a cost chooses among the logged candidates. Its regret is its state error R minus the smallest R available in that bank, averaged over exact minimum-cost ties. Over gaps of 1–100 steps, mean regret is 0.0056 for raw distance. Before and after tuning, it is 0.0056 and 0.0060 for the linear cost, 0.0064 and 0.0065 for the nonlinear cost, and 0.0275 and 0.0484 for the joint scorer. These means do not reproduce the planning order. Raw distance has a mean regret close to the linear cost, and tuning increases joint regret while improving its planning success.

Table 20: The nonlinear cost orders logged observations by elapsed time more accurately than the linear cost, but by state error less accurately. Entries are mean per-goal Kendall τ_b over 500 goals and three training runs. The lower panel gives the four primary comparisons and selected secondary contrasts with 95% paired goal-bank intervals. Loss differences are multiplied by 10^3.

			Untuned			Tuned		
Reference	Gap	Raw	D	N	J	D	N	J
Elapsed time	1–100	0.533	0.834	0.869	0.823	0.831	0.869	0.829
	1–25	0.883	0.867	0.858	0.816	0.868	0.858	0.841
	26–50	0.391	0.595	0.657	0.565	0.590	0.658	0.569
	51–100	0.156	0.706	0.790	0.719	0.702	0.789	0.725
State error R	1–100	0.526	0.629	0.604	0.569	0.631	0.604	0.581
	1–25	0.812	0.796	0.785	0.749	0.801	0.785	0.773
	26–50	0.462	0.436	0.401	0.322	0.437	0.401	0.335
	51–100	0.256	0.404	0.363	0.320	0.407	0.364	0.333

Measure	Gap	Contrast	Untuned	Tuned
Concordance, R (primary)	1–25	N−D	-0.006 $[-0.010, -0.001]$	-0.008 $[-0.013, -0.004]$
		J−D	-0.024 $[-0.030, -0.017]$	-0.014 $[-0.019, -0.009]$
Concordance, R	1–25	D−raw	-0.008 $[-0.012, -0.003]$	-0.005 $[-0.009, -0.001]$
Concordance, time	1–25	D−raw	-0.008 $[-0.013, -0.003]$	-0.007 $[-0.013, -0.002]$
τ_b, time	1–100	N−D	$+0.035$ $[0.025, 0.044]$	$+0.037$ $[0.028, 0.047]$
		J−D	-0.0108 $[-0.0217, -0.0002]$	-0.0025 $[-0.0126, 0.0076]$
τ_b, state error	1–100	N−D	-0.025 $[-0.034, -0.016]$	-0.026 $[-0.036, -0.018]$
		J−D	-0.060 $[-0.070, -0.049]$	-0.050 $[-0.060, -0.040]$
Loss $\times 10^3$	1–100	N−D	-0.90 $[-1.09, -0.72]$	-0.99 $[-1.18, -0.80]$
		J−D	-0.66 $[-0.88, -0.45]$	-0.78 $[-1.01, -0.57]$
	1–25	N−D	-0.05 $[-0.19, 0.09]$	-0.11 $[-0.25, 0.04]$
		J−D	$+0.15$ $[-0.06, 0.37]$	$+0.08$ $[-0.09, 0.26]$

F Ordering and Representation Diagnostics

F.1 Logged observations and the illustrated goal

The observation bank contains 8,000 frames, 20 from each of the 400 source episodes excluded from original cost training and pair validation. Frames are sampled without replacement in sorted episode order using `default_rng(42)`. These episodes supplied earlier planning evaluations. Each of the three frozen LeWM models encodes the same observations with its own encoder. Costs use the original training-seed-42 D/N/J instances and each model's frozen position probe, evaluated as deployed with observation first and goal second. These are encoded observations, rather than predicted endpoints.

Figure 4 uses world model 42 and fixed illustration goal B: episode 241, row 22137, at $(174.9, 130.8)$ pixels. It is the first task in the earlier opposite-room manifest with $|x_{g,1} - 112| \geq 16$, chosen by that geometry rule. Colors are within-panel percentiles $100(\mathrm{rank} - 1)/7999$, using average ranks for ties. They describe sampled orderings rather than equal cost increments or fitted contours. The illustration uses one model and one trained instance of each cost, and its goal lies outside the 399 goals summarized in Appendix F.2.

F.2 Doorway ordering across 399 goals

For physical position $x = (x_1, x_2)$ and goal x_g, let $r(x) = \mathbf{1}\{x_1 > 112\}$ and $o = (112, 49)$. The doorway distance estimate is

$$d(x, x_g) = \begin{cases} \|x - x_g\|_2, & r(x) = r(x_g), \\ \|x - o\|_2 + \|x_g - o\|_2, & r(x) \neq r(x_g). \end{cases} \tag{7}$$

It is a geometric estimate in pixels, not an exact shortest-path or travel-time target. Write $e_i = \|x_i - x_g\|_2$ and $d_i = d(x_i, x_g)$. For each goal, eligible pairs (i, j) have i in the opposite room, j in

the goal room, $e_i < e_j$ and $d_i > d_j$. Geometry alone selects these pairs before costs are read. For deployed costs c_i, c_j, concordance is

$$Q = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left[\mathbf{1}(c_i > c_j) + \tfrac{1}{2}\mathbf{1}(c_i = c_j)\right]. \tag{8}$$

The doorway estimate orders all selected pairs with $Q = 1$ and Euclidean distance with $Q = 0$. Pairs share observations and are not treated as independent sampling units.

The goal pool contains the 399 of 400 cached original task-goal observations satisfying $|x_{g,1} - 112| \geq 16$. All three models use the same goals and observation identities. Illustration goal B is outside this pool. For each goal and cost, we compute Q on all eligible pairs and Spearman correlation ρ between positive cost and the doorway estimate over opposite-room observations, using average ranks for ties. All quantities are defined for every goal, cost and model. Table 21 gives equal-goal medians and interquartile ranges; its lower panel summarizes paired D-minus-position differences. These are descriptive distributions, with no bootstrap or confidence intervals.

D has higher Q than the position cost for 379, 359 and 336 of the 399 goals in models 42, 2026 and 3072. The position cost performs better for some goals, and the selected-pair statistic does not measure global ranking quality. For illustration goal B, the position cost has higher all-observation correlation with the doorway estimate than D, 0.986 versus 0.967. Neither the correlations nor concordance identify which ordering changes cause online success.

Table 21: Ordering over all 399 eligible goals. Entries give medians and interquartile ranges across goals, not confidence intervals. The upper panel uses one original trained instance per cost/model. The lower panel summarizes each goal's paired D-minus-position difference; the last column counts strictly positive differences. All statistics are defined for every goal.

World model	Cost	Opposite-room ρ	Detour concordance Q
42	Raw	$0.708\ [0.672, 0.733]$	$0.324\ [0.265, 0.409]$
	D	$0.950\ [0.931, 0.969]$	$0.675\ [0.550, 0.779]$
	N	$0.945\ [0.920, 0.962]$	$0.586\ [0.408, 0.702]$
	J	$0.945\ [0.921, 0.957]$	$0.693\ [0.518, 0.819]$
	P_{XY}	$0.810\ [0.666, 0.928]$	$0.151\ [0.091, 0.266]$
2026	Raw	$0.753\ [0.685, 0.769]$	$0.420\ [0.333, 0.507]$
	D	$0.954\ [0.936, 0.964]$	$0.685\ [0.538, 0.779]$
	N	$0.963\ [0.942, 0.970]$	$0.693\ [0.503, 0.798]$
	J	$0.944\ [0.929, 0.953]$	$0.629\ [0.497, 0.735]$
	P_{XY}	$0.786\ [0.672, 0.919]$	$0.195\ [0.094, 0.317]$
3072	Raw	$0.733\ [0.682, 0.790]$	$0.292\ [0.250, 0.435]$
	D	$0.961\ [0.947, 0.969]$	$0.556\ [0.411, 0.817]$
	N	$0.958\ [0.938, 0.967]$	$0.602\ [0.447, 0.764]$
	J	$0.962\ [0.946, 0.970]$	$0.650\ [0.508, 0.767]$
	P_{XY}	$0.794\ [0.662, 0.928]$	$0.183\ [0.107, 0.324]$

World model	Difference	Median [Q1, Q3]	Positive goals
42	$\rho_D - \rho_{P_{\mathrm{XY}}}$	$0.152\ [0.031, 0.253]$	344/399
	$Q_D - Q_{P_{\mathrm{XY}}}$	$0.479\ [0.344, 0.624]$	379/399
2026	$\rho_D - \rho_{P_{\mathrm{XY}}}$	$0.153\ [0.035, 0.262]$	343/399
	$Q_D - Q_{P_{\mathrm{XY}}}$	$0.459\ [0.199, 0.580]$	359/399
3072	$\rho_D - \rho_{P_{\mathrm{XY}}}$	$0.163\ [0.030, 0.274]$	326/399
	$Q_D - Q_{P_{\mathrm{XY}}}$	$0.390\ [0.128, 0.552]$	336/399

F.3 Position decoding and linear-map structure

The original position probes use every available cached observation from the 9,600 eligible episodes once, with the least-squares procedure of Appendix A. There is no feature rescaling or held-out selection. The model-3072 cache omits one eligible episode. Each probe is evaluated on one start+49

goal frame from each of the 400 excluded episodes, with equal weights. The episodes were previously used for planning evaluations; these errors measure goal decoding, not accuracy on arbitrary or predicted rollout endpoints. Table 22 gives training coverage and all three models' errors. These are the original probes, separate from the balanced-study refits.

For each original linear map, define $M_D = A^\top A$ or $M_F = (W_2W_1)^\top(W_2W_1)$ and normalize its eigenvalues as $p_i = \lambda_i / \sum_j \lambda_j$. Entropy effective rank is $\exp(-\sum_i p_i \log p_i)$ and participation rank is $1/\sum_i p_i^2$. Products and decompositions use float64; numerical rank uses a relative singular-value threshold of 10^{-6}. All 18 maps have numerical rank 192. Their trace fractions and effective ranks describe concentration rather than a rank constraint or a measure of planning performance.

Let V_B be an orthonormal basis for the rank-two rowspace of B. Its metric trace weight is $\eta = \mathrm{tr}(V_B V_B^\top M)/\mathrm{tr}(M)$. A random rank-two subspace has expected weight $2/192 = 1.04\%$. The original comparison uses 10,000 Gaussian-QR subspaces per model at seed 42, shared across its six maps. Every observed η exceeds all its reference draws. F has larger η than D in all nine paired fits, although the original F−D planning interval includes zero. This is a measure of matrix weight in decodable directions, not a fraction of planning success explained by position. Table 22 summarizes the original maps without refitting.

Table 22: Original position probes and linear-map summaries. The upper panel evaluates each probe on the same 400 excluded source-episode goal frames. Coordinate entries are ordered (x, y); errors are pixels. The lower panel gives the minimum and maximum across all nine original fits per linear construction, not confidence intervals. Trace fractions are percentages, including the probe rowspace weight η. Every map has numerical rank 192.

Model	Training frames	Training episodes	R^2 (x, y)	RMSE (x, y)	Mean position error
42	190,691	9,600	$(0.9965, 0.9935)$	$(1.75, 2.64)$	2.73
2026	190,866	9,600	$(0.9934, 0.9926)$	$(2.38, 2.82)$	3.17
3072	190,767	9,599	$(0.9976, 0.9920)$	$(1.43, 2.92)$	2.78

Map	Entropy rank	Participation rank	Top 2 trace (%)	Top 10 trace (%)	η (%)
F	16.12–24.58	4.84–7.97	40.9–52.9	68.1–75.5	12.91–21.74
D	20.98–30.66	6.45–10.24	36.3–46.7	62.9–71.2	10.92–17.12

G Earlier Quadratic Comparison

An earlier version of the linear cost used a trace-normalized quadratic. With $\delta = z - g$ and $d = 192$,

$$q_M(z, g) = \delta^\top M \delta, \qquad M = \frac{d\,(LL^\top + 10^{-8} I)}{\mathrm{tr}(LL^\top + 10^{-8} I)}, \tag{9}$$

where L is lower triangular and M is initialized at the identity. Training regresses

$$\widetilde{c}_M(z, g) = \frac{\mathrm{softplus}(a)\sqrt{q_M(z, g) + 10^{-8}} + b}{224} \tag{10}$$

onto $|\Delta t|/224$, and planning uses q_M. It was compared with a shared nonlinear embedding and a joint scorer (earlier TRM) with the architectures of Appendix A.1. All three used the original training and validation pairs, AdamW with learning rate 10^{-3} and weight decay 10^{-4}, batch size 1,024 and Smooth-L1 loss with $\beta = 1$, but only 20 epochs and one trained instance per world model. Full SPD selected epoch 20, its final epoch, for all three models; the joint scorer selected epochs 7, 7 and 9.

On the original 200 tasks with three world models and three planner seeds, full SPD, the embedding and the joint scorer reach 89.83%, 97.00% and 97.67% success. The predeclared primary contrast, embedding minus full SPD, is $+7.17$ $[4.83, 9.61]$ points; joint minus full SPD is $+7.83$ $[5.33, 10.50]$ and joint minus embedding is $+0.67$ $[-0.44, 1.89]$.

Full SPD and the direct linear cost define the same family of quadratic costs, up to a scale that leaves the planner's ranking unchanged. They differ in parameterization, initialization, training readout and budget, and full SPD selected its final epoch for every world model, which is consistent with a schedule too short for it to converge. Adding the readout of Equation 10 to the direct map does not clearly change planning (Appendix A.4). Under the common procedure of Section 3.3, the joint scorer's advantage over the linear cost on the same tasks shrinks from about eight points to under one point, with an interval that includes zero (Appendix C.3). The earlier gap therefore reflects how the linear cost was built and trained, which is why the main comparison trains every family under the same procedure.