A SPECTRAL TRAJECTORY OBJECTIVE FOR WORLD MODEL PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective planning with world models depends not only on predictive accuracy, but also on how imagined trajectories are evaluated. Learning auxiliary planning objectives can improve control, but introduces additional training and may require task-specific supervision. We instead improve planning by redesigning the objective, without training any additional model. We introduce a spectral trajectory objective that stacks goal-relative latent residuals along an imagined rollout into a time-by-feature matrix and minimizes its horizon-normalized squared spectral norm. Unlike terminal distance, this objective considers the full predicted trajectory; unlike summed per-step errors, it accounts for alignment between residuals across time and emphasizes the dominant joint discrepancy. Across four LeWorldModel tasks and three planning horizons, Spectral improves task- and horizon-averaged success over Frobenius scoring by 13.3 percentage points, with a separate uniform-configuration evaluation yielding a similar 13.8-point gain. The same objective transfers to DINO-WM, where it shows clearer gains over terminal scoring at longer planning horizons. These results establish trajectory-objective geometry as an effective design dimension for world-model planning without additional model training.

1 INTRODUCTION

Visual world models turn image sequences into compact predictive states and action-conditioned forecasts. A planner searches these imagined futures before executing a short action sequence, as in latent planning from pixels, visual foresight, and imagination-based control (Ha & Schmidhuber, 2018; Hafner et al., 2019; Finn & Levine, 2017; Ebert et al., 2018; Hafner et al., 2020; Chua et al., 2018). Recent systems add latent optimizers and learned value interfaces (Hansen et al., 2024; 2022). Prediction alone does not determine how those futures should be ranked.

The gap is especially clear when the predictor is already trained and frozen. A learned goal, reachability, or value model can improve control, but adds data, optimization choices, and task dependence (Nair et al., 2018; 2023; Ma et al., 2023b;a). This is one form of objective mismatch: useful predictions can still be ranked by the wrong quantity (Lambert et al., 2020; Singh, 2026). We ask what can be changed in the trajectory readout alone.

Terminal distance uses only the final residual. A Frobenius objective includes intermediate residuals but aggregates only their squared magnitudes. This nested baseline design tests first whether the full rollout matters, then whether anything beyond trajectory energy matters: do residuals repeatedly point along one latent direction, or spread their energy across directions?

We answer this question with a spectral trajectory objective. For an imagined rollout, we stack goal-relative residuals into a time-by-feature matrix

$$\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_H]^\top.$$

The score is the horizon-normalized squared operator norm of $\mathbf{E}$, measuring the largest accumulated discrepancy along a unit feature direction. Thus equal endpoint error and equal Frobenius energy can still receive different scores. Figure 1 makes this distinction exact. The construction uses the frozen rollout directly: no auxiliary scorer, task-specific labels, or model update is introduced. Training-free means that no additional planning model is trained.

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.

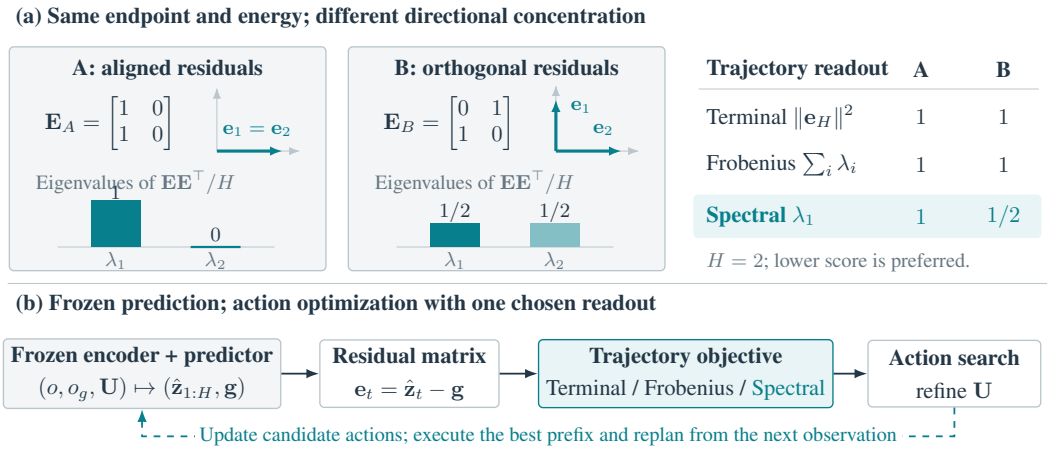

Figure 1: **Equal endpoint error and total energy can hide different residual geometry.** (a) The two $H = 2$ residual matrices have identical final rows and Frobenius energy. Their normalized Gram spectra differ: Spectral assigns 1 to aligned residuals and $1/2$ to orthogonal residuals. This algebraic example illustrates a scoring preference, not physical reachability. (b) A frozen rollout supplies the residual matrix; only the trajectory readout changes. Action search uses one of the three alternative scores, then executes a prefix and replans. Model weights remain fixed, so the method adds no planning-specific training stage.

The interface applies to jointly learned predictive representations and pretrained visual features, provided they supply goal-relative latent rows. We evaluate the same objective on LeWorldModel and DINO-WM, using each model's native representation and evaluation protocol.

Across four LeWorldModel tasks and $H \in \{5, 10, 15\}$, the selected panel gives success rates averaged equally over tasks and horizons of 28.2%, 48.9%, and 62.2% for Terminal, Frobenius, and Spectral. Full-trajectory energy adds 20.7 points over Terminal; spectral aggregation adds a further 13.3 points over Frobenius. Uniform, replication, solver, and geometry controls test this contrast. Configurations are selected using Spectral performance on development episodes and then applied unchanged to all three objectives; the uniform panel and task decomposition provide complementary checks. DINO-WM supplies a second frozen family: longer horizons separate Spectral from Terminal more clearly than from Frobenius, while short horizons define the transfer boundary.

We contribute a spectral trajectory readout with no additional planning-model training, a geometric analysis of endpoint error, residual energy, and directional concentration, and a matched evaluation across horizons, solvers, and frozen model families.

2 Related Work

World models, visual representations, and control objectives. World models support latent planning, visual foresight, imagination-based control, and sampled model-predictive optimization (Ha & Schmidhuber, 2018; Hafner et al., 2019; Finn & Levine, 2017; Ebert et al., 2018; Hafner et al., 2020; Chua et al., 2018; Hansen et al., 2022; 2024). MuZero and value equivalence emphasize decision-relevant latent predictions (Schrittwieser et al., 2020; Grimm et al., 2020). LeWorldModel is the jointly learned family in our main panel, whereas DINO-WM predicts pretrained features (Maes et al., 2026b;a; Zhou et al., 2025). Self-supervised and joint-embedding predictors provide reusable visual features (Caron et al., 2021; Oquab et al., 2024; Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025); RIG, R3M, VIP, LIV, and reachability metrics turn related features into goal or value signals (Nair et al., 2018; 2023; Ma et al., 2023b;a; Li et al., 2026). These approaches learn a representation, reward, or value interface. Video-prediction models additionally represent uncertainty over futures (Denton & Fergus, 2018; Babaeizadeh et al., 2018); we instead hold the rollout fixed and change its readout.

Objective mismatch formalizes the gap between the quantity a world model predicts and the quantity a planner optimizes (Lambert et al., 2020), and frozen-model analyses identify terminal distance as a possible control interface failure (Singh, 2026). The same residual matrix can be scored by Terminal, Frobenius, or Spectral without fitting a new cost. The This distinction is geometric and complements representation learning rather than replacing it.

Search and matrix objectives. The trajectory readout is evaluated inside standard action-search procedures, including projected-gradient methods (Birgin et al., 2000; Barzilai & Borwein, 1988), the cross-entropy method (Rubinstein, 1999), and iCEM (Pinneri et al., 2021). Differentiable MPC differentiates a controller or its optimality conditions to learn planning components (Amos et al., 2018); our planner differentiates only a fixed rollout score with respect to actions. The operator norm and its subdifferential are standard matrix objects (Horn & Johnson, 2013; Watson, 1992). Spectral structure in learned transition operators has been studied from a different perspective (You et al., 2026); that work does not define the candidate trajectory score used here. Our contribution is a trajectory-level use of residual spectra, evaluated across planning horizons, solvers, and two frozen model families.

3 Trajectory Objectives

A frozen encoder maps the current observation to $\mathbf{z}_0$ and the goal image to $\mathbf{g}$. Let $\mathbf{U} \in \mathcal{B} \subset \mathbb{R}^{bH \times a}$ contain bH normalized physical action vectors of dimension a. A frozen world model predicts $\hat{\mathbf{z}}_t(\mathbf{U}) = f_t(\mathbf{z}_0, \mathbf{U}_{1:bt})$ for $t = 1, \dots, H$, packing $b = 5$ consecutive physical actions into each model input. The planner executes at most bR actions, with $R \leq H$, and replans from the next observation. Native observation history or proprioceptive context, when required by the backbone, is supplied to f but does not enter the scoring matrix.

Let $\mathbf{e}_t = \hat{\mathbf{z}}_t(\mathbf{U}) - \mathbf{g}$ and $\mathbf{E}(\mathbf{U}) = [\mathbf{e}_1, \dots, \mathbf{e}_H]^\top \in \mathbb{R}^{H \times d}$. The matrix excludes input history and the initial observation, which are not predicted consequences of the candidate actions. We compare

$$J_{\mathrm{T}}(\mathbf{U}) = \|\mathbf{e}_H\|_2^2, \qquad \text{endpoint error,} \tag{1}$$

$$J_{\mathrm{F}}(\mathbf{U}) = H^{-1}\|\mathbf{E}\|_F^2, \qquad \text{mean trajectory energy,} \tag{2}$$

$$J_{\mathrm{op}}(\mathbf{U}) = H^{-1}\|\mathbf{E}\|_{\mathrm{op}}^2 = H^{-1}\lambda_{\max}(\mathbf{E}\mathbf{E}^\top), \qquad \text{dominant directional energy.} \tag{3}$$

Terminal omits intermediate residuals. Frobenius includes every row but retains only total energy; it is therefore the control that isolates the additional effect of spectral aggregation. At $H = 1$ all three scores coincide. Squaring and dividing by H preserve exact minimizers at a fixed horizon, although finite-step optimization can change (Appendix F).

4 Spectral Geometry and Action Optimization

4.1 What the spectral score measures

The variational form

$$J_{\mathrm{op}} = \max_{\|\mathbf{v}\|_2 = 1} H^{-1} \sum_{t=1}^{H} (\mathbf{e}_t^\top \mathbf{v})^2 \tag{4}$$

measures accumulated residual energy in the most affected latent direction. For the matrices in Figure 1, both terminal scores and both Frobenius scores equal 1, whereas the spectral scores are 1 and $1/2$. This example controls both baselines while varying directional concentration. More generally, H parallel or orthogonal unit residuals have the same Frobenius score but spectral scores of 1 and $1/H$, respectively, when $d \geq H$.

For nonzero $\mathbf{E}$, write

$$J_{\mathrm{op}} = a(\mathbf{E}) J_{\mathrm{F}}, \qquad a(\mathbf{E}) = \frac{\lambda_{\max}(\mathbf{E}\mathbf{E}^\top)}{\mathrm{tr}(\mathbf{E}\mathbf{E}^\top)}.$$

The score couples residual energy with directional concentration. At fixed energy it favors dispersion; across different energies it can prefer a rollout with more total error if that error is sufficiently dispersed. Whether this helps control depends on the representation, motivating the matrix controls

Algorithm 1 Receding-horizon planning with a spectral trajectory readout

Require: Frozen encoder ϕ and predictor f; goal o_g; action box $\mathcal{B}$;
horizon H, block size b, executed blocks $R \leq H$;
proposal count N, starts K, update budget L, physical budget B

1: $\mathbf{g} \leftarrow \phi(o_g)$; $B_{\mathrm{rem}} \leftarrow B$; initialize warm tail and paired random stream
2: **while** $B_{\mathrm{rem}} > 0$ and the environment has not terminated **do**
3: Observe o, construct native context c, and draw N feasible action sequences
4: Define $\mathbf{E}(\mathbf{U}) \leftarrow \mathrm{rowstack}_{t=1}^{H}(\hat{\mathbf{z}}_t(c, \mathbf{U}) - \mathbf{g})$, $J(\mathbf{U}) \leftarrow \lambda_{\max}(\mathbf{E}\mathbf{E}^\top)/H$
5: Select the K lowest-J proposals as independent starts; reset per-start solver state
6: Refine each start for at most L projected updates using $\nabla_{\mathbf{U}} J$
7: Retain the best feasible incumbent per start; failed updates preserve that incumbent
8: Choose $\mathbf{U}^\star$ with the lowest J among the retained incumbents
9: Denormalize and execute up to bR actions, bounded by remaining budget and native termination
10: Update remaining budget and the optional warm tail from actions actually executed
11: **end while**

Only action variables are updated; gradients pass through the frozen model. Terminal and Frobenius replace J throughout proposal ranking and refinement. Appendix B specifies proposal rules, per-start state, Armijo acceptance and early stopping.

and execution diagnostics. The score is invariant to a common orthogonal feature rotation, row permutation and individual row sign changes. It captures cross-time alignment but not temporal order or arrival time. Orthogonal invariance does not imply invariance to arbitrary feature scaling. Appendix A gives the bounds and an equivalent variational form over time coefficients.

4.2 Optimizing actions through a frozen model

For a unit leading eigenvector $\mathbf{q}$ of $\mathbf{E}\mathbf{E}^\top$, a simple leading eigenvalue gives

$$\nabla_{\mathbf{e}_t} J_{\mathrm{op}} = \frac{2q_t}{H} \sum_{s=1}^{H} q_s \mathbf{e}_s, \qquad \nabla_{\mathbf{e}_t} J_{\mathrm{F}} = \frac{2}{H} \mathbf{e}_t. \tag{5}$$

Spectral thus couples residuals across time before differentiating through the rollout to actions. When the leading eigenvalue is repeated, $2\mathbf{q}\mathbf{q}^\top \mathbf{E}/H$ is a valid subgradient. Evaluation via the $H \times H$ Gram matrix costs $O(H^2 d + H^3)$ and avoids a feature-dimension eigendecomposition.

Algorithm 1 summarizes projected action search with SPG-style step sizes (Birgin et al., 2000) and Armijo safeguards. The step-size rule is independent of the Spectral objective. A failed update retains its feasible incumbent; only actions are optimized. The default configuration uses up to ten updates, 128 proposals and four starts. Shared random streams pair proposal generation, although warm candidates depend on each objective's preceding plan. Selected configurations may also change diagonal scaling, update budgets or the executed prefix; Appendix B specifies all settings, acceptance rules and numerical stopping criteria.

5 Experiments

LeWorldModel. Table 1 evaluates released Reacher, TwoRoom, PushT, and single-Cube checkpoints on 50 episodes per task at $H = 5, 10, 15$. Configurations are selected on development episodes and then fixed for all three objectives. We label the selected panel P_{sel}: its 50 Reacher episodes are reserved by the 2026-09-26 focus study, while the 50- episode TwoRoom, PushT, and Cube panels are retained from the 2026-09-25 validation panel. These 200 episodes are disjoint from the 2026-09-19 uniform panel P_{uni}. The independent 2026-09-23 replication contains 20 additional episodes per task and remains separate (Appendix E). Goal images are held fixed across planning horizons, and each episode has an execution budget of 100 physical steps. Larger H expands prediction and action search. State is used only for restoration and native success evaluation. All normalization, restoration, and configuration details are in Appendix B.

DINO-WM transfer. Frozen PushT and Wall models provide a second representation family. The grid uses $H \in \{5, 10\}$, PushT goal offsets 25/50, native Wall goals, 16 PushT trajectories, 50 Wall episodes, three paired planner seeds, and a shared SPG solver. The matched-start extension changes five of 16 PushT confirmation starts and one development start to admit offset-50 goals; its original-start panel is kept separate in Appendix G. LeWorldModel tuning is not transferred. Native restoration and success predicates differ across families, so rates are never pooled; protocol details appear in Appendix G.

Pairing and uncertainty. Objectives share checkpoints, normalization, search settings, and physical budgets. Seeds $\{42, 123, 2026\}$ repeat the same episodes. LeWorldModel intervals use paired within-task episode bootstrap; the statistical families are specified in Appendix D, while DINO pool construction and interval scopes are specified in Appendix G.

5.1 Endpoint, trajectory energy, and residual geometry

Table 1 separates two improvements. Full-trajectory energy increases success rates averaged equally over tasks and horizons from 28.2% to 48.9%; spectral aggregation increases it further to 62.2%. These descriptive increments are 20.7 and 13.3 percentage points. Table 2 reports paired Spectral advantages of 13.28 points over Frobenius and 34.00 over Terminal.

Table 1: **LeWorldModel success rates (%).** T/F/S denote Terminal/Frobenius/Spectral. Each cell has 50 episodes and three seeds under a shared configuration. †Reacher uses newly reserved episodes; other samples are retained. Spectral development performance selects the configurations (Section 5); bold marks numerical maxima.

	$H = 5$			$H = 10$			$H = 15$		
Task	T	F	S	T	F	S	T	F	S
Reacher†	80.0	98.0	**100.0**	15.3	**100.0**	99.3	10.0	**97.3**	91.3
TwoRoom	38.0	29.3	**72.0**	20.7	27.3	**82.0**	10.0	20.0	**67.3**
PushT	4.7	6.0	**11.3**	3.3	7.3	**9.3**	0.7	7.3	**8.0**
Cube	59.3	62.0	**68.0**	49.3	67.3	**69.3**	46.7	64.7	**68.0**

Table 2: Task-equal Spectral advantage in percentage points [95% episode-cluster interval]. Tasks receive equal weight; seeds and included horizons remain together when resampling each episode. Intervals use 20,000 draws and are not multiplicity adjusted.

H averaged	vs. Frobenius	vs. Terminal
5	+14.0 [+10.7, +17.5]	+17.3 [+13.7, +21.3]
10	+14.5 [+10.8, +18.2]	+42.8 [+38.7, +47.2]
15	+11.3 [+8.0, +14.8]	+41.8 [+37.7, +46.0]
5,10,15	+13.3 [+10.7, +16.0]	+34.0 [+30.9, +37.3]

Table 3: **Task dependence under three comparison protocols.** Spectral minus Frobenius, in percentage points. Selected averages $H = 5, 10, 15$; Uniform and Replication average $H = 5, 10$. Each averages three paired seeds on 50/50/20 episodes per task. The last row retains the original aggregate intervals; Appendix C gives post-hoc leave-one-task-out analyses.

Task	Selected (L-S)	Uniform (L-U)	Replication (L-R)
Reacher	-1.56	+1.00	-12.50
TwoRoom	+48.22	+55.00	+55.83
PushT	+2.67	-0.67	+3.33
Cube	+3.78	-0.33	+1.67
All tasks	+13.28	+13.75	+12.08
95% interval	[10.7, 16.0]	[11.0, 16.5]	[6.9, 17.5]

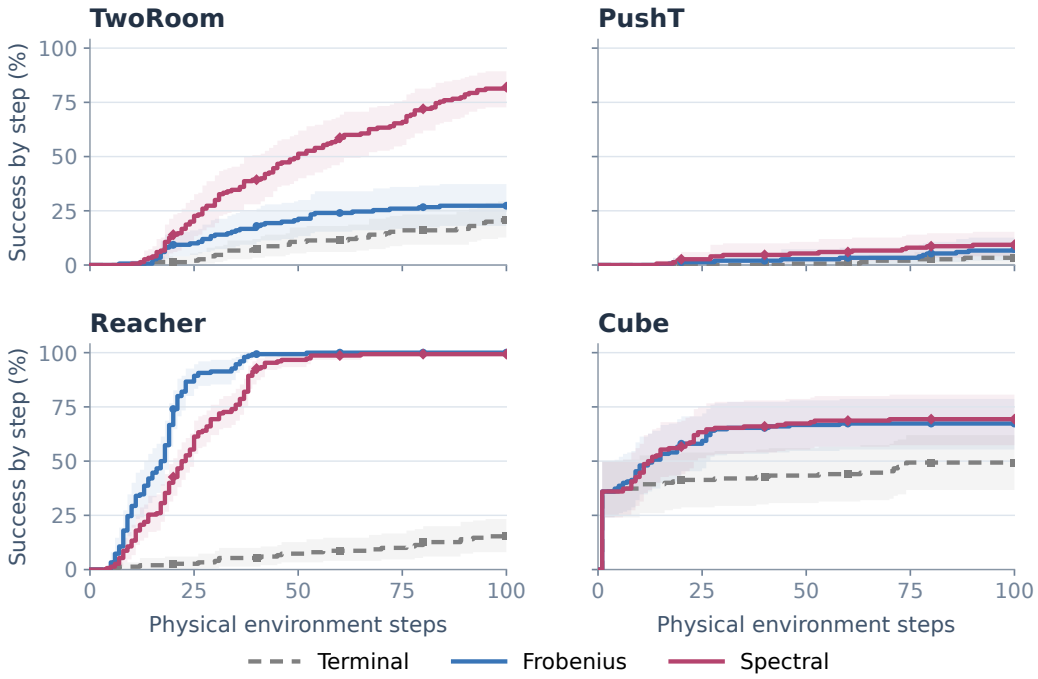

Figure 2: **Full-trajectory objectives sustain control beyond endpoint scoring.** Cumulative success within the 100-step environment budget at $H = 10$, using the same complete four-task panel as Table 1. Each curve averages 50 episodes and three planner seeds; shading gives pointwise 95% episode-bootstrap intervals with seeds kept together. Spectral gives the largest separation on TwoRoom, while Reacher saturates for both trajectory objectives and the gaps on PushT and Cube are smaller.

The gain is strongly task-dependent (Table 3). TwoRoom accounts for most of the aggregate improvement, whereas the contrasts on the other tasks are small or mixed. On the reserved Reacher episodes, Spectral reaches 100.0%, 99.3% and 91.3% at $H = 5, 10, 15$; at $H = 15$, Frobenius is numerically higher (97.3% versus 91.3%). Appendix C reports leave-one-task-out sensitivity, and Appendix D gives the paired intervals and statistical tests. The separate 80-episode replication under the original $H = 5/10$ configuration also gives a positive Spectral–Frobenius contrast of 12.08 points $[6.88, 17.50]$ (Appendix E).

Spectral is numerically highest in 10 of twelve primary settings. This count is distinct from statistical evidence: in the exploratory 24-contrast Holm analysis, 3 comparisons against Frobenius and 9 against Terminal pass the adjusted threshold. Appendix D gives all intervals and the separate prespecified six-contrast Reacher family.

5.2 Extending the planning horizon

We first isolate the effect of extending prediction and action search by holding the solver and executed prefix fixed (Figure 3, left). Terminal falls from 49.2% at $H = 5$ to 22.0% at $H = 10$ and 16.3% at $H = 15$; Spectral obtains 61.2%, 64.3% and 58.2%, respectively. Increasing H expands the forecast and action-search problem without changing the goal or physical budget. Spectral retains useful performance in this longer, higher-dimensional planning problem, although its success does not increase monotonically. The prespecified $H = 5/10$ Spectral–Frobenius contrast in this uniform panel is 13.75 points $[11.00, 16.50]$.

The selected and uniform panels differ in both configurations and episode samples, so their difference is not an estimate of the effect of tuning. Appendix D reports a separate, partial same-episode configuration diagnostic.

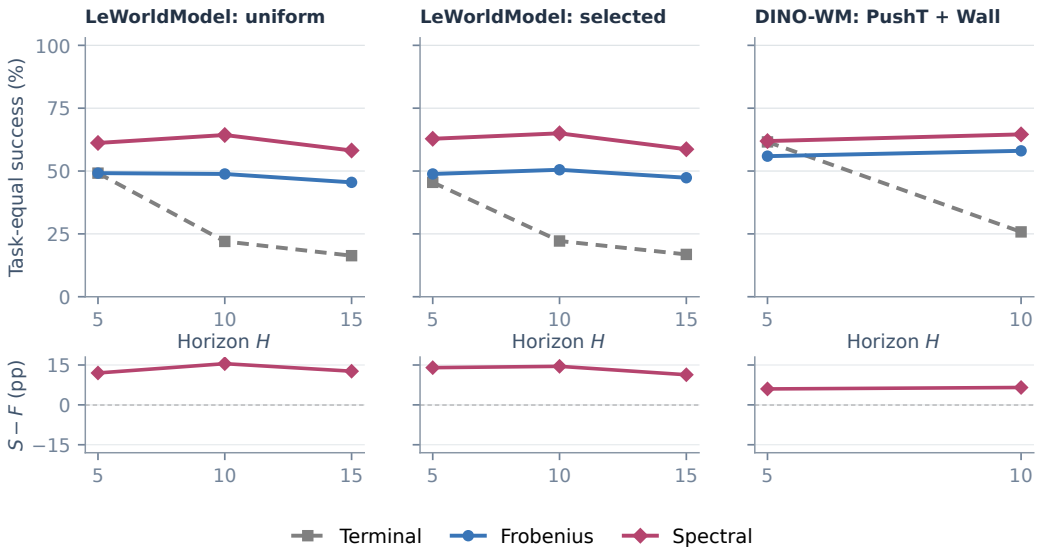

Figure 3: Observed task-equal success by planning horizon (top) and the corresponding Spectral–Frobenius difference in percentage points (bottom). The two LeWorldModel panels use distinct 200-episode allocations: the uniform panel fixes the search settings across horizons, while the selected panel uses the task-specific configurations in Table 6; its Reacher episodes are newly reserved and the other task samples are retained. DINO-WM averages offset-50 PushT and native-goal Wall equally. Goals and the 100-step execution budget are fixed within each panel; larger H extends prediction and action search. Points are observed means without confidence intervals; $H = 15$ and the DINO-WM extension are exploratory.

5.3 Transfer to DINO-WM

The DINO-WM grid in Table 4 reproduces the separation from Terminal at $H = 10$. Spectral reaches 79.2% and 29.2% on PushT at offsets 25 and 50, versus Terminal's 25.0% and 6.2%; Wall reaches 100.0% versus 45.3%. Frobenius also improves over Terminal, with Spectral numerically highest in each of these settings.

Table 4: **DINO-WM success rates (%).** Each cell uses three paired seeds, with 16 matched-start PushT trajectories or 50 Wall episodes. Bold marks numerical maxima. Appendix G gives the original-start counts and the interval protocol for the matched-start comparison.

Task	H	Goal offset	Terminal	Frobenius	Spectral
PushT	5	25	**75.0**	66.7	68.8
PushT	10	25	25.0	68.8	**79.2**
PushT	5	50	**29.2**	**29.2**	**29.2**
PushT	10	50	6.2	18.8	**29.2**
Wall	5	native	94.0	82.7	**94.7**
Wall	10	native	45.3	97.3	**100.0**

The design-specified task-equal $H = 10$ pool of offset-50 PushT and Wall gives a Spectral–Terminal difference of 38.79 points $[29.04, 49.71]$ and a Spectral–Frobenius difference of 6.54 points $[-0.04, 13.79]$. Evidence is therefore clearer for the advantage over endpoint scoring. Offset 50 aligns PushT's recorded goal time with the nominal forecast span; at offset 25, endpoint timing is an additional possible explanation for Terminal's weakness.

Short-horizon results limit this transfer: Terminal leads on matched-start PushT at offset 25 (75.0% versus 68.8%), and the common three-seed offset-50 cell ties at 29.2%. Appendix G gives the

Table 5: **Configuration, optimizer and geometry controls.** Task-equal Spectral differences in percentage points with original 95% episode-bootstrap intervals. n counts distinct episodes across four tasks. All rows average $H = 5, 10$ except the selected panel ($H = 5, 10, 15$); samples overlap across some controls.

Control	Comparator	n	Spectral difference [95% interval]
Selected configuration	Frobenius	200	$+13.3$ $[+10.7, +16.0]$
Uniform configuration	Frobenius	200	$+13.8$ $[+11.0, +16.5]$
Independent replication	Frobenius	80	$+12.1$ $[+6.9, +17.5]$
Equal step-size grids	Frobenius	200	$+13.8$ $[+11.2, +16.3]$
Matched subset: SPG	Frobenius	100	$+14.5$ $[+10.7, +18.3]$
Matched subset: CEM	Frobenius	100	$+18.5$ $[+14.5, +22.3]$
Max-row comparator	Max-row	100	$+30.2$ $[+24.7, +35.7]$
Schatten-4 comparator	Schatten-4	100	-1.5 $[-4.0, +1.0]$

formal counts, paired intervals and original-start protocol. The clearest transfer is the advantage over endpoint scoring at longer horizons; the Frobenius comparison remains closer.

5.4 Testing residual geometry

Table 5 summarizes configuration, optimizer, and matrix-objective controls.

Optimizer dependence. On a matched 100-episode subset, Spectral–Frobenius gains are 14.5 points $[10.7, 18.3]$ under SPG and 18.5 points $[14.5, 22.3]$ under CEM. These are within-solver comparisons, with different computation budgets across solvers. Equal development-only step-size grids also retain a positive 13.8-point difference $[11.2, 16.3]$.

Spectral alternatives. Spectral improves over max-row scoring by 30.2 percentage points, whereas its contrast with Schatten-4 is -1.5 points (95% interval: $[-4.0, 1.0]$). Thus the experiments support considering residual spectra beyond total energy, but do not resolve whether retaining only the leading singular value is preferable to aggregating several dominant modes. Appendix F reports these controls with their sample scopes.

Physical execution diagnostic. A separate mechanism diagnostic scores 128 common proposals on 80 episodes before gradient refinement. Each selected proposal is replayed open-loop for all $5H$ physical steps (25 or 50 steps); the diagnostic reports a $+3.1$-percentage-point Spectral–Frobenius difference in selected-candidate success $[0.0, 6.2]$. The objectives choose the same candidate in 47.5% of the 160 episode–horizon cases. Figure 4 reports the post-hoc rank-correlation heatmap: in TwoRoom, lower spectral scores computed from imagined or executed encodings accompany lower terminal physical costs, while row-wise feature permutation removes this association. For Reacher, the imagined and executed score correlations with physical cost are both weak. These diagnostics characterize candidate ranking without establishing a causal decomposition of the MPC gain.

6 Discussion and Limitations

The strongest gains occur on TwoRoom, where residual geometry also has the clearest association with physical progress. Reacher and short-horizon DINO-WM provide complementary cases in which the ranking differs, indicating that residual-spectrum aggregation interacts with the task and representation.

Several properties characterize this design choice. The objective captures cross-time alignment while remaining sensitive to feature scaling and indifferent to row order. Competitive Schatten-4 performance suggests that useful aggregation is not unique to the largest singular value. These observations point to residual geometry as a practical planning dimension and motivate future criteria for selecting the aggregation that matches a task.

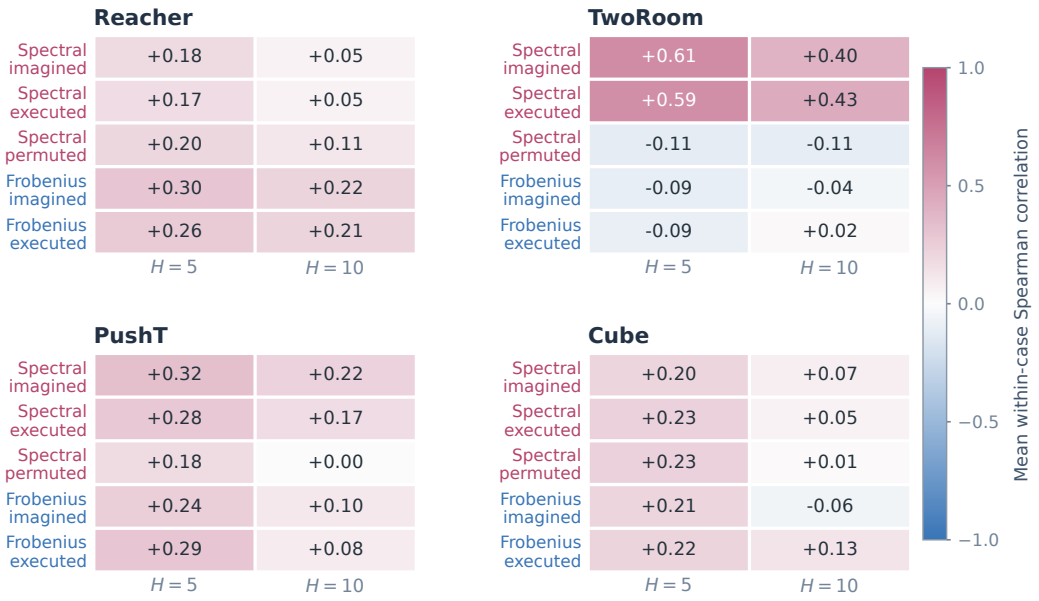

Figure 4: **Alignment between residual scores and physical costs varies by task.** Mean within-case rank correlations use eight presampled random candidates on 20 episodes per task at each horizon; Cube has 17 and 16 defined correlations. Positive values associate lower scores with lower physical cost. "Executed" encodes actual trajectories and is unavailable to the planner. "Permuted" independently shuffles feature coordinates within each residual row while preserving its energy.

Longer horizons increase forecast and search burden; the fixed task set, small DINO-WM PushT sample, and Cube restoration protocol bound the present evaluation. Uniform-setting and within-solver controls show that the aggregate pattern is not tied to a single solver configuration.

7 Conclusion

We introduced a spectral objective for planning with frozen world models. By aggregating goal-relative residuals through their dominant singular value, it distinguishes trajectories that endpoint distance and total residual energy score identically. Experiments show task-dependent improvements over Frobenius scoring on LeWorldModel and an advantage over endpoint scoring in longer-horizon DINO-WM settings. Competitive Schatten-4 results suggest that useful aggregation is not unique to the leading mode. Together, these findings make residual-spectrum aggregation a concrete design choice for world-model planning without additional model training.

AI USE STATEMENT

Generative AI tools were used only for language refinement in the Introduction, Related Work, and Experiments sections, as well as in the Discussion and Limitations section, primarily to improve grammatical correctness, clarity, and overall readability. All scientific ideas, analyses, and contributions presented in the paper were developed by the authors. Any AI-assisted revisions were carefully reviewed and validated by the authors.

REFERENCES

Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J. Zico Kolter. Differentiable MPC for end-to-end planning and control. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://papers.neurips.cc/paper_files/paper/2018/hash/ba6d843eb4251a4526ce65d1807a9309-Abstract.html.

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

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

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rk49Mg-CW.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2404.08471.

Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988. doi: 10.1093/imanum/8.1.141.

Ernesto G. Birgin, José Mario Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000. doi: 10.1137/S1052623497330963.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, pp. 4759–4770, 2018.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1174–1183. PMLR, 2018. URL https://proceedings.mlr.press/v80/denton18a.html.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation*, pp. 2786–2793, 2017.

Christopher Grimm, Andre Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

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

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

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

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2013. doi: 10.1017/CBO9781139020411. URL https://www.cambridge.org/highereducation/books/matrix-analysis/FDA3627DC2B9F5C3DF2FD8C3CC136B48.

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

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

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-image representations and rewards for robotic control. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23301–23320. PMLR, 2023a. URL https://proceedings.mlr.press/v202/ma23b.html.

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. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.00030.

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 arXiv:2605.21800*, 2026a. URL https://arxiv.org/abs/2605.21800.

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

Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909. PMLR, 2023. URL https://proceedings.mlr.press/v205/nair23a.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

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 *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1049–1065, 2021.

Reuven Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1:127–190, 1999. doi: 10.1023/A:1010091220143.

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

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

G. A. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992. doi: 10.1016/0024-3795(92)90407-2. URL https://www.sciencedirect.com/science/article/pii/0024379592904072.

Hanzhe You, Yonggang Zhang, Maohao Ran, Zhiqin Yang, Zhenyuan Zhang, Wei Xue, Jun Song, Xinmei Tian, and Yike Guo. A control theory of predictability in latent world models. *arXiv preprint arXiv:2607.10362*, 2026. URL https://arxiv.org/abs/2607.10362v1.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 79115–79135, 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

A SPECTRAL IDENTITIES

For $\mathbf{E} \neq 0$, let $G = \mathbf{E}\mathbf{E}^\top$. The Rayleigh quotient and $\lambda_{\max}(G) \leq \mathrm{tr}(G)$ give

$$H^{-1} \max_t \|\mathbf{e}_t\|_2^2 \leq J_{\mathrm{op}} \leq J_{\mathrm{F}}, \qquad \frac{1}{\min(H,d)} \leq a(\mathbf{E}) \leq 1.$$

The lower bound on a follows because the trace is at most $\mathrm{rank}(\mathbf{E})$ times the largest eigenvalue. For two rollouts with Frobenius scores $F_A > F_B > 0$, Spectral prefers A precisely when $a_A/a_B < F_B/F_A$. This expresses an energy–dispersion tradeoff, not a physical reachability guarantee. The Gram matrix has size $H \times H$; forming it costs $O(H^2 d)$ and its eigendecomposition costs $O(H^3)$, in addition to the model rollout and action differentiation.

An equivalent time-domain form is $J_{\mathrm{op}} = H^{-1} \max_{\|\mathbf{q}\|=1} \|\sum_t q_t \mathbf{e}_t\|^2$. The coefficients q_t can be signed and are not time probabilities.

B IMPLEMENTATION AND EVALUATION SETTINGS

Frozen models and normalized actions. The LeWorldModel adapter uses native current/goal image preprocessing and the last encoded observation as its initial latent. Encoder and dynamics parameters remain frozen. Physical actions are transformed coordinatewise as $u = (a - \mu)/s$, including both action bounds. The evaluation pipeline fits μ, s to all finite action rows of the released task dataset; these statistics are shared by all objectives and include records later selected for evaluation. Excluding episodes from local tuning does not exclude them from action-scaler estimation or world-model training. Privileged states, success labels and recorded future controls are never candidate-scoring inputs.

Initialization and case-local randomness. The default uses 128 scrambled Sobol proposals transformed to normal coordinates, with five control knots per physical action coordinate. Linear interpolation supplies H model-time values, each repeated across the five physical actions in a block and clipped to the normalized action box. The first candidate is clipped normalized zero, corresponding to the clipped dataset action mean in physical coordinates. Each objective selects its own four lowest-score starts. Subsequent optimization varies all $5H$ physical action vectors independently.

For planner seed s_0 and decision index j within a goal image, the random seed is $(s_0 + h_g + 1000003j) \bmod (2^{31} - 1)$, where h_g is the integer encoded by the first eight hexadecimal digits of the processed goal-image SHA256. This stream supplies the proposal seed, so another episode's early success does not shift its randomness. Shared decision indices need not have shared observations after actions diverge.

Table 6 specifies the configurations used in Table 1. Mixed initialization allocates one quarter, one half and one quarter of the proposals to 1, 5 and H knots, respectively. With warm initialization, later decisions replace the final quarter of the pool with continuations of the preceding unexecuted plan. This pool includes exact held-tail and zero-padded continuations; additional candidates use smooth perturbations clipped to $[-3, 3]$ and scaled by 0.25 in normalized coordinates. The first decision uses the ordinary pool. Each objective retains its own preceding plan, so the paired rules and random streams need not produce identical warm candidates. Proposal scoring is batched at 512 candidates and ranked globally.

Projected action search. Each start maintains a feasible action vector x_k, gradient g_k, and positive diagonal metric D_k. The default is $D_k = I$. For the RMS variant, $v_k = 0.9 v_{k-1} + 0.1 g_k^2$, with $v_{-1} = 0$; its square root is divided by its coordinate mean (floored at 10^{-12}) and clipped to $[0.01, 100]$ to obtain the diagonal of D_k. The time variant first averages v_k across packed action coordinates at each model step. These metrics change the update direction, not the trajectory objective. Table 7 identifies the selected rules.

The initial scale is $\alpha_0 = \mathrm{clip}(\eta_0 / \max(\|D_0^{-1} g_0\|_\infty, 10^{-12}), 10^{-6}, 10^3)$. Later scales use $s_k^\top D_k s_k / (s_k^\top y_k)$, where $s_k = x_k - x_{k-1}$ and $y_k = g_k - g_{k-1}$ (Barzilai & Borwein, 1988). A denominator at most 10^{-12} retains the previous scale; all scales are clipped to the same range. For

Table 6: Frozen configurations in Table 1, applied to all three objectives. R is the maximum executed prefix in model blocks; each block contains five physical actions. The original uniform-configuration panel uses the default 10/4/128/R=5 throughout. Reacher update rules are specified in Table 7.

Task	H	Max. updates	Starts	Proposals	R	Warm	Mixed
Reacher	5	20	4	128	3	yes	no
Reacher	10	20	8	1024	3	yes	no
Reacher	15	20	4	128	3	yes	no
TwoRoom	5	10	4	128	5	no	no
TwoRoom	10	10	4	128	5	no	no
TwoRoom	15	10	4	128	5	no	no
PushT	5	10	4	128	3	no	no
PushT	10	10	4	128	5	no	no
PushT	15	10	4	128	5	no	no
Cube	5	10	16	512	5	no	no
Cube	10	10	4	128	5	no	no
Cube	15	10	4	128	5	no	no

Table 7: Reacher update rules selected on exposed development episodes and frozen before the new sample. Proposal and execution budgets appear in Table 6; the same rules apply to all objectives. All selected configurations optimize the full physical action sequence.

H	Update	Metric	η_0
5	Projected	time	0.2
10	Projected	identity	0.05
15	Projected	rms	0.2

$r = 0, \ldots, 15$, a start accepts the first finite $x_{k,r} = \Pi_{\mathcal{B}}(x_k - 2^{-r}\alpha_k D_k^{-1} g_k)$ satisfying

$$J(x_{k,r}) \leq J(x_k) + 10^{-4} g_k^\top (x_{k,r} - x_k).$$

Four consecutive factors are evaluated together, preserving their acceptance order. Failure retains the feasible incumbent. Updates stop at the declared limit, when all starts satisfy $\|x_k - \Pi_{\mathcal{B}}(x_k - g_k)\|_\infty \leq 10^{-5}$, or when no active start accepts an update. The lowest-score feasible incumbent is returned. The finite-budget, potentially nonsmooth objective does not inherit a smooth SPG convergence guarantee. No action, smoothness or arrival penalty is added.

Task predicates and timing. Reacher requires every joint-position error below 0.05 radians; TwoRoom requires position distance below 16 environment units; PushT requires joint agent/block position-error norm below 20 units and wrapped block-angle error below $\pi/9$. Cube checks block-position distance at most 0.04 metres, without orientation or arm-pose criteria. LeWorldModel records success during execution without retrospectively adding time-zero success. All studies retain the native task predicates.

One complete warmup decision is excluded from timing and restores the random generator state before measured planning. CUDA synchronization brackets each solve. Planning time includes encodings, initialization and optimization, and excludes model loading and simulator execution. Episode totals include all MPC decisions; differing success and stopping times change these totals. Timing measurements use two CPU math threads, matrix-multiplication TF32 disabled and cuDNN TF32 enabled.

Cube restoration scope. The native state setter restores positions and velocities while retaining auxiliary simulator state; all main results use this restoration. A sensitivity check on 16 episodes gave identical success labels and first-success-or-budget values after clearing the retained state.

C Evidence lineage and task sensitivity

We assign stable panel identifiers to prevent samples and tuning stages from being conflated. P_{uni} is the original uniform panel (50 episodes per task); P_{val} is a fixed validation panel (50 episodes per

task); and P_{sel} combines 50 held-out Reacher evaluation episodes with the three non-Reacher task rows from P_{val}. P_{sel} shares those 150 non-Reacher episodes with P_{val} and uses a disjoint Reacher split. P_{uni} and P_{sel} have no episode-ID overlap. An independent replication is disjoint from both panels. The selected configurations were chosen on a development split using Spectral performance, then frozen and applied to all three objectives; P_{uni} uses the original fixed configuration. Action-scaler statistics and world-model training data follow the release protocol described in Appendix B.

The task-sensitivity table in the main paper gives the corresponding aggregate decomposition. The leave-one-task-out sensitivities below reuse the same fixed task panels and are descriptive rather than intervals over unseen tasks.

Table 8: Post-hoc leave-one-task-out Spectral–Frobenius differences (percentage points), with unadjusted 95% within-task episode-bootstrap intervals. Tasks retained in each row receive equal weight. These sensitivities concern the fixed task set and are not confirmatory tests or intervals over unseen tasks.

Omitted task	Selected ($H = 5, 10, 15$)	Uniform ($H = 5, 10$)	Replication ($H = 5, 10$)
Reacher	$+18.22$ $[+14.67, +21.70]$	$+18.00$ $[+14.56, +21.44]$	$+20.28$ $[+14.17, +26.39]$
TwoRoom	$+1.63$ $[-0.22, +3.63]$	-0.00 $[-2.22, +2.33]$	-2.50 $[-6.94, +1.94]$
PushT	$+16.81$ $[+13.41, +20.07]$	$+18.56$ $[+14.89, +22.11]$	$+15.00$ $[+8.33, +21.94]$
Cube	$+16.44$ $[+13.11, +19.70]$	$+18.44$ $[+15.22, +21.56]$	$+15.56$ $[+9.44, +21.67]$

D STATISTICS AND CONFIGURATION CONTROLS

Table 1 contains 50 distinct episodes per task, reused across three horizons and three planner seeds. Reacher uses a held-out evaluation split; the other tasks use their fixed evaluation samples. For horizons $\mathcal{H}$, let D_{ti} denote the Spectral-minus-baseline success difference averaged over $\mathcal{H}$ and the three seeds on episode i of task t. The task-equal estimate is $\widehat{\Delta} = \frac{1}{4}\sum_{t=1}^{4}\frac{1}{50}\sum_{i=1}^{50} D_{ti}$. We use 20,000 within-task episode-bootstrap draws with generator seed 2026092617, keeping all methods, seeds and included horizons together. The 2.5th and 97.5th percentiles form the reported intervals. The four tasks are fixed; these intervals do not quantify variation across new task families. Original-panel intervals use their separately recorded analysis.

Table 9 reports every cellwise contrast. Exact two-sided sign-flip tests jointly exchange an episode's two objective labels across all three seeds. The null assumes conditional sign symmetry of paired differences, rather than only equality of their means. Holm adjustment uses $\alpha = 0.05$ across 24 comparisons. This family is exploratory because the three retained task panels were already observed when the integrated analysis was specified. Failure to reject does not establish equivalence.

Table 9: All paired contrasts from Table 1. Intervals are unadjusted; p_{H} is the two-sided exact episode-level sign-flip p-value after Holm correction across 24 tests. This 24-test summary is exploratory because the non-Reacher outcomes were already observed; the prospective six-test Reacher family is reported separately.

Task	H	vs. F, pp [95% interval]	p_{H}	vs. T, pp [95% interval]	p_{H}
Reacher	5	+2.0 [+0.0, +5.3]	1.000	+20.0 [+13.3, +28.0]	< 0.001
Reacher	10	-0.7 [-2.0, +0.0]	1.000	+84.0 [+76.0, +91.3]	< 0.001
Reacher	15	-6.0 [-10.7, -2.0]	0.215	+81.3 [+74.0, +88.0]	< 0.001
TwoRoom	5	+42.7 [+32.0, +53.3]	< 0.001	+34.0 [+23.3, +45.3]	< 0.001
TwoRoom	10	+54.7 [+42.7, +66.7]	< 0.001	+61.3 [+51.3, +71.3]	< 0.001
TwoRoom	15	+47.3 [+36.7, +58.0]	< 0.001	+57.3 [+46.7, +68.0]	< 0.001
PushT	5	+5.3 [+0.7, +10.7]	0.609	+6.7 [+1.3, +12.7]	0.459
PushT	10	+2.0 [-3.3, +7.3]	1.000	+6.0 [+1.3, +12.0]	0.492
PushT	15	+0.7 [-4.7, +6.7]	1.000	+7.3 [+2.7, +12.7]	0.094
Cube	5	+6.0 [+0.0, +12.7]	0.766	+8.7 [+4.0, +14.0]	0.013
Cube	10	+2.0 [-3.3, +8.0]	1.000	+20.0 [+10.0, +30.7]	0.006
Cube	15	+3.3 [-0.7, +8.0]	1.000	+21.3 [+12.0, +31.3]	< 0.001

Table 10: Successes out of 50 for seeds 42, 123 and 2026, and mean planning seconds per episode. T/F/S denote Terminal/Frobenius/Spectral. Seeds repeat the same episodes. Times include all MPC decisions and can differ because trajectories and stopping times differ.

Task	H	T counts	F counts	S counts	T sec.	F sec.	S sec.
Reacher	5	40,40,40	49,49,49	50,50,50	1.010	0.393	0.488
Reacher	10	8,7,8	50,50,50	50,49,50	3.791	0.863	1.197
Reacher	15	6,5,4	49,48,49	45,47,45	4.141	1.267	1.637
TwoRoom	5	20,20,17	16,16,12	36,38,34	0.396	0.416	0.398
TwoRoom	10	11,11,9	14,16,11	42,41,40	0.801	0.776	0.625
TwoRoom	15	6,4,5	8,12,10	32,39,30	1.207	1.172	0.969
PushT	5	1,2,4	2,3,4	5,4,8	0.825	0.805	0.954
PushT	10	1,1,3	5,1,5	5,3,6	0.861	0.839	0.924
PushT	15	0,1,0	4,4,3	4,6,2	1.226	1.171	1.308
Cube	5	31,27,31	31,30,32	34,37,31	0.395	0.371	0.401
Cube	10	25,22,27	34,33,34	36,33,35	0.566	0.434	0.475
Cube	15	23,21,26	33,33,31	34,36,32	0.834	0.674	0.679

Targeted Reacher development. At each horizon, twelve configurations were screened on 32 exposed episodes (seed 42); the two leaders and incumbent were evaluated on all 66 development episodes (seed 123). Spectral success determined selection, with ties favoring the incumbent and then lower planning time. The selected settings were frozen and applied to every objective. Search varied initialization, step scales, action parameterization, updates and executed prefix, not the model or score.

The Reacher evaluation split is disjoint from the episodes used for local configuration selection and prior objective evaluation. Before evaluating this split, the horizon-averaged Spectral–Frobenius contrast was designated primary, Terminal secondary, with a six-test Holm family (Table 11). Intervals use 20,000 episode-bootstrap draws (generator seed 2026092607).

Table 11: Prospectively specified six-contrast Reacher family on the new sample. Exact two-sided episode sign-flip tests exchange all three seeds jointly; Holm controls the family at $\alpha = 0.05$ under conditional sign symmetry.

H	Baseline	Holm p (six tests)
5	Frobenius	1.000
5	Terminal	< 0.001
10	Frobenius	1.000
10	Terminal	< 0.001
15	Frobenius	0.059
15	Terminal	< 0.001

E ORIGINAL UNIFORM-CONFIGURATION CONTROL

A uniform-configuration control on 200 episodes uses the default solver, initialization and five-block execution prefix at every horizon. Its prespecified $H = 5/10$ Spectral–Frobenius difference is 13.75 points $[11.00, 16.50]$; the corresponding descriptive Terminal contrast is 27.17 points $[24.33, 30.08]$. The initially-unsatisfied subset retains a 13.7-point difference $[10.3, 17.2]$ against Frobenius. These summaries refer to the original panel and are separate from the selected-configuration panel.

An independent replication uses 20 additional episodes per task while retaining the original $H = 5/10$ settings. Its Spectral–Frobenius contrast is 12.08 points $[6.88, 17.50]$. It remains separate from both 200-episode panels and supports a positive aggregate with task dependence.

F OPTIMIZER AND RESIDUAL-GEOMETRY CONTROLS

The controls here use the original episode allocations and are not new selected-panel outcomes. Independent step-size tuning gives every objective the grid $\eta \in \{0.05, 0.2, 0.8\}$ on the same development cases and two seeds; its confirmation contrast remains 13.8 points $[11.2, 16.3]$. CEM uses 30 rounds of 300 box-clipped Gaussian candidates, 30 elites and a coordinatewise standard-deviation floor of 10^{-3}. It retains the best sample and the four common-pool starts. Table 12 compares all three objectives under each optimizer on the same 25 episodes per task; CEM and SPG have different computation budgets.

Table 12: Objective-by-optimizer success rates on the same 100 episodes at both horizons and three search seeds. Each cell has 75 executions on 25 episodes. SPG outcomes are reused from the original uniform-configuration panel; matching adds no observations. The two solvers have different computational budgets.

Task	H	SPG Spectral	SPG Frobenius	SPG Terminal	CEM Spectral	CEM Frobenius	CEM Terminal
Reacher	5	92.0%	90.7%	93.3%	94.7%	89.3%	92.0%
Reacher	10	92.0%	92.0%	13.3%	92.0%	93.3%	34.7%
TwoRoom	5	66.7%	8.0%	17.3%	70.7%	4.0%	16.0%
TwoRoom	10	72.0%	8.0%	14.7%	89.3%	10.7%	12.0%
PushT	5	12.0%	14.7%	16.0%	8.0%	13.3%	18.7%
PushT	10	14.7%	13.3%	5.3%	22.7%	18.7%	0.0%
Cube	5	65.3%	69.3%	62.7%	65.3%	68.0%	66.7%
Cube	10	66.7%	69.3%	57.3%	70.7%	68.0%	56.0%

The matrix controls use max-row score $\max_t \|\mathbf{e}_t\|^2/H$ and Schatten-4 score $\sqrt{\sum_i \sigma_i^4}/H$ (Table 13). Squared versus unsquared spectral scoring changes success by -0.5 points $[-2.9, 1.8]$ in its separate 64-episode sensitivity panel, providing no clear difference there. On the original 200 episodes, post-hoc equally tuned comparisons yield Spectral advantages of 22.42 points over a linearly time-weighted residual-energy sum and 32.00 points over spectral scoring with independently permuted feature coordinates in each row. Permutation preserves row energies but changes cross-time feature correspondence and the finite-budget optimization landscape; it is not a causal decomposition.

Table 13: Matrix controls on the original 100-episode optimizer subset, two horizons and three seeds (75 executions per cell).

Task	H	Spectral	Frobenius	Max-row	Schatten-4
Reacher	5	92.0%	90.7%	54.7%	92.0%
Reacher	10	92.0%	92.0%	45.3%	93.3%
TwoRoom	5	66.7%	8.0%	2.7%	61.3%
TwoRoom	10	72.0%	8.0%	1.3%	74.7%
PushT	5	12.0%	14.7%	8.0%	16.0%
PushT	10	14.7%	13.3%	8.0%	21.3%
Cube	5	65.3%	69.3%	61.3%	68.0%
Cube	10	66.7%	69.3%	58.7%	66.7%

The common-candidate diagnostic uses 20 mechanism episodes per task at $H = 5/10$, scores 128 shared proposals, and executes selected plans open-loop for all $5H$ physical steps. Its terminal physical cost is dimensionless and normalized by the native success tolerance: $\max_j |q_j - q_j^*|/0.05$ for Reacher, $\|p - p^*\|_2/16$ for TwoRoom, $\max(\|[p_{\text{agent}}, p_{\text{block}}] - [p^*_{\text{agent}}, p^*_{\text{block}}]\|_2/20, |\operatorname{wrap}(\theta - \theta^*)|/(\pi/9))$ for PushT, and $\|p_{\text{block}} - p^*_{\text{block}}\|_2/0.04$ for Cube. Eight presampled random candidates per case supply the correlations in Figure 4; Spearman ties use average ranks, and a zero-variance score or cost is undefined. Reacher, TwoRoom, and PushT have 20/20 defined cases at both horizons; Cube has 17/20 at $H = 5$ and 16/20 at $H = 10$. The analysis is post-hoc, and executed encodings are unavailable to planning.

G DINO-WM PROTOCOL AND ORIGINAL-START RESULTS

PushT and Wall use frozen DINOv2 ViT-S/14 patch encoders and predictors, without a decoder. Development/confirmation splits contain 5/16 PushT trajectories and 16/50 Wall episodes. Native rollout and action-gradient parity checks passed before evaluation.

Future visual embeddings are flattened to length D, with residual rows $e_t = D^{-1/2}(\mathrm{vec}(z_t^{\mathrm{visual}}) - \mathrm{vec}(g^{\mathrm{visual}}))$. Input history, proprioception and action tokens do not enter the score, while native context and future action dependencies remain in the model and its gradients. The shared SPG setup uses at most ten updates, four starts, 128 proposals, five knots, $\eta = 0.05$ and four line-search trial factors. Each model step contains five physical actions; at most five blocks execute within the 100-step budget. Objectives share the normalized box $[-3, 3]$ and released action scaler (the PushT wrapper advertises absolute positions but executes relative actions). Batches contain at most eight plans, and case-local seeds hash the case cache, planner seed and decision index. The original study does not tune objectives separately.

The matched-start extension revises a small number of confirmation and development starts to admit offset-50 goals, then shares these starts across offsets. Table 4 reports the resulting common three-seed grid. The original-start protocol uses different starts and is kept separate rather than pooled with this comparison; Wall $H = 5$ reuses its original outcomes.

DINO-WM intervals use 10,000 within-task episode-bootstrap draws. The $H = 10$ pool weights offset-50 PushT and Wall equally, keeping paired seeds together.

PushT uses recorded future controls only to construct the target; its native restoration advances one tick. Wall uses reserved layouts with native seeded start/goal sampling. PushT requires joint agent/block position error below 20 and wrapped block-angle error below $\pi/9$; Wall requires position error below 4.5. DINO-WM records time-zero success, unlike the LeWorldModel evaluator. These predicates and restoration differences preclude direct cross-family rate comparisons; the samples are never pooled.