LATENT METRICS FOR PLANNING WITH WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint-Embedding Predictive Architectures (JEPAs) enable visual planning by predicting the future outcomes of candidate actions directly in latent space. However, existing JEPA planners typically evaluate candidate actions using Euclidean distance between the predicted outcome and the goal. This treats all latent directions equally, despite differences in their sensitivity to actions and in the model's prediction accuracy along them. We introduce Latent Metrics for Planning with World Models (LENS). LENS identifies the *action-response fidelity* of different latent directions from the action-conditioned dynamics already learned by a pretrained world model, while keeping both the encoder and predictor frozen. Specifically, LENS derives action-response fidelity from the consistency of action-conditioned variation between real and predicted futures together with the accuracy of the corresponding future predictions, and uses this fidelity to determine the relative influence of different latent directions in planning. Directions with higher action-response fidelity receive greater relative weight in the planning metric. The resulting metric is used both to evaluate predicted futures and to guide sampling-based search, without retraining the world model. Across four benchmarks, LENS improves average success rate from 85.8% to 95.8%. With candidate actions and world-model predictions fixed, it increases the average Spearman correlation between model rankings and real outcomes from 0.540 to 0.696. These results show that the action-conditioned dynamics already learned by a world model can provide not only a basis for predicting the future, but also a basis for determining the relative role of different latent directions in planning.

1 INTRODUCTION

Action-conditioned world models learn how actions change the future, enabling an agent to evaluate the possible consequences of different behaviors before executing them. Given a current observation and a candidate action sequence, a world model predicts the future state induced by that sequence and uses these predictions for goal-conditioned control and model-based planning (Ha & Schmidhuber, 2018; Hafner et al., 2019; Hansen et al., 2022). Recent work has increasingly moved toward reconstruction-free latent prediction based on the Joint-Embedding Predictive Architecture (JEPA), where future representations are predicted directly without reconstructing high-dimensional observations (Assran et al., 2023; Bardes et al., 2024). This makes action-conditioned latent dynamics an attractive foundation for visual planning (Sobal et al., 2025; Assran et al., 2025).

Existing latent-space planners commonly evaluate candidate actions using Euclidean distance between the predicted future and the goal (Zhou et al., 2025; Maes et al., 2026). This choice is natural: when there is no prior information indicating that one latent direction should be more important than another, the identity metric I provides an isotropic, task-agnostic default comparison rule, allowing predicted futures and visual goals to be compared directly in the same latent space without introducing an additional notion of relevance. Once Euclidean distance is used for planning, however, I is no longer merely a neutral default; it implicitly specifies how different latent directions should influence action selection. For

$$J_I(A) = \|\hat{z}_{t+H}(A) - z_g\|_2^2,$$

the Euclidean metric applies the same unit scale to every latent direction: changes of the same magnitude incur the same distance cost regardless of the direction in which they occur.

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.

Such an isotropic comparison rule is not obviously appropriate for an action-conditioned world model. Unlike a vector space with no structural information, an action-conditioned world model has already learned how actions change the future, and both the relation between latent directions and action-conditioned dynamics and the model's prediction accuracy can vary across directions. Some directions may be only weakly associated with the future differences produced by candidate actions, whereas others may closely reflect contact, object displacement, or other physical changes that distinguish candidate plans. Prediction accuracy can likewise vary across directions. Applying the same unit scale to all of them therefore ignores structure already present in the learned dynamics and predictions.

A natural approach is to use additional planning information to learn or shape a non-Euclidean metric. Downstream task losses, privileged task states, or temporal and reachability structure can, for example, indicate which latent differences are more useful for control (Bansal et al., 2023; Li et al., 2026b; Bai & Xiong, 2026; Hu et al., 2026). These signals can indeed produce non-uniform planning weights, but they also introduce limitations. Task-aware supervision requires additional task semantics and therefore weakens the advantage of latent world models that can plan without task-specific rewards or values. Temporal and reachability signals can remain reward-free, but they primarily define relevance through relationships between states in observed trajectories, which remain an indirect proxy for the future differences induced by candidate actions. More fundamentally, these approaches specify what matters for planning through external task semantics or trajectory structure rather than directly through the *action-conditioned dynamics that generate the differences among candidate futures*.

This distinction is particularly important for action-conditioned world models. Predictive training aims to predict the latent future state given the current state and actions, whereas deployment requires the planner to select among candidate actions. The former learns how actions map to future states, but does not specify how those predicted futures should be compared or guarantee that the predictive latent geometry is naturally suited to candidate ordering. We refer to this unresolved transition from *future prediction* to *action selection* as the Prediction–Planning Gap. Standard Euclidean planning defaults to the identity metric I for this comparison, implicitly assigning the same comparison scale to different latent directions.

Actions themselves provide a more direct signal for addressing this problem. Differences among candidate futures arise from differences among candidate actions. If a latent direction consistently reflects candidate-action variation in both real futures and model predictions, and the world model accurately predicts the corresponding future states, then that direction provides more reliable evidence for distinguishing candidate actions. We refer to the quality with which the world model predicts real action-conditioned dynamics across latent directions as *action-response fidelity*. It reflects both the consistency of action-conditioned variation between real and predicted futures and the accuracy of the corresponding future predictions. Accordingly, different latent directions need not be compared under the same unit scale; their relative influence should instead reflect action-response fidelity.

Based on this observation, we introduce Latent Metrics for Planning with World Models (LENS). LENS uses action-response fidelity to construct a latent planning metric that determines the relative influence of different latent directions on candidate comparison. The resulting metric is used not only to compare predicted futures with the goal, but also as a unified comparison scale for guiding finite-budget sampling-based search. LENS can be applied directly to an existing world model while keeping both its encoder and action-conditioned predictor frozen, without retraining the predictive model.

Across four visual goal-conditioned control environments, LENS achieves an average closed-loop success rate of 95.8%. In a controlled experiment, we fix the candidate actions and all world-model predictions and change only the metric used to compare predicted futures; LENS increases the average Spearman correlation between model-based rankings and real task outcomes from 0.540 to 0.696. LENS also transfers across multiple frozen world-model backbones, showing that the same metric-construction principle can be applied beyond a single predictive architecture.

Our main contributions are:

- **Prediction–Planning Gap.** We identify an unresolved transition from the future-prediction objective to action selection: world-model training predicts the futures produced by candidate actions, but does not determine how those futures should be compared for planning.
- **LENS.** We introduce *action-response fidelity*, which characterizes the quality of action-conditioned prediction across latent directions through the consistency of real and predicted action-conditioned variation together with the accuracy of the corresponding future predictions, and use it to construct a planning metric.
- **Empirical validation.** LENS reaches 95.8% average success across four visual control environments, improves candidate-ordering Spearman correlation by $+0.156$ under fixed predictions, and transfers across multiple world-model backbones.

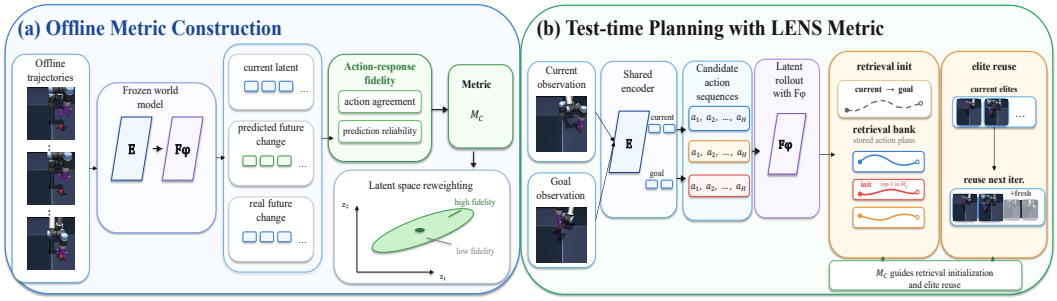


Figure 1: Overview of LENS. With the world model frozen, LENS constructs M_C from offline dynamics and uses it for candidate evaluation and search; the fidelity expression is schematic, with the exact construction given in Section 4.1.

2 Related Work

Latent world models for planning. Action-conditioned world models support decision making by predicting future outcomes under candidate actions, and reconstruction-free approaches perform this prediction directly in latent space. DINO-WM learns action-conditioned dynamics over frozen DINOv2 features (Zhou et al., 2025; Oquab et al., 2024); PLDM jointly learns an encoder and a latent dynamics model for goal-conditioned planning (Sobal et al., 2025); V-JEPA 2-AC adapts large-scale video representations into action-conditioned dynamics for robot planning (Assran et al., 2025); and LeWorldModel (LeWM) combines next-embedding prediction with SIGReg to enable end-to-end joint-embedding world modeling from pixels (Maes et al., 2026). These methods establish the practical utility of action-conditioned latent prediction as a basis for planning. LENS further uses the action-conditioned dynamics already learned by such models to determine the relative influence of different latent directions on planning distance, thereby improving candidate comparison over predicted futures.

Planning-aware representation learning. Planning-aware representation learning modifies the training process so that latent structure better supports downstream control. RC-aux combines multi-horizon prediction with budget-conditioned reachability supervision (Li et al., 2026b); Temporal-Distance JEPA captures directional temporal progress (Bai & Xiong, 2026); SCALE calibrates latent distances using task-state supervision (Hu et al., 2026); and DA-LeWM improves decision-metric alignment through additional action-conditioned training objectives (Wang et al., 2026a). More broadly, TaskMet, temporal-distance methods, value-shaped representations, and quasimetric learning all indicate that distances useful for decision making need not coincide with Euclidean proximity in the original representation (Bansal et al., 2023; Destrade et al., 2025; Wang et al., 2026b; 2023). These approaches primarily use additional information such as task semantics, temporal structure, or reachability to shape planning-aware geometry during representation or model learning. LENS instead derives action-response fidelity from the consistency of action-conditioned variation between real and predicted futures together with the accuracy of the corresponding future predictions, and constructs the planning metric from this quantity.

Planner adaptation for frozen world models. Another line of work keeps the predictive model fixed and adapts downstream planning. The Objective Is the Bottleneck shows that replacing

the latent planning objective can substantially improve planning without retraining the world model (Singh, 2026); TRM learns a terminal cost from temporal structure (Li et al., 2026a); PRISM learns a state-conditioned action prior (Wang et al., 2026c); and IMWM combines retrieval initialization with adaptive guidance (Gao et al., 2026). Related work also uses latent subgoals to improve candidate generation (Zhang et al., 2026). LENS differs in that it does not require training an additional independent scorer, prior, or optimizer. Instead, it identifies the action-response fidelity of different latent directions from the world model's action-conditioned dynamics and uses the resulting planning metric for both candidate comparison and search guidance.

3 PROBLEM

Action-conditioned world models provide a basis for prediction-based planning by learning how actions change the future. Given a current state and candidate action sequences, the model predicts the future states that different actions may reach, after which the planner compares the candidates and selects an action sequence.

Let E denote the visual encoder and $F^\star$ the trained and frozen action-conditioned predictor. Let the current and goal observations in latent space be $z_t = E(o_t)$ and $z_g = E(o_g)$, respectively. For a candidate action sequence A of horizon H, the predicted future endpoint is

$$\hat{z}_{t+H}(A) = F^\star(z_t, A). \tag{1}$$

Action-conditioned world models typically learn the mapping among the current state, actions, and latent future state through a predictive objective such as

$$\mathcal{L}_{\text{pred}} = \mathbb{E}\left[\ell\left(F^\star(z_t, A), z_{t+H}\right)\right]. \tag{2}$$

Predictive training asks whether the model can predict the latent future state given the current state and a candidate action sequence. During planning, these predictions must additionally be used to distinguish among candidate actions. Standard latent-space planning directly selects the action sequence whose predicted endpoint minimizes Euclidean distance to the goal:

$$A_I^\star = \arg\min_A \|F^\star(z_t, A) - z_g\|_2^2. \tag{3}$$

Equation 3 is equivalent to using the identity metric I in latent space. The identity metric assigns the same unit scale to every latent direction and therefore imposes a uniform comparison rule.

The purpose of using a world model for planning, however, is to rank candidate actions according to their predicted outcomes. Some latent directions may only weakly reflect the differences among candidate futures, whereas others may more directly reflect how actions change the future. Because planning relies on model predictions, prediction accuracy can also differ across directions. A uniform metric cannot represent these differences in action-conditioned structure and prediction reliability.

We refer to this unresolved transition from the future-prediction objective to the action-selection objective as the Prediction–Planning Gap. World-model training learns which latent future state follows a candidate action sequence, but does not determine how different predicted futures should be compared. Standard latent planning defaults to the identity metric I for this comparison, whose uniform weighting ignores differences in action-conditioned structure and prediction accuracy across latent directions. This work asks whether, while keeping both the visual encoder and action-conditioned predictor frozen, one can identify action-response fidelity directly from the learned dynamics and use it to construct a planning metric M_C better suited to candidate comparison.

4 METHOD

Action-conditioned world models learn how actions change the latent state by predicting the futures produced by different candidate actions. For planning, these dynamics are not uniform across latent directions: some directions more directly reflect future variation associated with candidate actions, whereas others are less useful for distinguishing among candidates. LENS characterizes this difference through *action-response fidelity*, which reflects both the consistency of action-conditioned

variation between real and predicted futures and the accuracy with which the world model predicts the corresponding future states. LENS uses this quantity to determine the relative role of different latent directions in candidate comparison and to construct the planning metric M_C, thereby turning the action-conditioned structure already present in the frozen predictive latent space into a non-uniform comparison rule for planning.

4.1 Action-Response Fidelity

Let the real and predicted latent future changes be

$$\Delta z_t = z_{t+H} - z_t, \qquad \Delta \hat{z}_t = \hat{z}_{t+H} - z_t, \qquad \hat{z}_{t+H} = F^\star(z_t, A). \tag{4}$$

Both actions and their future changes depend on information available before the candidate action sequence is applied. We define the pre-action context c_t as the concatenation of the latent history used by the current predictor and its corresponding previous action-block history:

$$c_t = \left[\mathrm{vec}(z_{t-L+1:t}), \mathrm{vec}(a_{t-L+1:t-1})\right]. \tag{5}$$

The environment-specific history length and implementation details are provided in Appendix A.1.

Because the pre-action context influences both the behavior distribution in the data and the subsequent latent change, directly estimating an action–future relation from offline trajectories can mix statistical correlation induced by different historical contexts with the future variation associated with action variation under the same pre-action conditions (Shi et al., 2026). To reduce this effect, for any variable X we remove the component predictable from c_t:

$$\widetilde{X} = X - \mathbb{E}[X \mid c_t]. \tag{6}$$

We apply this residualization to the candidate action sequence and to both Δz_t and $\Delta \hat{z}_t$. We further whiten the action residual:

$$\bar{A} = \left[\mathrm{Cov}(\widetilde{A}) + \varepsilon_A I\right]^{-1/2} \widetilde{A}, \tag{7}$$

where $\varepsilon_A > 0$ ensures positive definiteness under finite samples. The subsequent statistics therefore primarily reflect the relationship between action variation and latent future change after conditioning on the pre-action context, rather than correlations induced by differences in historical context or by the marginal action distribution across offline trajectories.

We define the real and predicted action–future cross-covariances as

$$C = \mathbb{E}\left[\widetilde{\Delta z_t}\, \bar{A}^\top\right], \qquad \hat{C} = \mathbb{E}\left[\widetilde{\Delta \hat{z}_t}\, \bar{A}^\top\right]. \tag{8}$$

For any latent direction u, $C^\top u$ and $\hat{C}^\top u$ describe how action variation is associated with real and predicted future change along that direction. Action-response fidelity is high when the world model both preserves the action-conditioned variation observed in real futures and accurately predicts the corresponding candidate futures.

To characterize future-prediction accuracy, we define the regularized prediction-error second moment

$$\Sigma_\epsilon = \mathbb{E}\left[(\Delta z_t - \Delta \hat{z}_t)\,(\Delta z_t - \Delta \hat{z}_t)^\top\right] + \varepsilon_\epsilon I, \tag{9}$$

where $\varepsilon_\epsilon > 0$ ensures positive definiteness under finite samples.

To characterize the consistency of action-conditioned variation, we consider the symmetric coupling between C and $\hat{C}$. Its principal modes satisfy

$$\frac{1}{2}\left(C\hat{C}^\top + \hat{C}C^\top\right) u_i = \lambda_i u_i, \tag{10}$$

with the standard orthonormalization $u_i^\top u_j = \delta_{ij}$. A larger positive λ_i indicates stronger agreement between real and predicted action-conditioned variation along mode u_i.

We then define the action-response fidelity of each mode as

$$\mathcal{F}(u_i) = \left[1 + B \log\left(1 + \max(\lambda_i, 0)\right)\right] u_i^\top \Sigma_\epsilon^{-1} u_i. \tag{11}$$

The spectral term increases with real–predicted action-conditioned agreement, while $B > 0$ controls the strength of this emphasis. The precision term $u_i^\top \Sigma_\epsilon^{-1} u_i$ reflects the prediction reliability associated with the same mode. Their product therefore determines the relative planning weight assigned to u_i.

4.2 Planning Metric

Equation 11 determines how much relative planning weight each mode should receive. To specify where this weight acts after accounting for prediction error, we define the prediction-error-adjusted mode projector

$$P_i = \frac{\left(\Sigma_\epsilon^{-1/2} u_i\right)\left(\Sigma_\epsilon^{-1/2} u_i\right)^\top}{u_i^\top \Sigma_\epsilon^{-1} u_i}. \tag{12}$$

The matrix P_i is the rank-one projector onto the direction $\Sigma_\epsilon^{-1/2} u_i$, specifying where $\mathcal{F}(u_i)$ acts. LENS then constructs the full-rank planning metric as

$$M_C \propto \sum_{i=1}^{d} \mathcal{F}(u_i) P_i. \tag{13}$$

We normalize the trace of M_C to the latent dimension d, fixing its global scale so that the metric changes the relative influence of latent directions rather than introducing an arbitrary overall rescaling of planning distance. Eqs. (11)–(13) separate the two roles clearly: $\mathcal{F}(u_i)$ determines how strongly each mode should influence planning, while P_i determines where that influence is applied in the prediction-error-adjusted latent geometry. Thus, M_C neither changes the future predicted by the world model nor relearns the latent representation. It changes how the planner compares these predictions: latent directions with higher action-response fidelity exert greater relative influence on candidate distance.

After obtaining M_C, LENS directly replaces the identity metric used in standard latent planning:

$$A^\star = \arg\min_A \left(F^\star(z_t, A) - z_g\right)^\top M_C \left(F^\star(z_t, A) - z_g\right). \tag{14}$$

Compared with the Euclidean planning objective in equation 3, the encoder, predictor, candidate actions, and predicted futures all remain unchanged. The only difference is the relative contribution of different latent directions to candidate distance: Euclidean distance compares all directions using the same unit weight, whereas LENS determines their relative influence on candidate comparison according to action-response fidelity.

4.3 Planning with M_C

The metric M_C is used not only to compare a predicted future with the goal, but also at other points in the planning process that require measuring latent proximity. Because M_C has already determined the relative role of different latent directions in planning, the same metric naturally determines which reference trajectories are close to the current start–goal pair and which low-cost candidates reach similar predicted futures.

If the offline data contain trajectories that are close to the current start–goal pair under M_C, these local samples can provide action information relevant to the current problem. For the current pair (z_t, z_g) and reference trajectory i with start and goal $(z_{s,i}, z_{g,i})$, their distance is

$$D_i = (z_t - z_{s,i})^\top M_C (z_t - z_{s,i}) + (z_g - z_{g,i})^\top M_C (z_g - z_{g,i}). \tag{15}$$

Nearby reference trajectories center part of the iteration-0 CEM proposal (Rubinstein & Kroese, 2004; Gao et al., 2026), directing the finite sampling budget toward more relevant action regions. As CEM proceeds, the current candidates themselves provide local information specific to the current problem. For two low-cost candidates with predicted futures $\hat{z}_i$ and $\hat{z}_j$, we compare

$$D_{ij}^{\text{pred}} = (\hat{z}_i - \hat{z}_j)^\top M_C (\hat{z}_i - \hat{z}_j). \tag{16}$$

When multiple low-cost candidates reach nearby predicted futures under M_C, LENS retains a small number of the corresponding complete action sequences for the next CEM iteration, while the remaining budget continues to sample new candidates (Pinneri et al., 2021). This preserves low-cost action modes already discovered during the current planning process while leaving most of the sampling budget available to explore new candidate regions. From candidate evaluation to reference-trajectory matching and candidate reuse during CEM iterations, LENS therefore uses the same M_C throughout.

5 Experiments

5.1 Experimental Setup

Environments. We follow the LeWM evaluation setup, including its offline datasets, observation preprocessing, and goal-conditioned planning protocol (Maes et al., 2026). Experiments cover TwoRoom (visual navigation), PushT (T-block manipulation) (Chi et al., 2023), Reacher (end-effector reaching) (Tassa et al., 2018), and OGBench-Cube (block pushing) (Park et al., 2025).

Baselines. We group existing methods according to where they modify world-model planning. Predictive World Models include PLDM, DINO-WM, and LeWorldModel; Training-time Planning Adaptation includes RC-aux, Temporal-Distance JEPA, SCALE, and Traj-LeWM (Huang et al., 2026), which introduce additional planning-oriented supervision or objectives during world-model training; and Frozen-World-Model Planning Adaptation includes IMWM, which keeps the predictive backbone fixed and adapts the planner.

Planning protocol. We follow the LeWM CEM-based receding-horizon planning protocol (Maes et al., 2026), while keeping the pretrained encoder and dynamics predictor frozen. Action-response fidelity, M_C, and the retrieval bank are constructed solely from offline trajectories before evaluation; evaluation rollouts are never used to estimate or update these components. The planning population, optimization horizon, and evaluation task set remain unchanged when comparing the Euclidean planner with LENS, so the resulting differences isolate the effect of the planning metric and the metric-guided search components rather than changes to the predictive model or evaluation budget. Full planning and evaluation details are provided in Appendices B and C.

5.2 Planning Performance

Table 1 compares LENS with existing predictive world models and planning-oriented extensions. LENS achieves an average success rate of 95.8% across the four environments, substantially higher than the 85.8% reported for LeWorldModel. This improvement is obtained without retraining the encoder or dynamics predictor. LENS also remains competitive with planning-aware methods that modify world-model training, while its core planning metric is constructed only from offline action-conditioned dynamics.

5.3 From Representation Structure to Decision Ordering

5.3.1 Physical Information after Metric Reweighting

We first study how metric reweighting changes the organization of information about physical variables in the latent representation. Following the PushT probing protocol used by LeWM and Fast-LeWM (Maes et al., 2026; Gao & Xu, 2026), we predict agent position, block position, and block orientation. We separately apply PCA to the raw representation z and the metric-transformed coordinates $M_C^{1/2} z$, and then train identical linear probes using only the first k directions. Table 2 reports the results.

5.3.2 Decision Ordering on Fixed Candidates

We next directly test whether this metric reweighting improves candidate-action ordering. For each environment, we use 50 anchors and fix exactly the same 64 candidate action sequences, world model predictions, and real execution outcomes. Each candidate is executed for 25 environment

Table 1: Planning success rate (%) on four visual goal-conditioned control tasks.

Method	**TwoRoom** ↑	**PushT** ↑	**Reacher** ↑	**Cube** ↑	**Avg.** ↑
Predictive World Models					
PLDM	97.0	78.0	78.0	65.0	79.5
DINO-WM	100.0	74.0	79.0	86.0	84.8
LeWorldModel	87.0	96.0	86.0	74.0	85.8
Training-time Planning Adaptation					
RC-aux	98.0	90.8	87.2	76.0	88.0
Temporal-Distance JEPA	100.0	86.0	97.0	82.2	91.3
SCALE	100.0	89.7	86.7	76.7	88.3
Traj-LeWM	94.0	99.0	93.0	88.0	93.5
Frozen-World-Model Planning Adaptation					
IMWM	99.2	92.7	83.8	94.7	92.6
LENS	**100.0**	**98.0**	**88.0**	**97.3**	**95.8**

steps only during offline evaluation. We then rank the same predicted futures using either the Euclidean metric or LENS M_C, and use Spearman ρ to measure agreement between the model-based ordering and the ordering induced by real task outcomes. Figure 2 summarizes the macro ordering behavior across the four environments. The horizontal axis shows the percentile of true physical endpoint distance and the vertical axis the percentile of model endpoint distance, with the diagonal representing ideal rank agreement.

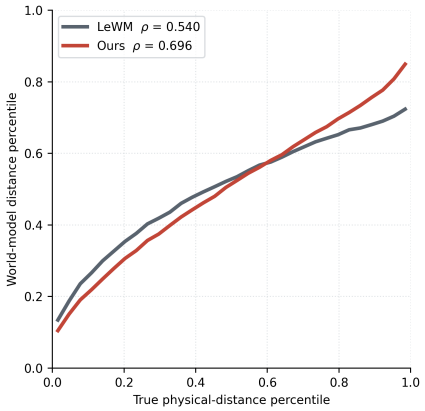


Figure 2: Fixed-candidate decision ordering.

Table 2: Physical property probing on PushT.

Property	k	**Raw** R^2	**LENS** R^2
Agent Position	8	0.172	**0.907**
	32	0.410	**0.923**
	192	0.958	0.958
Block Position	8	0.206	**0.649**
	32	0.499	**0.901**
	192	0.975	0.975
Block Orientation	8	**0.168**	0.108
	32	0.479	**0.696**
	192	0.922	0.922

Using only eight directions, agent-position R^2 increases from 0.172 to 0.907, while block-position R^2 increases from 0.206 to 0.649. At $k = 32$, LENS also improves the probing performance for all three physical variables, including block orientation. At the full 192-dimensional scale, Raw and LENS obtain the same probing results. Taken together, these results indicate that metric reweighting makes information about physical variables more concentrated in the leading directions of the transformed representation, while planning continues to use the full latent space.

At the environment level, LENS increases Spearman ρ from 0.359 to 0.694 on TwoRoom, from 0.821 to 0.825 on PushT, from 0.427 to 0.573 on Reacher, and from 0.551 to 0.691 on Cube. The macro average increases from 0.540 to 0.696 ($+0.156$), with improvement in all four environments. Because the candidate actions and model predictions are identical under the two conditions, this experiment isolates the effect of the planning metric on decision ordering rather than a change in prediction accuracy.

5.4 Ablation Study

We remove one component of LENS at a time while keeping the frozen world model and evaluation protocol fixed. As shown in Table 3, removing the Planning Metric produces the largest average performance degradation, consistent with the role of M_C derived from action-response fidelity as the central component of LENS. Retrieval Init. and Elite Reuse provide complementary improvements by improving initial candidate coverage and preserving previously discovered low-cost action modes across CEM iterations.

5.5 Applicability Across Frozen World Models

Finally, we examine whether action-response fidelity and the planning metric it induces depend on the particular architecture of LeWM. We apply LENS to PLDM and Fast-LeWM (Sobal et al., 2025; Gao & Xu, 2026) while keeping the encoder, dynamics predictor, and native prediction interface of each backbone completely frozen. For each backbone, LENS estimates the real and predicted action–future cross-covariances, prediction-error second moment, and M_C from its offline transitions, and constructs the retrieval bank from the same data. This experiment uses planner seed 42, with 50 evaluation episodes per environment; results are shown in Table 4. LENS increases the average success rate on PLDM to 90.5% and on Fast-LeWM to 97.0%, with improvements across every tested model–environment combination. The predictive backbone and its native querying interface remain unchanged, indicating that constructing a planning metric from action-response fidelity does not depend on a particular LeWM architecture.

Table 3: Leave-one-out ablation of LENS.

Method	Two Room	PushT	Reacher	Cube	Avg.
w/o Metric	91.3	92.0	81.3	92.7	89.3
w/o Init.	100.0	92.7	91.3	80.7	91.2
w/o Reuse	100.0	98.7	82.7	91.3	93.2
LENS	**100.0**	**98.0**	**88.0**	**97.3**	**95.8**

Table 4: Cross-backbone results with LENS.

Method	Two Room	PushT	Reacher	Cube	Avg.
PLDM	97.0	78.0	78.0	65.0	79.5
+ LENS	**100.0**	**90.0**	**82.0**	**90.0**	**90.5**
Fast-LeWM	98.0	96.0	88.0	80.0	90.5
+ LENS	**100.0**	**100.0**	**90.0**	**98.0**	**97.0**

6 Conclusion

We study the Prediction–Planning Gap from *future prediction* to *action selection* in action-conditioned world models. Standard latent planning compares predicted futures using Euclidean distance and therefore applies the same comparison scale to different latent directions, even though their relation to action-conditioned dynamics and the corresponding prediction accuracy can vary across directions.

To address this problem, we introduce LENS and use action-response fidelity to characterize the quality of action-conditioned prediction across latent directions. LENS derives this fidelity from the consistency of action-conditioned variation between real and predicted futures together with the accuracy of the corresponding future predictions, and uses it to construct the planning metric M_C. This metric does not change the encoder, predictor, or predicted futures; instead, it determines the relative influence of different latent directions in candidate comparison and sampling-based search.

Our experiments show that, even when candidate actions and world-model predictions remain completely fixed, changing only the planning metric can substantially improve agreement between model-based rankings and real task outcomes. The representation analysis further shows that metric reweighting makes information about physical variables more concentrated in the leading directions of the transformed representation, while the cross-backbone results demonstrate that the same principle transfers across different frozen predictive models. These results show that the planning capability of an action-conditioned world model depends not only on which futures it can predict, but also on how the planner measures differences among those predictions. Action-conditioned dynamics themselves can provide a direct basis for such a non-uniform planning metric.

AI USE STATEMENT

Generative AI tools were used as research assistants for writing support, literature retrieval and discovery, research brainstorming, preliminary manuscript drafting, mathematical reasoning, and code generation and implementation support. In particular, they were used to improve wording and presentation, suggest search queries and potentially relevant literature, discuss alternative methodological formulations and experimental considerations, help organize preliminary drafts of manuscript sections, provide feedback on mathematical derivations and proof strategies, and assist in drafting, debugging, and revising experimental and analysis code. All AI-assisted research ideas, technical arguments, mathematical claims, citations, manuscript text, and code were critically reviewed, verified, and revised by the authors before use. Relevant references were checked against the original sources, mathematical derivations and claims were independently verified, and AI-assisted code was inspected and validated through the authors' own testing and experimental procedures. The authors made all final decisions regarding the research problem, methodology, implementation, experiments, interpretation of results, and presentation, and take full responsibility for the final content and results of this work.

REPRODUCIBILITY STATEMENT

We provide detailed mathematical, algorithmic, and experimental specifications to support reproducibility. Appendix A gives the derivation and finite-sample construction of the planning metric, including conditional residualization, action whitening, prediction-error estimation, bootstrap aggregation, and properties of the deployed metric. Appendix B specifies the complete implementation of offline metric construction, Retrieval Initialization, Elite Reuse, and the resulting CEM planning procedure. Appendix C documents the frozen checkpoints and offline datasets, comparison protocols, all LENS hyperparameters, evaluation task manifests, planner seeds, ablation definitions, and implementation environment. The supplementary material additionally includes the source code used to construct the LENS metric, run the planning procedure, and reproduce the reported experiments and analyses. Across all experiments, the pretrained encoders and dynamics predictors remain frozen, and both the planning metric and retrieval bank are constructed exclusively from offline trajectories before evaluation. Evaluation rollouts are excluded from metric fitting and retrieval-bank construction and never modify the frozen world-model parameters.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 15619–15629. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01499. URL https://doi.org/10.1109/CVPR52729.2023.01499.

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

Jiaxin Bai and Jiaxuan Xiong. Temporal-distance jepa: Plan-aware representation learning for latent world model predictive control, 2026. URL https://arxiv.org/abs/2607.25337.

Dishank Bansal, Ricky T. Q. Chen, Mustafa Mukadam, and Brandon Amos. Taskmet: Task-driven metric learning for model learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91a5742235f70ae846436d9780e9f1d4-Abstract-Conference.html.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu (eds.), *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://doi.org/10.15607/RSS.2023.XIX.026.

Matthieu Destrade, Oumayma Bounou, Quentin Le Lidec, Jean Ponce, and Yann LeCun. Value-guided action planning with jepa world models, 2025. URL https://arxiv.org/abs/2601.00844.

Baoqi Gao, Ruize Han, Miao Wang, and Song Wang. Imwm: Intuition models complement world models for latent planning, 2026. URL https://arxiv.org/abs/2606.01626.

Yuntian Gao and Xiangyu Xu. Fast leworldmodel, 2026. URL https://arxiv.org/abs/2606.26217.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 2455–2467, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL http://proceedings.mlr.press/v97/hafner19a.html.

Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Jiaming Hu, Yan Zheng, and Tian Wang. Scale: State-calibrated latent embeddings for jepa planning in the right geometry, 2026. URL https://arxiv.org/abs/2608.16287.

Xiaodi Huang, Ziyi Ding, Jingtian Wan, Yuchen Liu, Yuan Zhang, Xiao-Ping Zhang, Jiayu Chen, Zhang Zhang, and Tao Huang. Traj-lewm: Path-aware world-model planning via latent trajectory cost, 2026. URL https://arxiv.org/abs/2608.14125.

Liangyu Li, Shengzhi Wang, Libin Qiu, Mingliang Xiong, and Qingwen Liu. World model control by trajectory reachability metrics, 2026a. URL https://arxiv.org/abs/2605.22164.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Predictive but not plannable: Rc-aux for latent world models, 2026b. URL https://arxiv.org/abs/2605.07278.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworldmodel: Stable end-to-end joint-embedding predictive architecture from pixels, 2026. URL https://arxiv.org/abs/2603.19312.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick

Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=M992mjgKzI.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In Jens Kober, Fabio Ramos, and Claire Tomlin (eds.), *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1049–1065. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/pinneri21a.html.

R.Y. Rubinstein and D.P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Information Science and Statistics. Springer, 2004. ISBN 9780387212401. URL https://books.google.com/books?id=9KZCZIogU4YC.

Yuhong Shi, Zhenhao Chu, Jie Wei, Jun Hao, Jianyi Liu, and Jingwen Fu. Overcoming statistical bias in action-controllable world models, 2026. URL https://arxiv.org/abs/2608.04653.

Joyjeet Singh. The objective is the bottleneck: Latent world models encode what their planners cannot use, 2026. URL https://arxiv.org/abs/2608.12959.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/3e7cf447f21cd11c846463affefce665-Abstract-Conference.html.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018. URL https://arxiv.org/abs/1801.00690.

Jiawei Wang, Ke Rui, Yushen Zuo, Yichun Feng, and Minglei Li. Decision-metric alignment in latent world models: Diagnostics and action-conditioned objectives for mpc planning, 2026a. URL https://arxiv.org/abs/2608.18746.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36411–36430. PMLR, 2023. URL https://proceedings.mlr.press/v202/wang23al.html.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim G. J. Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning, 2026b. URL https://arxiv.org/abs/2603.12231.

Yuhai Wang, Jiawei Xia, Rongxuan Zhou, Xiao Hu, Yongliang Shi, Jing Du, and Yang Ye. Prism: Prior-guided imagination sampling in world models, 2026c. URL https://arxiv.org/abs/2606.07974.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, and Nicolas Ballas. Hierarchical planning with latent world models, 2026. URL https://arxiv.org/abs/2604.03208.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: world models on pre-trained visual features enable zero-shot planning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

APPENDIX

A MATHEMATICAL DERIVATIONS AND PROPERTIES

A.1 PRE-ACTION CONTEXT AND CONDITIONAL RESIDUALIZATION

For each valid offline segment, the pre-action context is constructed as

$$c_t = \left[\mathrm{vec}(z_{t-L+1:t}), \mathrm{vec}(a_{t-L+1:t-1})\right]. \tag{A.1}$$

The actions in c_t are action blocks aligned with latent transitions rather than the individual primitive actions contained within those blocks. We use $L = 1$ for TwoRoom, in which case the action-history component is empty and $c_t = z_t$, and $L = 3$ for PushT, Reacher, and OGBench-Cube.

One model action corresponds to a block of five primitive environment actions. With a planning horizon of $H = 5$ action blocks, each valid segment spans a 25-step endpoint horizon. The complete future action sequence is represented by

$$A_t = \mathrm{vec}(a_{t:t+H-1}).$$

We define the real and predicted latent future changes over the same horizon as

$$\Delta z_t = z_{t+H} - z_t, \qquad \Delta \hat{z}_t = \hat{z}_{t+H} - z_t.$$

Both the behavior action and the subsequent latent future change can depend on the pre-action context. The law of total covariance gives

$$\mathrm{Cov}(\Delta z_t, A_t) = \mathbb{E}\left[\mathrm{Cov}(\Delta z_t, A_t \mid c_t)\right] + \mathrm{Cov}\left(\mathbb{E}[\Delta z_t \mid c_t], \mathbb{E}[A_t \mid c_t]\right).$$

Directly estimating action–future dependence can therefore mix variation associated with the action itself and variation predictable from the observed context. We reduce the latter by separately regressing the future action sequence, real latent future change, and predicted latent future change on c_t. Denoting the three fitted functions by f_A, f_T, and f_P, the held-out residuals are

$$\begin{aligned} r_t^A &= A_t - f_A(c_t), \\ r_t^T &= \Delta z_t - f_T(c_t), \\ r_t^P &= \Delta \hat{z}_t - f_P(c_t). \end{aligned} \tag{A.2}$$

The regressions are estimated using episode-grouped five-fold cross-fitting, with all segments from the same episode assigned to the same fold. For each held-out fold, the three regressions are fitted independently on the remaining four folds. Context features are standardized using statistics computed from the corresponding training folds, and the held-out fold is transformed using the same statistics.

Each ridge coefficient is independently selected from ten logarithmically spaced values between 10^{-6} and 10^{3} using deterministic generalized cross-validation on the corresponding training folds. Endpoint prediction error is computed separately as

$$e_t = \Delta z_t - \Delta \hat{z}_t = z_{t+H} - \hat{z}_{t+H},$$

and the prediction-error estimator operates directly on these endpoint errors.

A.2 Action Whitening and Symmetric Action-Response Coupling

The covariance of the residual action sequence is

$$\Sigma_A = \mathrm{Cov}(r_t^A).$$

Its finite-sample estimate is stabilized using covariance shrinkage and an eigenvalue floor. The resulting action residual is whitened according to

$$\bar{A}_t = \Sigma_A^{-1/2} r_t^A.$$

Real and predicted action–future cross-covariances are then estimated as

$$C = \mathbb{E}\left[r_t^T \bar{A}_t^\top\right], \qquad \hat{C} = \mathbb{E}\left[r_t^P \bar{A}_t^\top\right]. \tag{A.3}$$

For any unit latent direction u, the inner product

$$\left\langle C^\top u, \hat{C}^\top u \right\rangle = u^\top C \hat{C}^\top u$$

measures agreement between the real and predicted residual action-associated variation along that direction. Since a scalar quadratic form depends only on the symmetric part of its matrix,

$$u^\top C \hat{C}^\top u = u^\top \left[\frac{C\hat{C}^\top + \hat{C}C^\top}{2}\right] u.$$

This motivates the symmetric coupling

$$S = \frac{C\hat{C}^\top + \hat{C}C^\top}{2}, \qquad Su_i = \lambda_i u_i, \tag{A.4}$$

where the eigenvectors satisfy $u_i^\top u_j = \delta_{ij}$. The corresponding eigenvalue has the equivalent form

$$\lambda_i = u_i^\top S u_i = \left\langle C^\top u_i, \hat{C}^\top u_i \right\rangle.$$

Positive λ_i therefore indicate modes along which real and predicted residual action-associated variation are positively aligned. The subsequent spectral construction retains only the positive part

$$[\lambda_i]_+ = \max(\lambda_i, 0).$$

A.3 Prediction-Error Precision

Endpoint prediction errors have mean

$$\mu_e = \mathbb{E}[e_t],$$

so their second moment can be decomposed as

$$\mathbb{E}[e_t e_t^\top] = \mathrm{Cov}(e_t) + \mu_e \mu_e^\top.$$

In the finite-sample estimator, Oracle Approximating Shrinkage (OAS) is applied to the centered error covariance. The empirical error-mean outer product is then added, followed by an eigenvalue floor before inversion. We denote the resulting positive-definite estimate by $\widehat{\Sigma}_e$.

For a positive-definite matrix X, trace normalization is defined as

$$\mathrm{trnorm}(X) = \frac{d}{\mathrm{tr}(X)} X,$$

where $d = 192$ is the latent dimension. The prediction-error precision used by LENS is

$$E = \mathrm{trnorm}\left(\widehat{\Sigma}_e^{-1}\right). \tag{A.5}$$

Its directional quadratic form satisfies

$$u^\top E u \propto u^\top \widehat{\Sigma}_e^{-1} u = \left\| \widehat{\Sigma}_e^{-1/2} u \right\|_2^2,$$

providing a direction-dependent scale determined by endpoint prediction reliability. The matrix E is fitted once from the offline fitting data and remains fixed thereafter.

A.4 Bootstrap-Aggregated Action-Response Metric

Finite-sample estimation of the action-response component uses $N_{\text{boot}} = 100$ episode-level bootstrap replicates with bootstrap seed 42. Episodes are sampled with replacement from the cross-fitted residual dataset in each replicate, after which the corresponding real and predicted action-response statistics are recomputed.

For bootstrap replicate b, the symmetric coupling has eigendecomposition

$$S_b = \frac{C_b \hat{C}_b^\top + \hat{C}_b C_b^\top}{2} = U_b \Lambda_b U_b^\top,$$

with

$$\Lambda_b = \operatorname{diag}(\lambda_{b1}, \ldots, \lambda_{bd}).$$

The corresponding positive-definite action-response matrix is

$$A_b = \operatorname{trnorm}\left[I + B U_b \operatorname{diag}\left(\log(1 + [\lambda_{bi}]_+)\right) U_b^\top\right], \qquad B = 100. \tag{A.6}$$

The spectral scale B acts outside the logarithm, so the directional emphasis is

$$B \log(1 + [\lambda]_+).$$

Because

$$1 + B \log(1 + [\lambda_{bi}]_+) > 0,$$

every A_b is symmetric positive definite.

The replicate matrices are combined using a log-Euclidean mean:

$$A = \operatorname{trnorm}\left\{\exp\left[\frac{1}{N_{\text{boot}}} \sum_{b=1}^{N_{\text{boot}}} \log A_b\right]\right\}. \tag{A.7}$$

Prediction-error precision is fitted once rather than re-estimated within individual action-response bootstrap replicates. The deployed planning metric is then

$$M_C = \operatorname{trnorm}\left(E^{1/2} A E^{1/2}\right). \tag{A.8}$$

Here, A describes alignment between real and predicted residual action-associated variation, while E adjusts that structure according to prediction-error geometry.

A.5 Directional Decomposition of the Deployed Metric

The bootstrap-aggregated action-response matrix admits the eigendecomposition

$$A = U \operatorname{diag}(a_1, \ldots, a_d) U^\top.$$

After defining transformed directions

$$v_i = E^{1/2} u_i,$$

the unnormalized sandwich matrix can be written as

$$E^{1/2} A E^{1/2} = \sum_{i=1}^{d} a_i v_i v_i^\top.$$

The final trace-normalization coefficient is

$$\gamma = \frac{d}{\operatorname{tr}(E^{1/2} A E^{1/2})}.$$

For each transformed direction, define

$$\widetilde{\mathcal{F}}_i = \gamma a_i \|v_i\|_2^2, \qquad P_i = \frac{v_i v_i^\top}{\|v_i\|_2^2}.$$

The deployed metric therefore has the exact decomposition

$$M_C = \sum_{i=1}^{d} \widetilde{\mathcal{F}}_i P_i. \tag{A.9}$$

Each P_i is the orthogonal projector onto the one-dimensional subspace spanned by v_i, satisfying

$$P_i^\top = P_i, \qquad P_i^2 = P_i.$$

When E is anisotropic, however, the transformed directions $E^{1/2}u_i$ are not generally mutually orthogonal. The projectors $\{P_i\}$ therefore need not be mutually orthogonal, and the coefficients $\widetilde{\mathcal{F}}_i$ are not generally eigenvalues of M_C.

Equation (A.9) provides the finite-sample counterpart of the directional form used in Section 4.2, with the action-response component first aggregated over episode-level bootstrap replicates.

A.6 Positive Definiteness and Special Cases

The construction in Eq. (A.6) gives $A_b \succ 0$ for every bootstrap replicate. Log-Euclidean aggregation consequently preserves positive definiteness and gives

$$A \succ 0.$$

The stabilized prediction-error second moment is also positive definite, implying

$$E \succ 0.$$

For every nonzero vector x,

$$x^\top E^{1/2} A E^{1/2} x = (E^{1/2}x)^\top A (E^{1/2}x) > 0.$$

Trace normalization multiplies the matrix by a positive scalar and therefore preserves symmetry, rank, and positive definiteness. Consequently,

$$M_C \succ 0, \qquad \operatorname{rank}(M_C) = d, \qquad \operatorname{tr}(M_C) = d.$$

The same construction contains several useful special cases. When the action-response component is isotropic,

$$A = I \quad \Longrightarrow \quad M_C = \operatorname{trnorm}(E).$$

When prediction-error geometry is isotropic,

$$E = I \quad \Longrightarrow \quad M_C = A.$$

When both components are isotropic,

$$A = I, \qquad E = I \quad \Longrightarrow \quad M_C = I.$$

Euclidean distance is therefore recovered as the isotropic special case.

B Algorithmic Implementation

B.1 Offline Metric Construction

For each environment, M_C is constructed once before evaluation. Valid offline trajectory segments contain the required pre-action context, five future action blocks, and the corresponding 25-step endpoint. The same future action sequence is passed through the frozen world model to obtain its predicted endpoint.

Episode-grouped five-fold residualization is first applied to the offline segments. The resulting action residuals are whitened, the real and predicted cross-covariances in Eq. (A.3) are estimated, and the fixed prediction-error precision E is constructed. The action-response component is then estimated over 100 episode-level bootstrap replicates and aggregated using Eq. (A.7). Finally, the deployed metric is obtained from Eq. (A.8) and remains fixed throughout evaluation.

B.2 Retrieval Initialization

The retrieval bank contains valid reference segments from the offline trajectories. Each entry stores a start latent $z_{s,i}$, the corresponding 25-step future latent $z_{g,i}$, and the complete five-block action plan A_i.

For the current start–goal pair (z_t, z_g), reference trajectories are compared using

$$D_i = (z_t - z_{s,i})^\top M_C (z_t - z_{s,i}) + (z_g - z_{g,i})^\top M_C (z_g - z_{g,i}). \tag{B.1}$$

A factor L_C satisfying

$$M_C = L_C^\top L_C$$

turns Eq. (B.1) into Euclidean nearest-neighbor search over the concatenated transformed keys

$$q_t = \begin{bmatrix} L_C z_t \\ L_C z_g \end{bmatrix}, \qquad q_i = \begin{bmatrix} L_C z_{s,i} \\ L_C z_{g,i} \end{bmatrix}.$$

The implementation uses the equivalent row-vector convention with a right-multiplying Cholesky factor R, satisfying $M_C = RR^\top$, and transforms latent row vectors as zR.

Retrieval is exact. The nearest 30 plans determine local plan dispersion, while the nearest plan provides the center of the retrieved proposal. For standardized action-plan vectors $A_i^{\mathrm{std}} \in \mathbb{R}^{D_A}$, the per-coordinate squared plan distance is

$$d_{\mathrm{plan}}(i,j) = \frac{1}{D_A} \left\| A_i^{\mathrm{std}} - A_j^{\mathrm{std}} \right\|_2^2.$$

Local dispersion is computed from all pairwise distances among the top-30 plans:

$$U_{\mathrm{local}} = \mathrm{median}_{1 \le i < j \le 30}\, d_{\mathrm{plan}}(i,j).$$

Global dispersion is estimated once for each environment. At most $8{,}192$ plans are deterministically sampled from the retrieval bank using a fixed random seed, randomly arranged into non-overlapping pairs $\mathcal{P}_{\mathrm{global}}$, and summarized by

$$U_{\mathrm{global}} = \mathrm{median}_{(i,j) \in \mathcal{P}_{\mathrm{global}}}\, d_{\mathrm{plan}}(i,j).$$

Their ratio determines the adaptive retrieval weight

$$\rho = \mathrm{clip}\left(1 - \frac{U_{\mathrm{local}}}{U_{\mathrm{global}}}, 0, 1\right). \tag{B.2}$$

The number of iteration-0 candidates sampled around the retrieved reference is

$$N_{\mathrm{ret}} = \lfloor \rho N + 0.5 \rfloor. \tag{B.3}$$

The remaining $N - N_{\mathrm{ret}}$ candidates are drawn from the regular proposal. The iteration-0 refit uses source-aware weighting according to ρ; subsequent CEM iterations no longer retain candidate-origin labels.

B.3 Elite Reuse

Elite Reuse preserves low-cost action modes found during the current CEM solve. Each reuse set contains $M_{\mathrm{reuse}} = 8$ complete action sequences. Its onset is $S = 8$ under zero-based CEM iteration indexing, so iterations $0, \ldots, 7$ contain no injected reused candidates. The first reuse set is constructed after iteration 7 and injected into iteration 8, corresponding to the ninth CEM iteration.

For the $K = 30$ elite predicted endpoints $\hat{z}_1, \ldots, \hat{z}_K$, pairwise similarity is computed from the LENS endpoint distance

$$D_{ij}^{\mathrm{pred}} = (\hat{z}_i - \hat{z}_j)^\top M_C (\hat{z}_i - \hat{z}_j). \tag{B.4}$$

The median off-diagonal distance

$$\delta = \mathrm{median}_{i<j}\, D_{ij}^{\mathrm{pred}}$$

sets the scale of the similarity kernel

$$\kappa_{ij} = \exp\left[-\frac{D_{ij}^{\text{pred}}}{\delta + 10^{-8}}\right].$$

Consensus for elite i is then measured by

$$s_i = \frac{1}{K-1}\sum_{j \neq i} \kappa_{ij}. \tag{B.5}$$

The eight elites with the largest consensus scores are retained as complete action sequences for the next eligible iteration. Once reuse becomes active, each CEM population contains 292 fresh samples and 8 reused sequences, keeping the total population fixed at 300. Reused sequences are re-evaluated by the frozen world model in every iteration in which they appear, and the reuse state is reset at the beginning of each MPC solve.

B.4 Complete Planning Procedure

LENS follows the LeWorldModel receding-horizon MPC setup with a metric-guided modified CEM solver. All four environments use $N = 300$ candidates, $K = 30$ elites, $I = 30$ optimization iterations, and $H = 5$ action blocks. Each action block contains five primitive actions, so every candidate prediction spans 25 primitive environment steps. Only the first block of the returned plan is executed before replanning.

Candidate sequence A_j is evaluated through the terminal cost

$$J(A_j) = (\hat{z}_j - z_g)^\top M_C (\hat{z}_j - z_g). \tag{B.6}$$

Algorithm 1 *LENS CEM Planning Process*

Input: current observation o_t, goal observation o_g, frozen encoder E_θ, frozen predictor $F^\star$, planning metric M_C, retrieval bank $\mathcal{B}$.
Encode $z_t \leftarrow E_\theta(o_t)$ and $z_g \leftarrow E_\theta(o_g)$.
Retrieve the top-30 reference plans using Eq. (B.1).
Compute ρ using Eq. (B.2) and N_{ret} using Eq. (B.3).
Initialize the reuse set $\mathcal{R} \leftarrow \varnothing$.
for CEM iteration $i = 0, \ldots, I-1$ **do**
  if $i = 0$ **then** sample N_{ret} retrieved-centered proposals and $N - N_{\text{ret}}$ regular proposals.
  else sample N candidates from the current CEM proposal.
  if $i \geq S$ **then** replace M_{reuse} fresh candidates with the sequences in $\mathcal{R}$.
  Roll out all candidates for H action blocks using $F^\star$.
  Evaluate all predicted endpoints using Eq. (B.6).
  Select the K lowest-cost candidates as elites.
  Refit the CEM proposal using the elite empirical mean μ_i and standard deviation σ_i; use the retrieval source-aware weighting at $i = 0$.
  if $i < I-1$ and $i+1 \geq S$ **then** form $\mathcal{R}$ from the M_{reuse} elites with the largest scores in Eq. (B.5).
end for
Return the final CEM mean action sequence $A^\star \leftarrow \mu_{I-1}$.
Execute the first action block of $A^\star$ and replan from the resulting observation.

C Experimental Protocol

C.1 Frozen Backbone and Offline Datasets

The main experiments use released LeWorldModel checkpoints as frozen predictive backbones. Both the visual encoder and the action-conditioned predictor remain fixed throughout metric fitting and evaluation. All observations are 224×224 RGB images, and the latent dimension is $d = 192$.

Table 5: Frozen backbones and offline data used in the main LENS experiments.

Environment	Frozen checkpoint	Offline dataset	Action dim.	Context L
TwoRoom	`quentinll/lewm-tworooms`	TwoRoom	2	1
PushT	`quentinll/lewm-pusht`	`pusht_expert_train`	2	3
Reacher	`quentinll/lewm-reacher`	`dmc/reacher_random`	2	3
OGBench-Cube	`quentinll/lewm-cube`	`ogbench/cube_single_expert`	5	3

Valid segments from the offline trajectories are used for metric fitting and retrieval-bank construction. Evaluation episodes are excluded from both sources, and cross-fitting folds are grouped by episode. Evaluation rollouts do not modify the frozen world model, M_C, or the retrieval bank.

C.2 Main-Table Comparison Protocols

The source configurations associated with the literature rows in Table 1 are summarized in Table 6.

Table 6: Planning and evaluation configurations associated with the main comparison table. A dash indicates that the corresponding quantity is not used as a common reported specification for that entry.

Method	Candidates	Iterations	Elites	Horizon / goal setting	Evaluation aggregation
PLDM	300	30	30	5 blocks / 25 steps	LeWM benchmark
DINO-WM	300	30	30	5 blocks / 25 steps	LeWM benchmark
LeWorldModel	300	30	30	5 blocks / 25 steps	LeWM benchmark
RC-aux	300	30 / 10	30	goal offset 25	5×50 episodes
Temporal-Distance JEPA	300	30 / 10	–	$H = 5,\ G = 25$	10×50 episodes
SCALE	300	30	30	5 blocks / 25 steps	6×100 episodes
Traj-LeWM	300	–	30	5 blocks / 25 steps	3×50 queries
IMWM	300	30	30	$H_{\text{macro}} = 5$	12×50 episodes
LENS	300	30	30	5 blocks / 25 steps	3×50 tasks

PLDM, DINO-WM, and LeWorldModel. These three entries are taken from the common four-environment benchmark reported with LeWorldModel (Maes et al., 2026). The corresponding LeWorldModel codebase uses CEM with 300 candidates, 30 elites, 30 optimization iterations, and a five-block horizon in all four environments. DINO-WM uses the pixels-only benchmark configuration, while PLDM follows the corresponding benchmark implementation.

RC-aux. The reported RC-aux configuration uses 300 candidates and 30 elites on the four environments shared with LENS (Li et al., 2026b). TwoRoom, Reacher, and PushT use 30 planning iterations, whereas Cube uses 10. Results are averaged over five fixed evaluation groups of 50 episodes. The reported Cube configuration sets the planner reachability weight to zero.

Temporal-Distance JEPA. Its primary evaluation protocol uses 300 candidates per iteration (Bai & Xiong, 2026). TwoRoom and Reacher use iCEM-30 with the learned temporal-distance cost, whereas PushT and OGBench-Cube use latent ℓ_2 with CEM-30 and CEM-10, respectively. The reported results average ten independent planning-seed runs.

SCALE. SCALE reports CEM, iCEM, and MPPI across multiple compute tiers (Hu et al., 2026). Table 1 uses iCEM-T1, the reported solver/tier configuration with the highest four-task macro suc-

cess over the environments shared with LENS. T1 uses 300 candidates, 30 optimization iterations, and 30 elites. Its CEM/iCEM results are averaged over six paired evaluation sets of 100 episodes.

Traj-LeWM. The reported full endpoint-plus-LTC configuration samples 300 candidate sequences and retains 30 elites in each planning round (Huang et al., 2026). Formal evaluation uses planner seeds 42, 43, and 44, with 50 start–goal queries per seed.

IMWM. The headline reliability-gated configuration uses 300 candidates, 30 CEM iterations, and 30 elites per replan (Gao et al., 2026). Its headline evaluation contains 12 cells per task with 50 episodes per cell. On Reacher, the deployed reliability gate selects the forward-only fallback and disables retrieval.

LENS. All four environments use $N = 300$, $K = 30$, $I = 30$, and $H = 5$. Planner seeds 42, 43, and 44 evaluate the same fixed 50-task manifest.

The methods in Table 1 share the same four visual goal-conditioned control environments and use closely related finite-budget model-predictive planning settings. Most employ CEM/iCEM-style optimization with approximately 300 candidates per iteration and comparable short-horizon latent rollouts. Literature rows retain the task manifests, trained checkpoints, randomization, and success criteria of their respective source evaluations. Section D.1 provides the locally matched comparison between LeWorldModel and LENS under the same backbone, task manifest, planner seeds, horizon, and nominal CEM budget.

C.3 LENS HYPERPARAMETERS

Table 7: Hyperparameters used by LENS in the main evaluation.

Component	**Setting**
Frozen backbone	LeWorldModel
Latent dimension	192
Primitive actions per block	5
Planning horizon	5 action blocks
Endpoint horizon	25 primitive steps
Residualization context	$L = 1$ TwoRoom; $L = 3$ otherwise
Conditional residualization	episode-grouped 5-fold cross-fitting
Regressions	separate ridge fits for action / real / predicted future
Ridge grid	10 log-spaced values, 10^{-6} to 10^{3}
Ridge selection	deterministic GCV
Residual-action covariance	shrinkage + eigenvalue floor
Prediction-error estimator	OAS + error-mean outer product + eigenvalue floor
Action-response scale B	100
Bootstrap replicates	100
Bootstrap seed	42
Bootstrap aggregation	log-Euclidean mean
CEM candidates N	300
CEM elites K	30
CEM iterations I	30
Retrieval neighborhood	exact Top-30
Retrieval reference	Top-1 plan
Retrieval mixing	adaptive ρ
Elite Reuse size	8
Elite Reuse onset	$S = 8$ (zero-based)
MPC execution	execute one action block and replan

C.4 Evaluation Tasks and Planner Seeds

The main evaluation uses the same fixed set of 50 tasks for each planner seed. Seeds 42, 43, and 44 repeat those tasks while changing only planner randomness. Success is first computed separately for each seed and environment and then averaged over the three seeds. The macro average gives equal weight to TwoRoom, PushT, Reacher, and OGBench-Cube.

The cross-backbone experiment in Section 5.5 uses planner seed 42 with 50 evaluation episodes per environment.

C.5 Leave-One-Out Ablation Definitions

w/o Retrieval Init. Retrieval Initialization is disabled while the planning metric M_C and Elite Reuse remain active.

w/o Elite Reuse. Elite Reuse is disabled while M_C and Retrieval Initialization remain active.

w/o Planning Metric. The identity matrix replaces M_C in every metric-dependent planner operation, including terminal candidate evaluation, retrieval keys, and Elite Reuse endpoint consensus. Retrieval Initialization and Elite Reuse themselves remain enabled.

C.6 Implementation Environment

Formal evaluation uses MuJoCo 3.10.0 with `MUJOCO_GL=egl` and `PYOPENGL_PLATFORM=egl`. The frozen world model, planning metric, and retrieval bank remain fixed throughout evaluation.

D Additional Controlled Experiments

D.1 Matched LeWorldModel Evaluation

In addition to the published LeWorldModel result in Table 1, we rerun the LeWorldModel planner using the same frozen backbone, evaluation task manifest, planner seeds, planning horizon, and nominal CEM budget used by LENS.

Table 8: Published LeWorldModel performance and the locally matched LeWorldModel–LENS comparison.

Method	TwoRoom	PushT	Reacher	Cube	Avg.
LeWM – Published	87.00	96.00	86.00	74.00	85.80
LeWM – Matched	87.33	95.33	77.33	66.00	81.50
LENS	**100.00**	**98.00**	**88.00**	**97.33**	**95.83**

Per-seed success counts out of 50 tasks are reported below.

Table 9: Per-seed success counts for the locally matched LeWorldModel and LENS evaluation.

Environment	LeWM Matched: 42 / 43 / 44	LENS: 42 / 43 / 44
TwoRoom	42 / 45 / 44	50 / 50 / 50
PushT	47 / 49 / 47	50 / 49 / 48
Reacher	39 / 40 / 37	43 / 44 / 45
Cube	34 / 32 / 33	49 / 48 / 49

Under this locally matched protocol, the four-environment macro success rate is 81.50% for LeWorldModel and 95.83% for LENS.

D.2 Leave-One-Out Ablations

The full leave-one-out results corresponding to the definitions in Appendix C are reported in Table 10.

Table 10: Leave-one-out ablations of the three LENS planning components.

Method	TwoRoom	PushT	Reacher	Cube	Avg.
w/o Retrieval Init.	100.00	92.67	91.33	80.67	91.17
w/o Elite Reuse	100.00	98.67	82.67	91.33	93.17
w/o Planning Metric	91.33	92.00	81.33	92.67	89.33
Full LENS	**100.00**	**98.00**	**88.00**	**97.33**	**95.83**

The three components act at different stages of the same planning process. The planning metric defines the comparison geometry used for candidate evaluation, retrieval, and Elite Reuse; Retrieval Initialization shapes the iteration-0 proposal; and Elite Reuse preserves low-cost action modes during subsequent CEM iterations.

D.3 Prediction Accuracy and Action Response Components

The two components of the planning metric are isolated through

$$M_{\text{Accuracy}} = E, \qquad M_{\text{Response}} = A,$$

while the full metric remains

$$M_{\text{Full}} = \text{trnorm}\left(E^{1/2} A E^{1/2}\right).$$

The planning algorithm, hyperparameters, and candidate-evaluation budget are identical across the three metric variants. Each metric is applied consistently to every metric-dependent operation, including terminal candidate evaluation, retrieval, and Elite Reuse consensus.

Table 11: Closed-loop component analysis of prediction-error precision and action-response structure.

Method	TwoRoom	PushT	Reacher	Cube	Avg.
LeWM – Matched	87.33	95.33	77.33	66.00	81.50
Prediction Accuracy Only	98.67	96.00	84.00	95.33	93.50
Action Response Only	96.67	95.33	82.00	94.00	92.00
Full LENS	**100.00**	**98.00**	**88.00**	**97.33**	**95.83**

Both single-component metrics improve the four-task macro success rate over the locally matched LeWorldModel baseline, while their combination yields the highest overall performance. Prediction-error precision supplies direction-dependent prediction reliability, whereas the action-response component captures alignment between real and predicted action-associated variation.