A Predeclared Negative Result for Paid Calibration of Token-Sparse Planning Readouts in a Frozen JEPA World Model

Anonymous submission

2026-10-25

Draft status. This is an unsubmitted research draft prepared with substantial AI assistance (Section 10). It reports model-internal results for one frozen public checkpoint on synthetic scenes. It does not establish physical control, closed-loop planning benefit, an end-to-end speedup, novelty, or venue acceptance, and nothing in it is a submission. A Chinese companion abstract is provided as a separate file (`abstract-zh.md`).

Evidence inventory. All 11 evidence layers, their predeclared protocols, immutable snapshots, independent-verification reports and SHA-256 bindings are listed in `claims-ledger.md` and the machine-readable manifests under `outputs/research-lab/world_models/`. The body of this paper cites each numeric result together with its artifact/verification hashes; the full per-file hash inventory is reproduced in the claims ledger and the anonymous source bundle, rather than in this header.

Abstract

We report a predeclared negative result for paid calibration of token-sparse planning readouts in a frozen JEPA world model. Planning with a frozen visual world model is usually scored by a full-token cost functional, but deploying the model on a fixed subset of $K = 128$ of 256 tokens changes both *which* token costs are averaged (projection) and *which* tokens the predictor sees (sparse prediction). We study this readout problem in the official frozen JEPA-WMs "Wall" model (DINOv2 ViT-S/14 encoder, 17.6M-parameter AdaLN predictor, 256 tokens, horizon 6) with a position-preserving sparse adapter. In a predeclared four-scene diagnostic, projecting full-token terminal costs onto a gradient-selected 128-token subset inverted 793/1984 comparable action pairs (39.97%); keeping that scoring subset fixed but running the frozen predictor on it inverted 174/1984 (8.77%). The effects are non-additive and must not be added, and the combined ranking left the selected action at zero full-cost gap in three of four scenes. We then specified before execution a 16-context development pilot in which nine paid full-model action queries fit a regularized minimax simplex QP over the 128 per-token sparse losses, compared with uniform sparse scoring, full-integrand positive-cubature weights, anchored-kernel scalar surrogates, and best-paid-pilot selection under the same nine paid labels. The predeclared seven-condition engineering gate returned `NO_GO_FIXED_RECIPE`: 3 of 7 conditions passed. The candidate's mean 64-unqueried-action full-reference gap was 0.076410 versus 0.029254 for the best paid comparator, Scalar anchored-kernel (full-sparse residual) (161.2% larger, where the gate required a 20% reduction); the normalized-gap improvement was -0.054599 (required +0.02); the union-73 gap was 0.071960 versus 0.049337; and the candidate had 6/6 wins/losses against the best paid pilot (at most two losses required). The full-integrand baseline (0.057841) and even the plain sparse mean (0.064461) were better than the candidate on the primary metric. We therefore report a predeclared negative result for this fixed nine-query recipe and a diagnostic account of token-sparse readout distortion. The failure stops this recipe in this setting; it does not show that calibration cannot work. An additional independently verified open-loop simulator anchor (the same fixed action bank executed for 30 official Wall environment steps per band; no CEM, no replanning, no closed-loop feedback) supports a restricted task-level claim: *on this fixed action bank, the full-token readout shows moderate-to-moderately-strong rank alignment with simulator*

task distance (not strong), and all sparse/calibrated readouts incur larger simulator task regret; this is not a planning-success or closed-loop claim. The full-token readout achieves a mean task regret of 1.353476 (median 0.000000) versus 3.956109 for the smallest sparse/calibrated mean observed (the candidate on the gradient selector), with model-cost/task-distance Spearman 0.485838 (median 0.532771). An exploratory replication on a second official checkpoint (`dino_wm_wall`, concat predictor; same Wall domain, not a second domain, not predeclared, not confirmatory) finds the same qualitative sparse-readout task-regret ordering on that bank (full-token mean evaluation task regret 1.081926 versus 5.764936 for random-K and 8.049391 for gradient-K), while the gradient-versus-random selector ordering is architecture-dependent; its implementation, metrics and final snapshot audit are independently verified PASS. A predeclared planner-level CEM comparison (fixed reduced 10×64 budget, open-loop 30-step execution, not the official CEM protocol and not closed-loop) gives the strongest task-level signal in this study: full-token plans reach 11/16 official successes (68.75%) versus 8/16 (50.00%) for gradient-K and 5/16 (31.25%) for random-K, with mean task distance 6.713579 versus 10.638763 and 14.804347; paired full-only successes are +3 versus gradient and +6 versus random. This result is fixed-budget and single-model, and each cell is a single unseeded CEM run, so it is reported as a descriptive planner-level task consequence under a reduced budget, not as a statistical-significance or method claim. Its independent verification is PASS: the saved metrics were reproduced with zero mismatch, and the independent CEM reimplementation preserved the aggregate direction in every run (full $\geq$ both sparse readouts; independent subset mean distances $2.17 < 7.40 < 11.89$), while per-context sparse single-run outcomes are unstable. At the success level no sparse readout reverses full in any context, but at the continuous-distance level gradient is closer on contexts 6,12,15 and random on 7,10, so the "no reversal" statement holds only at the success level. We do not claim that token-sparse planning degrades task success: COSTGRAD reports Wall parity at 128 tokens, and ARC-Bench shows that task success can mask ranking failure. We claim a readout-fidelity gap and a predeclared failure of a paid calibration to close it, with no physical, task-level, speed, or novelty claim. The candidate was better than the comparator in 2 of 16 paired contexts, worse in 7, and tied in 7, so the negative result is not borderline.

1 Introduction

Learned visual world models are increasingly used to propose and rank action sequences for control and planning [2, 20, 24]. In a token-based model, a candidate action sequence is rolled out into a sequence of visual token predictions, and the candidate is scored by a cost functional evaluated on those tokens. The computational cost of planning therefore scales with the number of scored tokens, the planning horizon, and the number of action candidates. A natural acceleration is to keep only $K < 256$ tokens per rollout — by pruning, merging, or selecting a fixed subset — and to score candidates using only the retained tokens. This is the setting we study: a *token-sparse planning readout* for a frozen model.

Token-sparse deployment changes two things at once. Prior work on token selection for visual world-model planning optimizes the first and reports end-task scores [22]. First, the scored functional changes: even if every full-token prediction were available and unchanged, averaging a K-token subset of per-token losses is a projection of the full cost and can reorder actions. Second, the predictions themselves change: a predictor that only ever sees K tokens produces different retained-token states, because attention and context are restricted. We call these the *projection effect* and the *sparse-prediction effect*. They are non-additive and, in the model we examine, they can point in opposite directions for the same action pair.

Task success is a weak instrument for readout fidelity: COSTGRAD reports Wall task success preserved at half the tokens with its selector [22], while ARC-Bench shows that task success can mask ranking failure entirely [23]. We therefore audit the readout directly and test, under a predeclared protocol, the natural remediation hypothesis: that a scoring-weight calibration fitted to a small number of paid full-model action contrasts closes the projection part of the gap. The predeclared experiment below shows that it does not; we report that negative result and the decomposition that makes it interpretable.

This paper is a model-fidelity study of that readout problem. The full model is used as a higher-compute computational reference, not as physical truth. Our study has five parts, in this order.

1. A four-scene diagnostic on the official frozen JEPA-WMs "Wall" checkpoint, in which full-token predictions,

projected full-token costs, and actual sparse-predictor costs are stored for the same 32 action bands per scene, so the projection and sparse-prediction effects can be counted separately and together. This diagnostic is complete and its numbers are reported in Section 5.1.

2. A predeclared 16-context *development* calibration pilot in which nine paid full-model action queries per context fit a regularized minimax simplex QP over the $K = 128$ per-token sparse losses, and the calibrated readout is compared on 64 unqueried audit actions with uniform sparse scoring, two full-integrand cubature baselines, two anchored-kernel scalar surrogates, and best-paid-pilot selection. The recipe, comparators, metrics, and gate were frozen before any outputs were generated (Section 4). Its results are reported in Section 5 (Section 5.2) under the frozen gate.
3. A descriptive *task-level Wall anchor*: the same fixed 73-band action bank is executed open-loop for 30 official vendor-environment steps per band, and simulator task distance is compared with the frozen readouts (Section 6). It adds task-level context to the readout study but is not CEM planning, not closed-loop feedback, and not a planning-success claim. Its independent verification has PASSED (1,168/1,168 trajectories exactly reproduced; zero metric mismatch).
4. An *exploratory replication* on a second official checkpoint, `dino_wm_wall` with a concat ViTPredictor, on the same Wall domain and action bank (Section 7). It is a second architecture, not a second domain, it is not predeclared or confirmatory, and it makes no CEM, closed-loop or method claim; the independent verifier reports PASS on implementation, metrics and the final snapshot audit.
5. A predeclared *planner-level CEM comparison* (G2) under a fixed reduced 10×64 budget with a dense floor gate, executing each final elite OPEN-LOOP for 30 environment steps; it is not the official 30×300 CEM and not closed-loop. Its independent verification is PASS (zero saved-metric mismatch; aggregate direction preserved in independent runs), with the stochastic single-run caveats recorded in Section 8.

Section 3 fixes notation and defines the protocol; Sections 4–7 describe and report the four parts; Section 9 separates what the artifacts establish from what they do not.

What this work is not. We make no claim of a new optimization principle, verified originality, or novelty: a capped-simplex minimax fit of positive quadrature weights, and scalar surrogate regression, are known tools, and the project that produced this draft has explicitly not cleared the combination as an original method. We also make no physical-control, closed-loop planning, end-to-end speed, or neural-network-training claim; no environment step is executed and no model parameter is updated anywhere in this study. Section 9 lists the claims we do not make in detail.

Contributions under the selected framing. Under framing B (negative result) the contribution is a measurement and evaluation contribution with three parts: (i) a hash-bound, predeclared decomposition of token-sparse readout distortion into a cost-projection component and a sparse-prediction component on the same frozen checkpoint, showing that the two are non-additive and that neither inversion count translates directly into chosen-action loss; (ii) a predeclared negative result, read out by the unmodified frozen readout, in which a fixed nine-query sparse-token contrast QP is dominated at the same paid-label budget by a full-integrand positive-cubature baseline and by a scalar full–sparse residual surrogate (and, on the primary selector, by the plain sparse mean); (iii) a descriptive, predeclared *task-level anchor* that executes the same fixed action bank open-loop in the official simulator and shows that the full-token readout has moderate-to-moderately-strong rank alignment with simulator task distance (not strong) while sparse and calibrated readouts incur larger simulator task regret on that bank (independently verified; no planning-success or closed-loop claim); (iv) an exploratory, independently verified replication on a second official checkpoint with a different action pathway (`dino_wm_wall`, concat predictor), which shows the same qualitative sparse-readout task-regret ordering on the same Wall bank and an architecture-dependent selector ordering, bounded as a same-domain architecture replication rather than a second-domain result; (v) a predeclared planner-level CEM comparison (fixed reduced 10×64 budget, dense floor gate, open-loop 30-step execution, not official CEM and not closed-loop) in which full-token plans succeed more often than sparse-readout plans (11/16 versus 8/16 and 5/16) with paired full-only successes +3/+6, reported as a fixed-budget descriptive task consequence whose independent verification is PASS (Section 8); and (vi) an explicit protocol, artifact set, and claims ledger that make the measurement reproducible and prevent the negative result from being restated as a positive method claim. This is not a new method, not a claim that token-sparse planning is harmful, and not a claim that calibration cannot work; it is a bounded, predeclared falsification study whose closest prior art is discussed in Section 2.

Transparency. The study was produced under a zero-additional-spend constraint with substantial AI assistance (Section 10). A resource-only protocol amendment after a clock stop, a pre-execution bookkeeping incident, and earlier

diagnostic runs are disclosed in Sections 4.3 and 9 rather than dropped. All quantitative claims are mapped to artifact paths and SHA-256 hashes in `claims-ledger.md`.

2 Related Work

This section positions the measurement against the closest prior art. We do not claim any component as new: token selection, post-compression functional calibration, positive weighted-subset reconstruction, and surrogate readout calibration all have direct precedents. The two pieces of prior work that most constrain this paper are named explicitly and are discussed before the positioning paragraph: the COSTGRAD token selector for visual world-model planning [22], and the ARC-Bench fixed-candidate audit of frozen JEPA-WM action ranking [23].

Token selection and sparsity for visual world-model planning. Training-free goal-cost-gradient token selection for sparse visual world-model planning has been demonstrated on the same released JEPA-WMs model family, including the Wall environment [22]. That work already names the two effects we measure — token removal changes which positions enter the cost and what the predictor produces — and reports a released-checkpoint Wall control in which its selector preserves task success at 128 tokens relative to the full 256 tokens (87.0% vs 87.0%; random 128 tokens: 75.7%) [22] (arXiv:2610.10274). These percentages are *not comparable* to our task regrets: COSTGRAD uses a different selector, plan-generation method, CEM budget and success protocol, and we do not insert its numbers as ours. This is important for the reading of our diagnostic: a pair-inversion count is not by itself evidence of task-level harm, and we do not claim otherwise. Trained sparse world-model planning approaches include Scope-WM, which distills an action-conditioned selector, summarizes foreground content, and reuses action sequences [16], and sparsely trained imagination models with a runtime-adjustable token count [4]. Training-free future-frame token pruning has also been applied to world action models [15], and dynamic density pruning is a standard video-LLM technique [19]. Our deliberately simple gradient-norm selector is a transparent reimplementation for diagnostic purposes, not a reproduction of any of these selectors, and we make no speed or selector-quality claim.

Post-compression functional and output calibration. A separate line of work reconstructs dense-model behaviour after compression. SparseGPT fits retained sparse-layer weights to dense activations on calibration inputs and applies inverse-Hessian/OBS compensation [7]; GPTQ performs one-shot second-order quantization on calibration data [8]; SparseDecoding builds the calibration matrices from the dense model's own deployment distribution rather than pre-collected text [21]. These pre-empt broad claims such as "first post-pruning functional calibration". Our target is different — nonnegative weights on a fixed sparse-token *planning cost* readout, fitted to full-model action-cost contrasts from nine paid action queries, with the frozen weights untouched — but the general idea of calibrating a compressed function's output against its dense reference is known, and we claim no originality for it.

Positive quadrature, cubature, and coreset weighting. Fitting nonnegative weights on a simplex so that a weighted subset approximates a full integral is classical quadrature and coreset construction. Empirical cubature with manifold-adaptive weights performs greedy entity pruning with convex quadratic weight redistribution and positivity [11]; continuous empirical cubature treats the cubature points as design variables [10]; Frank–Wolfe Bayesian quadrature and coresets provide probabilistic/approximation guarantees [3, 12]. Our capped-simplex regularized minimax fit is an application of these tools to a per-token terminal loss with eight action contrasts, not a new optimization principle. A closely related action-contrast reweighting idea appears in world-model *training* [5]: action-conditioned contrast magnitudes become positive token weights that are detached and used in the training loss. That pre-empts "action contrast $\rightarrow$ positive token weights" as a new idea; our use is deployment-time readout calibration against full-model planning costs rather than training-signal reweighting.

Surrogate readouts and budgeted calibration. Using a cheap surrogate to screen candidates before a costly evaluation is standard in model-predictive control, multi-fidelity optimization, and black-box optimization [13, 18]. The value equivalence principle makes the planning-functional target explicit and shows that models can be compared by the decisions they induce rather than by transition accuracy alone [6, 9]; this is the conceptual justification for calibrating a readout to a planning objective at all. Our scalar anchored-kernel comparators instantiate the simplest such surrogate: a kernel ridge regression on the 60-dimensional action tape fitted to the same nine paid labels, with an anchored kernel that makes the physical no-op contrast exactly zero. A calibration candidate must beat such surrogates at the same budget to be interesting, and in the frozen pilot it did not.

Table 1: Frozen model components and strict loading checks. Counts are recorded in `panel-repaired/result.json` (SHA-256 `2ff116b0...418c95c0`). The sparse adapter reuses these exact weights; only the token count and attention indexing change.

Component	Parameters	State tensors	Loaded strictly	Role
DINOv2 ViT-S/14 encoder	22,056,576	175	yes	visual tokens (16×16, dim 384)
AdaLN ViT predictor	17,626,480	92	yes	token dynamics, 6 blocks
Proprioceptive encoder	48	2	yes	proprio features (16-dim per token)

Model and benchmark provenance. The JEPA-WMs paper defines the model family and releases the official Wall/PointMaze/PushT checkpoints and environments used by the frozen panel [2, 20]. The visual encoder is DINOv2 [1, 17], and DINO-WM is the direct baseline family for goal-feature planning with frozen visual features [24]. We use the official code and weights unmodified at the recorded commits, and we evaluate only the Wall environment.

Closest diagnostic prior art and what is not claimed. ARC-Bench audits whether frozen JEPA-WM latent objectives rank candidate actions correctly, using a no-leak fixed-candidate protocol on official released checkpoints, reports severe structural ranking failures, and shows that closed-loop replanning can mask the defect with provenance, undertraining, backbone-scale, and metric-circularity controls [23]. AnisoWM further shows that accurate prediction and non-collapsed representations do not guarantee a task-aligned planning cost [14]. We therefore explicitly do not claim discovery of "broken action ranking in frozen JEPA world models" or of closed-loop masking as new. The object we study is narrower: how *token sparsity* changes a planning readout, decomposed into cost projection and sparse prediction, and whether a fixed nine-query calibration recipe repairs it under a predeclared gate. The pilot result is negative. The defensible statement is a measurement statement: we provide a predeclared, hash-bound protocol that separates two token-sparsity effects and reports an honest null result for one natural calibration recipe, together with the artifact set needed to reproduce it. Whether even the measurement framing is novel enough for a venue is a decision for the literature review, not something this draft asserts.

Positioning. We formulated and tested that specific object: calibrating the scoring weights of an already-deployed sparse-token predictor against full-model action-cost contrasts, with all comparators sharing one paid-label budget and evaluation on sealed unqueried actions. It did not clear its predeclared gate and is dominated by known baselines; no novelty is claimed for the object. The relevant prior art is COSTGRAD (token selection for planning) [22], ARC-Bench (rankability audit and closed-loop masking) [23], SparseGPT/GPTQ/SparseDecoding (calibration-data reconstruction) [7, 8, 21], MAW-ECM/CECM (positive weighted subsets) [10, 11], CAER (action-contrast reweighting) [5], and multi-fidelity estimation (budgeted calibration) [13, 18]. The paper's contribution is the readout-fidelity measurement and the falsification, not a method.

3 Problem and Protocol

This section fixes notation for the frozen model (Section 3.1), defines the position-preserving sparse adapter (Section 3.2), the two selectors (Section 3.3), the all-token and tie controls (Section 3.4), the calibration QP and its baselines (Section 3.5), and the decision metrics, predeclared gate, and query accounting (Sections 3.6–3.7).

3.1 Frozen model and terminal cost functional

We use the official JEPA-WMs "Wall" world model and its official DINOv2 ViT-S/14 visual encoder as distributed by the publishers [1, 2, 17, 20]. Table 1 records the frozen component sizes and the strict loading check. All weights are evaluated in inference mode, no parameter is updated, and the full-token path is checked against the adapter path with a control described in Section 3.4.

Inputs and actions. Each scene provides an initial image and a goal image rendered by the publishers' Wall renderer, plus a 2-dimensional initial and goal position. The visual encoder receives the official 65×65 image resized to 224×224 and ImageNet-normalized, producing $N = 256$ patch tokens of dimension $d_v = 384$. The proprioceptive encoder maps each 2-dimensional position to a 16-dimensional feature per token. A *physical action tape* for horizon $H = 6$ is an array $a \in \mathbb{R}^{H \times 5 \times 2}$: five raw 2-dimensional controls per model step, normalized per coordinate with the

official Wall statistics and flattened to a 10-dimensional predictor input; H steps form 60 raw physical coordinates. A *physical no-op* a_0 has all raw coordinates zero (it is not the normalized zero vector).

Cost functional. For action tape a, let $z^v_{H,i}(a) \in \mathbb{R}^{384}$ and $z^p_{H,i}(a) \in \mathbb{R}^{16}$ be the predicted terminal visual token and proprioceptive token feature at original token index i after H model steps, and let $z^v_{g,i}, z^p_{g,i}$ be the corresponding goal encodings. The per-token terminal loss and the full-token cost are

$$\varphi_{\mathrm{F}\,i}(a) = \frac{1}{384}\left\|z^v_{H,i}(a) - z^v_{g,i}\right\|_2^2 + 0.1 \cdot \frac{1}{16}\left\|z^p_{H,i}(a) - z^p_{g,i}\right\|_2^2, \tag{1}$$

$$C_{\mathrm{F}}(a) = \frac{1}{256}\sum_{i=1}^{256} \varphi_{\mathrm{F}\,i}(a). \tag{2}$$

This is the terminal objective of the publisher's representation-target MPC objective with $\alpha = 0.1$ and no sum over horizons; Eq. (2) defines the full-token cost used as the computational reference throughout. Per-token losses are computed in float64 from the saved float32 model outputs. We report horizons $H = 1$ and $H = 6$ from the same rollout; the full-model cost is a higher-compute *computational reference*, not physical truth and not a physical regret.

3.2 Position-preserving sparse adapter

A token subset $S \subset \{1, \ldots, 256\}$ with $|S| = K = 128$ is fixed before any rollout and shared over all action candidates and all six steps. The sparse adapter runs the same frozen predictor restricted to the tokens in S:

- the initial visual tokens and the propagated proprioceptive features are restricted to S;
- rotary position embeddings use the *original* token IDs in S, and the temporal attention mask is sliced accordingly;
- AdaLN modulation is broadcast over K tokens instead of 256, and reshapes use the original 6-dimensional Linear input layout (see Section 3.4).

No weight is added, removed, or updated; the adapter is a deployment-time restriction of the same checkpoint. Its terminal per-token loss $\varphi_{\mathrm{P},S_i}(a)$ for $i \in S$ and the sparse readout are

$$C_{\mathrm{P}|S}(a) = \frac{1}{K}\sum_{i\in S} \varphi_{\mathrm{P},S_i}(a), \qquad C_{\mathrm{F}|S}(a) = \frac{1}{K}\sum_{i\in S} \varphi_{\mathrm{F}\,i}(a). \tag{3}$$

$C_{\mathrm{F}|S}$ is the projection of the *full-token* cost onto the same subset (it is computable only with full-token predictions); $C_{\mathrm{P}|S}$ is what a deployed sparse model can actually compute. The distinction between Eqs. (2) and (3) is the core of the diagnostic: $C_{\mathrm{F}|S}$ isolates the scoring-subset projection of the per-token losses $\varphi_{\mathrm{F}\,i}$ of Eq. (1), and $C_{\mathrm{P}|S}$ adds the change in the prediction path.

3.3 Token selectors

Two selectors are frozen and reported separately; neither may be chosen after seeing its results.

- **Random** (`random128`): 128 original token IDs drawn with a fixed per-scene seed, fixed over all candidates and steps. In the pilot, seeds are 3181 + context.
- **Gradient** (`gradient128`): one full-token H=6 physical-no-op rollout is differentiated with respect to the initial visual tokens, and the per-token L2 norm of the terminal combined-cost gradient selects the 128 largest tokens with a stable sort, then the IDs are sorted ascending before the fixed sparse rollout. In the panel, the random seed is 971 + scene; the gradient selector is a transparent reimplementation of a standard idea and is *not* the official implementation of any published token-selection method.

Selecting the gradient subset requires one full no-op forward and one backward pass; it is charged to the selector and reused by all methods (Section 3.7).

3.4 Controls, tie handling, and inversion counting

All-token control. For every context, every evaluated action batch (including the pilots), and all six steps, the adapter is also run with $S = \{1, \ldots, 256\}$ and compared against the original dense predictor path, with output tolerances atol $= 2 \times 10^{-5}$ and rtol $= 10^{-5}$. Any mismatch would stop interpretation. In the reported panel the maximum absolute difference was 0 for both visual and proprioceptive outputs over all four scenes, 32 candidates, and six steps (16 checks), and in the pilot the implementation runs the same control per context and batch (Section 4.4).

Implementation repair (recorded before interpreting pruning). The first full panel run failed the all-token control: a non-contiguous sliding history made the pointwise predictor embedding choose a different 4-dimensional Linear grouping and accumulated float32 differences up to $4.57763671875 \times 10^{-5}$ in a same-input diagnostic. The repair restores the official six-dimensional input layout (row width 16 for K divisible by 16) before flattening; all model parameters, token IDs, masks, scenes, actions, selectors, dtype, thresholds, and the cost definition are unchanged (recorded in `implementation-amendment.json`, SHA-256 `ccbffd1e...08525334`). Sparse rankings from the failed run were not inspected, and the failed artifacts are preserved. This is an implementation repair, not a protocol change, and it is disclosed here because it explains why the verified panel is the "repaired" run.

Tie tolerance. Costs are compared with the fixed tolerance $\tau = 7.62939453125 \times 10^{-6} = 2^{-17}$. For a pair of actions (u, v), the pair is *comparable* if $|C_{\mathrm{F}}(u) - C_{\mathrm{F}}(v)| > \tau$ and $|\hat{C}(u) - \hat{C}(v)| > \tau$; it is an *inversion* if the two cost differences have opposite signs. The inversion fraction is the number of inversions divided by comparable pairs. The panel has 32 action candidates per scene, i.e. $\binom{32}{2} = 496$ pairs per scene and 1984 pooled pairs. Top-choice ties are reported as tie sets with the first-index selection rule; a selected action's "full-reference gap" is its full cost minus the minimum full cost.

3.5 Calibration readout and the predeclared QP

The pilot uses nine *paid* full-model action queries per context: $\mathcal{P} = \{a_0, a_1, \ldots, a_8\}$ with a_0 the physical no-op, and eight constant-heading action tapes. Full-model outputs at the paid actions give the contrast targets of Eq. (4),

$$b_j = C_{\mathrm{F}}(a_j) - C_{\mathrm{F}}(a_0), \qquad j = 1, \ldots, 8, \tag{4}$$

and the sparse predicted per-token contrast matrix of Eq. (5),

$$A^P_{j,i} = \varphi_{\mathrm{P},S_i}(a_j) - \varphi_{\mathrm{P},S_i}(a_0), \qquad i \in S,\ j = 1, \ldots, 8. \tag{5}$$

Let $u_i = 1/K$, $K = 128$, and let $s = \max(\max_j |b_j|,\ \tau)$ be the context scale. The candidate weights solve the regularized minimax capped-simplex problem

$$\min_{w,t}\ t + \eta K \|w - u\|_2^2 \quad \text{s.t.} \quad \left|\frac{(A^P w - b)_j}{s}\right| \le t,\ \sum_i w_i = 1,\ 0 \le w_i \le \frac{4}{K}, \tag{6}$$

with fixed $\eta = 0.01$, cap $4/K = 1/32$ and scale floor τ. The strictly convex penalty makes the solution of Eq. (6) unique and prevents an arbitrary vertex choice; the cap guarantees an inverse-concentration effective sample size $1/\sum_i w_i^2 \ge K/4$. The fitted readout is anchored to the paid full no-op,

$$\hat{C}(a) = C_{\mathrm{F}}(a_0) + \sum_{i \in S} w_i \big[\varphi_{\mathrm{P},S_i}(a) - \varphi_{\mathrm{P},S_i}(a_0)\big], \tag{7}$$

so that $\hat{C}(a_0) = C_{\mathrm{F}}(a_0)$ and absolute sparse offsets cannot affect the pilot-versus-evaluation decision. The QP is solved once per context and selector with SLSQP (`maxiter`= 500, `ftol`= 10^{-10}, analytic Jacobian), and an independent linearized LP certificate checks primal feasibility ($\le 10^{-7}$), a Frank–Wolfe gap in $[-10^{-7}, 10^{-5}]$, and stationarity and duality gap of the LP certificate ($\le 10^{-7}$); the certificate is a first-order gap at the returned point, not a generalization guarantee. There is no hyperparameter retry and no solver swap after seeing evaluation labels (Appendix A).

Comparators. All comparators use the same nine paid full labels, the same anchor, the same s, cap, penalty, and solver settings; only the fitted matrix or the scoring function changes.

- *Uniform sparse + anchor* (`uniform_paid`): $w = u$ with the same anchored prediction (7).
- *Full-integrand absolute cubature* (`full_absolute`): fit $\varphi_{\mathrm{F},i}(a_j)$ (all nine paid actions, including the no-op row) to the absolute full costs $C_{\mathrm{F}}(a_j)$ with the same QP, then apply the weights to the sparse outputs. This is classical positive quadrature and is a mandatory baseline.
- *Full-integrand contrast cubature* (`full_contrast`): fit the full-token contrast matrix $A^F_{j,i} = \varphi_{\mathrm{F}\,i}(a_j) - \varphi_{\mathrm{F}\,i}(a_0)$ to b with the same QP, then apply the weights to the sparse outputs.
- *Sparse-integrand contrast* (`sparse_contrast`): the candidate, fitting A^P to b.
- *Scalar anchored-kernel (full)* (`scalar_full`): kernel ridge regression on the 60-dimensional raw action tape with $k(a, a') = \exp[-\overline{(a - a')^2}/(2 \cdot 0.5^2)]$ and anchored kernel $k_0(a, a') = k(a, a') - k(a, a_0) - k(a_0, a') + 1$, fitted to b/s on the eight non-anchor paid actions with ridge 10^{-3} and no intercept.

- *Scalar anchored-kernel (residual)* (`scalar_residual`): the same surrogate fitted to the residual between b and the uniform sparse contrast, deployed as anchor plus uniform sparse contrast plus residual prediction.
- *Best paid pilot* (`best_paid_pilot`): choose the paid pilot with the smallest full cost; it uses nine full queries and no sparse evaluation forward pass.
- *Uniform sparse without anchor* (`uniform_unqueried`): raw sparse mean over K tokens, no labels and no anchor; for the gradient selector it still requires the selector's full no-op forward and backward.

3.6 Decision metrics and the predeclared gate

Let $E_{64} = \{a_9, \ldots, a_{72}\}$ be the 64 unqueried evaluation actions and $U_{73} = \mathcal{P} \cup E_{64}$ the union. All full-model costs at E_{64} are *sealed audit labels*: they are not used for fitting, weighting, mask selection, normalization, or fallback. For a method with prediction $\hat{C}$:

$$G_E = C_{\mathrm{F}}(\hat{u}_E) - \min_{a \in E_{64}} C_{\mathrm{F}}(a), \qquad \hat{u}_E = \arg\min_{a \in E_{64}} \hat{C}(a), \tag{8}$$

$$G_O = C_{\mathrm{F}}(\hat{u}_O) - \min_{a \in U_{73}} C_{\mathrm{F}}(a), \qquad \hat{u}_O = \arg\min_{a \in U_{73}} \tilde{C}(a), \tag{9}$$

where $\tilde{C}(a) = C_{\mathrm{F}}(a)$ on the nine paid pilots and $\tilde{C}(a) = \hat{C}(a)$ on E_{64}; ties are broken by the first action index within τ. The pilot-benefit comparator is $\Delta_{\mathrm{pilot}} = C_{\mathrm{F}}(\hat{u}_O) - \min_{a \in \mathcal{P}} C_{\mathrm{F}}(a)$: a value below $-\tau$ is a win over the best paid pilot, above $+\tau$ a loss, otherwise a tie. Per-context normalized gaps divide by $\max(\mathrm{ptp}_{E_{64}} C_{\mathrm{F}}, \tau)$; contexts with a flat range ($\mathrm{ptp} \le \tau$) are flagged, retained, and counted. The primary metric is the mean G_E over the 16 contexts for the gradient selector; the union gap and the best-paid-pilot comparison are secondary but gating.

Frozen gate. Before execution the project fixed a seven-condition engineering gate (`method_spec.json`, section `go_development_only`; SHA-256 `c0678a16...9c59c2323f`):

1. all solver certificates and all-token controls pass for all 32 context/selector records;
2. the best comparator's mean G_E exceeds the tie floor τ;
3. the candidate's mean G_E is at most $0.80\times$ the best comparator's mean G_E (20% relative reduction);
4. the candidate improves the mean normalized G_E by at least 0.02 relative to the best comparator;
5. the candidate's mean union G_O is no worse than the best comparator's plus τ;
6. the candidate is strictly better than the best paid pilot in at least four contexts;
7. the candidate is worse than the best paid pilot in at most two contexts.

One global comparator B^* is chosen by the raw mean G_E on the gradient selector over all 16 contexts, with a fixed comparator order (`uniform_paid`, `full_absolute`, `full_contrast`, `scalar_full`, `scalar_residual`) breaking exact ties; the same B^* is used in all gate terms, with no per-context oracle. A missing context, a solver failure, or an all-token failure is NO_GO; no context may be dropped. Passing yields `GO_DEVELOPMENT_ONLY`; failing yields `NO_GO_FIXED_RECIPE`. The gate is an engineering continuation threshold fixed before outputs; it is not statistical significance, universal correctness, physical-control benefit, or a journal standard. Failure stops this fixed recipe, not all calibration.

3.7 Query and compute accounting

Equal paid-label counts are not equal end-to-end compute, and this study does not claim a speedup. The logical query counts are: paid methods use nine full action queries per context; the gradient selector additionally uses one full no-op forward and one backward pass (its forward is reused as the paid no-op); the random selector uses no full query; the 64 evaluation full rollouts per context are audit-only and available to no method. *scalar_full* requires no sparse deployment forward pass, while the other calibrated methods score 64 sparse candidates per selector. Because the recorded execution shares the sparse rollouts across methods and runs both selectors for the same contexts, the total execution ledger is not a per-method deployment cost; Appendix B (Table 19) lists the protocol accounting. The pilot requires 16 contexts $\times$ 73 full-model H=6 rollouts (9 paid pilots and 64 sealed audit actions per context) and 16 contexts $\times$ 2 selectors $\times$ 73 sparse-token H=6 rollouts, before the K=256 control rollouts (not counted here).

4 Predeclared Development Experiment

The experiment has two frozen parts. Section 4.1 describes the four-scene diagnostic; Section 4.2 the 16-context calibration pilot; and Sections 4.3–4.5 the freezing order, the resource amendment log, and the technical controls. All quantities in this section are protocol facts fixed before outputs, not results.

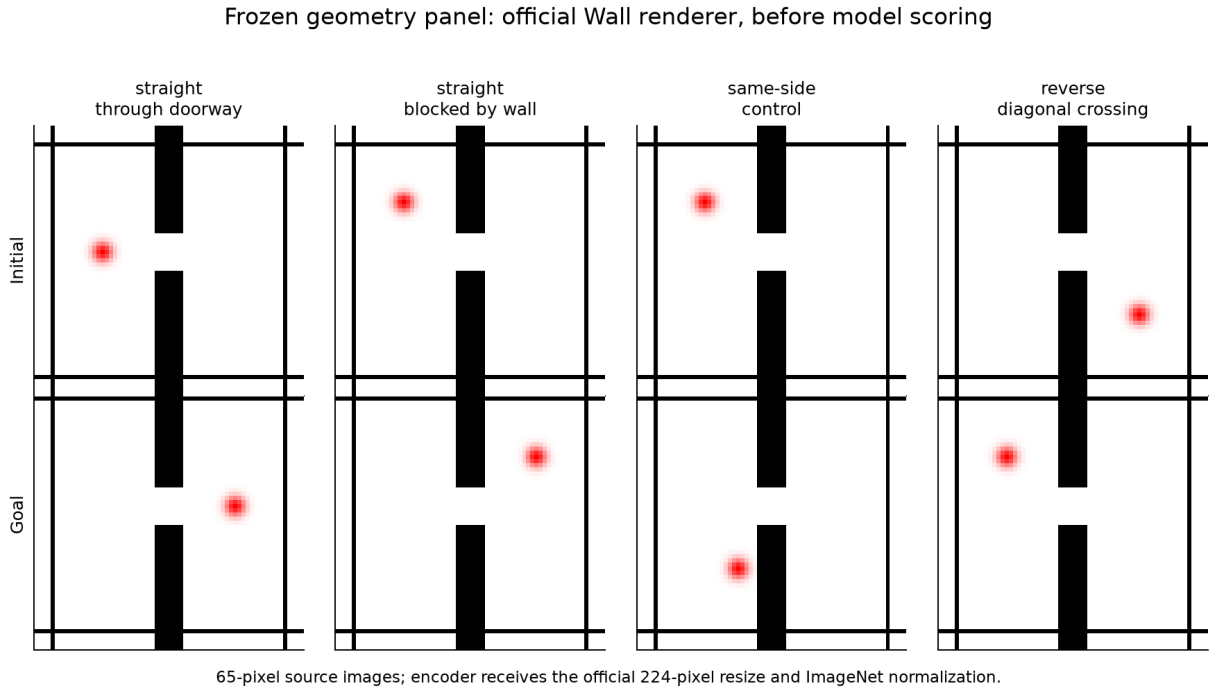

Figure 1: The four synthetic Wall scenes used in the diagnostic (initial and goal images, top and bottom rows). Images are produced by the publisher's official Wall renderer for the fixed scene configurations in `panel-protocol.json`; no real-world image, simulator rollout, or human data is involved. Reproduced unmodified from `panel-repaired/figures/fixed_scenes.png` (SHA-256 `25747a9c...f6d8a537409`).

4.1 Diagnostic panel design

Four synthetic Wall geometries are fixed (straight through doorway, straight blocked by wall, same-side control, reverse diagonal crossing), each with one goal and 32 fixed action tapes (panel protocol SHA-256 `74806dfd...1b4f461edf00`). The 32 tapes are: one physical no-op; 16 constant-heading tapes at amplitude 0.75; 8 constant-heading tapes at amplitude 0.35; 7 seeded three-waypoint tapes in $[-0.75, 0.75]$, each waypoint repeated for two model steps and each model step using five raw controls. Six steps are run once per action tape and the terminal costs are reported at $H = 1$ and $H = 6$ from the same rollout. For each scene and selector we store full-token outputs, full-token projected costs, and sparse adapter outputs, and count the three comparisons of Section 5.1, reported in Table 3 and Figure 2. The initial and goal images are produced by the publisher's official Wall renderer; the four scenes are shown in Figure 1.

4.2 Calibration pilot design

Sixteen new geometry contexts and goals are fixed before any model output, four per geometric stratum (doorway crossing, blocked crossing, same-side, reverse crossing), generated with geometry seed 1141 and per-coordinate jitter in $[-0.75, 0.75]$; wall location and door location are the official Wall values. The 16 initial/goal pairs are recorded in `input_design.json` (SHA-256 `015e9caf...297b97ddad`) and in `inputs.npz` (SHA-256 `cc8c1b36...2ca8c33`). One 73-action bank is shared across all contexts: nine paid pilots $a_0, \ldots, a_8$ (physical no-op plus eight constant headings at amplitude 0.5) and 64 unqueried evaluation actions (16 constant headings at amplitude 0.7 with a fixed phase offset, 8 constant headings at amplitude 0.4 with a different phase offset, and 40 seeded three-waypoint tapes, each repeated for two model steps with five raw controls per step). The bank has 73 unique actions and contains no physical no-op outside index 0. Table 2 lists the frozen protocol constants.

4.3 Freezing, chronology, and saved evidence

The runner enforces the following per-context order and the readout verifies it from timestamps: (i) paid pilot forwards complete; (ii) weights and kernel coefficients are fitted and saved; (iii) sparse evaluation starts; (iv) all predictions are

Table 2: Frozen calibration-pilot protocol constants (`method_spec.json` SHA-256 `c0678a16...9c59c2323f`; `input_design.json` SHA-256 `015e9caf...297b97ddad`). These are protocol facts, not results.

Item	Frozen value
Contexts	16 new geometries, 4 strata × 4; geometry seed 1141
Action bank	73 actions shared across contexts (9 paid, 64 unqueried)
Horizon	$H = 6$ model steps; six-step rollouts, terminal cost
Tokens	full model 256; sparse adapter $K = 128$; same subset across steps
Selectors	random128 (seed 3181 + context); gradient128 (full no-op H=6 backward)
Paid queries	9 full-model actions per context (one physical no-op, eight constant headings)
Audit labels	64 unqueried full-model actions per context, sealed until predictions are frozen
QP	regularized minimax, $\eta = 0.01$, cap $4/K = 1/32$, scale floor τ, SLSQP fixed
QP certificate	primal violation $\leq 10^{-7}$; FW gap $\in [-10^{-7}, 10^{-5}]$; stationarity/duality $\leq 10^{-7}$
Scalar kernel	anchored RBF on 60-dim raw action tape; length 0.5; ridge 10^{-3}
Tie tolerance	$\tau = 2^{-17} = 7.62939453125 \times 10^{-6}$
Normalization	per-context G_E divided by $\max(\mathrm{ptp}_E C_{\mathrm{F}}, \tau)$; flat contexts retained
Gate	seven predeclared conditions (Section 3.6); one global comparator B^*

frozen to a saved artifact with a SHA-256; (v) only then does the full-model audit of the 64 unqueried actions start; (vi) the context completes. The readout refuses to run unless the run is complete, all 16 contexts are present in order, the frozen method-spec, core, inputs, runner, and base-array hashes match, the saved prediction arrays are identical to the arrays audited, and the pilot arrays are unchanged after fitting. The 64 audit costs are never used to choose an action or a fallback; the readout's own synthetic tests exercise this boundary (`check_readout_synthetic.json`, 9 checks, PASS).

4.4 Resource amendment and integrity log

The calibration pilot was originally run under a cooperative 600-second development boundary. That run stopped at 601.4792 seconds with 7 of 16 contexts complete (inside context 07 sparse evaluation) and remains preserved unmodified as `FAILED_FIXED_DEVELOPMENT_EXPERIMENT`. Because the frozen readout refuses any partial run and requires one complete run identity, a *resource-only* amendment raised the boundary to 7200 seconds and re-launched from scratch in a fresh run directory (`budget-amendment.json`, SHA-256 `83fd078a...9e2ae7e4e`; written before execution). The byte-identical inputs, adapter core, calibration module, and method spec (`inputs.npz`, `probe_core.py`, `calibration.py`, `method_spec.json`) and every context, action, selector, solver, tolerance, normalization, and gate value were unchanged; only the cooperative clock and the output directory changed. The first launch of the amended directory aborted before any model execution because the runner looked up a bookkeeping key in the wrong JSON layout; the failed pre-execution record is preserved, the lookup was corrected, and a new pre-execution amendment record was hashed before the successful relaunch (`bootstrap-incident.md`). No numerical code changed. The completion was required regardless of outcome: the 16-context run was executed and read out under the frozen gate, and the partial run's seven-context descriptive metrics (computed after its predictions had been frozen and before its sealed audit labels) did not influence any input, method, solver, or gate choice. The amended run completed all 16 contexts with recorded wall time 887.9167737999996 s and peak allocated GPU memory 1,345,822,720 bytes (`run/result.json`, SHA-256 `311f2349450ba78d5fdc74cf8bfa843a692403ff602cbb9e0930d3836c606acd`); the unmodified readout was then generated as `run/readout.json` (SHA-256 `c0cc4456abf60cb5cded4a7f8f8155ba320e64505253a256133ca0cd9fa49480`), and both the stopped original run and the complete amended run remain preserved under their own identities. The immutable evidence

Table 3: Pairwise action-order inversions in the four-scene diagnostic. Each row pools 4 scenes $\times\binom{32}{2} = 1984$ comparable action pairs; percentages are descriptive fractions, not estimates of a population quantity. "Score projection" compares the full-token ranking with the ranking of the same full-token costs projected onto the 128-token subset; "sparse prediction" compares the projected full-token ranking with the actual sparse-predictor costs on the same subset; "combined" compares the full-token ranking with the actual sparse-predictor ranking. The effects are non-additive and must not be added. Source: `panel-repaired/analysis.json` (SHA-256 `a915a2e4...f7f7f7f347`); counts also appear in `panel-repaired/result.json` and `decision.json`.

Selector	Comparison	Horizon	Inversions	Fraction
random128	score projection	H=1	684	34.48%
random128	sparse prediction	H=1	296	14.92%
random128	combined	H=1	756	38.10%
random128	score projection	H=6	440	22.18%
random128	sparse prediction	H=6	374	18.85%
random128	combined	H=6	468	23.59%
gradient128	score projection	H=1	157	7.91%
gradient128	sparse prediction	H=1	61	3.07%
gradient128	combined	H=1	152	7.66%
gradient128	score projection	H=6	793	39.97%
gradient128	sparse prediction	H=6	174	8.77%
gradient128	combined	H=6	785	39.57%

snapshot at `outputs/research-lab/world_models/contrast-calibration-pilot-amended-initial/` contains `snapshot_manifest.json` (SHA-256 `e7b03a0b80dd45da50329539d2594dee3f0130533afcd4c69f7d92c5ceb1349e`; `payload_count`=125, `payload_bytes`=93,250,531; the manifest excludes itself), and was passed through the independent verifier's audit.

4.5 Technical controls in the pilot

Every context run performs the K=256 all-token control for the no-op batch, the eight paid pilots, and each eight-action audit batch, and the readout requires all 160 checks to pass with finite reported differences (atol $= 2 \times 10^{-5}$, rtol $= 10^{-5}$); the complete run reports all 160 checks passed. Solver certificates are checked for all three weight methods (`full_absolute`, `full_contrast`, `sparse_contrast`) and both selectors per context, and the frozen readout reports them successful for all 32 context/selector records. A flat evaluation range is flagged and retained (none of the 16 contexts is flat); a missing context, solver failure, or all-token failure blocks the gate rather than being dropped. An analytic two-case check of the QP solver (exact capped-simplex solutions at zero and nonzero residual) passed before the pilot (`solver_analytic_check.certified.json`, SHA-256 `6d6d2033...c7cae9e2`).

5 Results

5.1 Four-scene diagnostic: projection and sparse prediction

Table 3 reports all twelve selector × horizon × comparison combinations (Figure 2 shows the same counts per scene). We highlight the gradient/128-subset H=6 setting, which is the deployment setting used by the pilot, but report every cell so that the reader can see that the qualitative pattern is not obtained by selecting a favourable scene or horizon: the projection effect is large, the sparse-prediction effect is smaller under the gradient selector, and the combined effect is not their sum.

Selected-action consequences. Inversion counts do not by themselves imply that the chosen action changes. Table 4 therefore reports, for the gradient/H=6 setting, the full-token top-tie set and the first-index selected action for the projected, sparse, and combined rankings, together with the full-cost gap of the combined selection. The first-index selected action changes in exactly one of four scenes (scene 1, gap 0.14230); in the other three scenes the

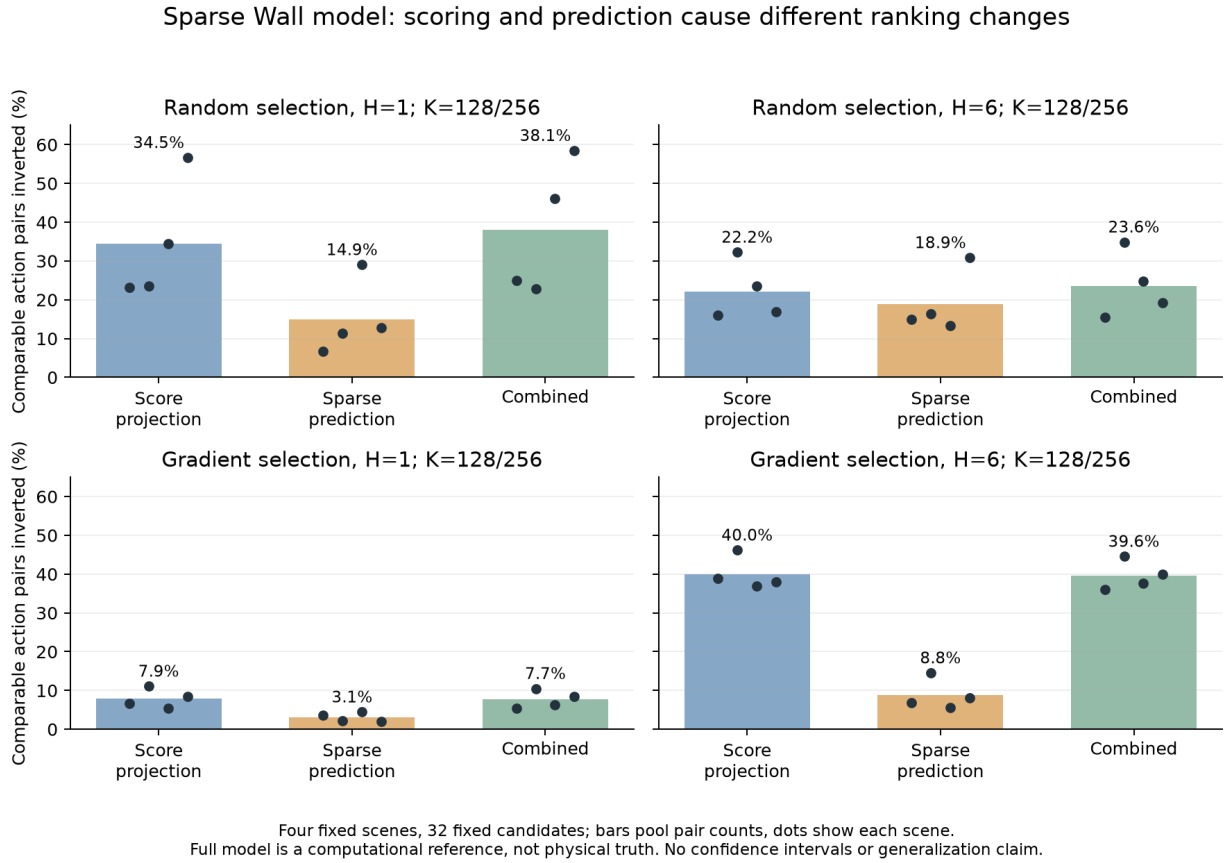

Figure 2: Pairwise inversion fractions for all four scenes, both selectors, and both horizons. Bars pool the 1984 comparable pairs; dots show the four individual scenes, so the spread is visible rather than hidden by the pooled bar. The gradient-selected 128-token subset tracks the full-token action response far better than the random subset (Section 5.1), yet at H=6 its score projection still reorders 39.97% of comparable pairs. Reproduced unmodified from `panel-repaired/figures/ranking_decomposition.png` (SHA-256 `ce71ac7f...2d16eecf1e8`); the parent readout states that the full model is a computational reference, not physical truth.

combined selection has zero full-cost gap within the tie tolerance, and in scene 0 the projection and sparse-prediction changes cancel (projection alone would select index 1, sparse prediction alone index 16, and the combined ranking selects the full-token index 16 again). The three counts of Table 3 cannot be added, subtracted, or normalized into "shares of error".

Action-response descriptives. To check that the gradient subset is not simply a bad subset, we compare the full and sparse predicted visual responses to the physical no-op action at the same original token IDs. At H=6 the gradient selector's median relative L2 response error is 0.136–0.164 across scenes with cosine 0.987–0.991, whereas the random subset's median relative L2 error is 0.359–0.938 with cosine 0.640–0.934. The sparse-adapter change is therefore small under the gradient selector, consistent with the inversion counts; this describes response fidelity inside one frozen model and is not a dynamical or physical guarantee.

Independent saved-array check. The saved H1/H6 arrays were independently re-aggregated on CPU with NumPy (no model, weight, encoder, gradient, or renderer recomputation): 1280 saved cost values across the five scoring views of the four scenes were recomputed from the saved float32 features, with maximum absolute error 2.32415×10^{-7}, below the fixed absolute check tolerance $1.9073486328125 \times 10^{-6}$ ($\tau/4$). The receipt reports 166 passed checks and a scope limited to saved H1/H6 arrays and metadata; H2–H5 outputs were not saved and

Table 4: Gradient/H=6 selected actions and full-cost gaps per scene (scene indices 0–3 correspond to the four panels of Figure 1). "Top–tie set" is the set of first-place actions within τ; "gap" is the combined selection's full cost minus the scene's minimum full cost. Source: `panel-repaired/analysis.json` (SHA-256 `a915a2e4...f7f7f7f347`).

Scene	Full top set	Projected top set	Sparse top set	Combined top set	Combined gap
0	{16}	{1}	{16}	{16}	0.0
1	{1}	{1}	{2}	{2}	0.14230
2	{19}	{19}	{19}	{19}	0.0
3	{11}	{11}	{11}	{11}	0.0

Table 5: Per-method mean selected-action gaps for the gradient K=128 selector (primary; 16 paired contexts). G_E is the 64-unqueried-action gap to the full-model optimum; the normalized gap divides by the per-context full-cost range (floored at τ); G_O is the union-73 gap including the nine paid pilots; W/L/T counts contexts strictly better/worse/tied against the best paid pilot. † marks the global best paid comparator B^* selected by raw mean G_E; the candidate row is bold. * best-paid-pilot has no E_{64} or normalized E_{64} selection, and its own W/L/T is the baseline. Generated from the frozen readout; see `claims-ledger.md` for the readout identity.

Method	mean E_{64} gap	mean norm. E_{64} gap	mean U_{73} gap	W/L/T*
Uniform sparse (raw, no anchor)	0.064461	0.085731	0.060012	7/5/4
Uniform sparse + full no-op anchor	0.064461	0.085731	0.120131	7/9/0
Full-integrand absolute cubature	0.057841	0.112618	0.034196	6/4/6
Full-integrand contrast cubature	0.057841	0.112618	0.034196	6/4/6
Sparse-integrand contrast (candidate)	0.076410	0.131333	0.071960	6/6/4
Scalar anchored-kernel (full-cost)	0.269665	0.336037	0.318985	2/12/2
Scalar anchored-kernel (full-sparse residual)†	0.029254	0.076734	0.049337	6/6/4
Best paid pilot (9 full queries)	n/a	n/a	0.068265	—

cannot be independently reaggregated, the K256 control tensors were not saved (their zero differences are source-consistent reported metadata), and the selector gradient itself was not independently recomputed. The independent re-aggregation checks the arithmetic of the saved evidence, not the network execution.

5.2 Calibration pilot results

This subsection reports the predeclared 16-context development pilot. Its result cells are generated from the frozen readout by `tools/extract_pilot_numbers.py` (readout SHA-256 `c0cc4456abf60cb5cded4a7f8f8155ba320e64505253a256133ca0cd9fa49480`); no number here is typed by hand. Table 5 summarizes the per-method mean gaps of Eqs. (8)–(9), Table 6 gives all 16 paired contexts for the primary selector, Table 8 reports the secondary pair-inversion and contrast-RMSE descriptives, and Table 7 evaluates the predeclared gate. The complete 16-context run and the unmodified readout are the only source of these numbers; the independent-verification status is recorded in the front matter and in `claims-ledger.md`. The secondary random-selector summary (Table 21) and the logical query accounting are in Appendices D and B.

Reading of the frozen gate (negative result). The predeclared gate returned `NO_GO_FIXED_RECIPE`: 3 of 7 conditions passed, and the failed conditions were 20% relative gap reduction; 0.02 normalized improvement; union gap no worse; at most 2 pilot losses (Table 7). The candidate's mean $G_E = 0.076410$ versus 0.029254 for Scalar anchored-kernel (full-sparse residual) means its full-reference gap is 161.2% *larger* than the best paid comparator's, where the gate required a 20% reduction; the normalized-gap change is -0.054599 (required +0.02); the union-73 gap is 0.071960 versus 0.049337; and the candidate is worse than the best paid pilot on 6 of 16 contexts (at most two allowed), better on 6 (at least four required), and tied on 4.

Table 6: Per-context gradient K=128 results. G_E/G_O are the candidate and comparator full-reference gaps; ΔG_E is candidate minus comparator (negative favours the candidate). Stratum labels follow the order in `input_design.json` (contexts 0–3 doorway, 4–7 blocked, 8–11 same-side, 12–15 reverse crossing). Generated from the frozen readout.

Ctx	Stratum	cand. G_E	comp. G_E	ΔG_E	cand. G_O	comp. G_O	vs. pilot
00	doorway crossing	0.025441	0.000000	0.025441	0.153430	0.127988	loss
01	doorway crossing	0.084268	0.000000	0.084268	0.000000	0.109906	tie
02	doorway crossing	0.145654	0.000000	0.145654	0.000000	0.000000	tie
03	doorway crossing	0.000000	0.000000	0.000000	0.000000	0.000000	tie
04	blocked crossing	0.196856	0.156197	0.040659	0.196856	0.156197	loss
05	blocked crossing	0.000000	0.000000	0.000000	0.000000	0.000000	win
06	blocked crossing	0.081528	0.064346	0.017182	0.105930	0.088748	loss
07	blocked crossing	0.048474	0.076012	-0.027538	0.048474	0.076012	loss
08	same-side	0.088824	0.088824	0.000000	0.088824	0.088824	win
09	same-side	0.000000	0.000000	0.000000	0.000000	0.000000	win
10	same-side	0.000000	0.000000	0.000000	0.000000	0.081492	win
11	same-side	0.000000	0.028793	-0.028793	0.000000	0.000000	tie
12	reverse crossing	0.450136	0.000000	0.450136	0.456470	0.006334	loss
13	reverse crossing	0.000000	0.000000	0.000000	0.000000	0.000000	win
14	reverse crossing	0.101375	0.053889	0.047486	0.101375	0.053889	loss
15	reverse crossing	0.000000	0.000000	0.000000	0.000000	0.000000	win

The candidate is dominated at the same label budget. The two full-integrand positive-cubature variants selected the same first-index action in every one of the 32 context/selector records and therefore report identical mean gaps (0.057841 on the primary selector); this baseline is better than the candidate's 0.076410. The scalar full–sparse residual surrogate achieved 0.029254, the best mean G_E of any method in the run on both selectors. Even the uncalibrated sparse mean achieved 0.064461, which is lower than the candidate's primary metric; the candidate's gap is 18.5% larger than the plain sparse mean and 32.1% larger than the full-integrand baseline. On the random selector the candidate achieves 0.114094 versus the comparator's 0.069862 (-0.6% relative to the uncalibrated mean; see Appendix D). The fixed sparse-token contrast QP is therefore not merely short of its 20% gate: it is worse than classical cubature on full-token per-token losses, worse than a scalar residual surrogate at the same nine paid labels, and on the primary selector worse than calibrating nothing. The negative result is not caused by a solver or control failure: the frozen readout reports all solver certificates and all 160 all-token checks passed, and neither selector has a flat evaluation range (flat-context counts 0 and 0).

Descriptive pair metrics do not track decision quality. Table 8 reports two secondary descriptives for the gradient selector. The candidate does have a lower mean pairwise inversion fraction than the best comparator (25.5% versus 28.1%) and a slightly lower eval-contrast RMSE (0.111327 versus 0.162174), yet a much worse selected-action gap. This mirrors the diagnostic panel, where pair inversion counts and chosen-action loss also diverge, and it is the reason the predeclared gate is defined on selected-action gaps rather than on pair metrics. The descriptive metrics are reported for completeness; they were never part of the gate and are not used to rescue the recipe. Figure 3 shows the per-method means graphically.

The inversion/decision divergence, both selectors. The divergence is not an artifact of one selector. Mean eval64 pair-inversion fractions are 25.46% for the candidate versus 34.12% for uniform sparse scoring and 25.29% for the full-integrand baseline on the gradient selector, and 27.02% versus 29.61% and 26.74% on the random selector, yet the candidate's selected-action gap is worse than the comparator's on both selectors. Calibration therefore reduced pair inversions relative to uniform sparse scoring while worsening the decision metric. This is not a discovery of the paper — COSTGRAD's two-effect statement and ARC-Bench's rankability audit already imply that ranking statistics need not track decisions [22, 23] — but it is the clearest evidence here that inversion counts are not a decision metric and that the 39.97% panel headline must not be read as task harm. We report the paired per-context picture in Section 9.3 and both selectors in full rather than presenting a cleaner failure than the data support. On the random selector the candidate's mean union gap (0.053987) is even better than the global comparator's (0.079498)

Table 7: Predeclared gate conditions evaluated on the gradient K=128 selector with the single global comparator B^* (`scalar_residual`). The gate is an engineering continuation threshold fixed before outputs, not a statistical test. Generated from the frozen readout.

Condition	Predeclared threshold	Observed (gradient K=128)	Status
All QP certificates and all-token controls pass	required	32/32 records pass	PASS
Best comparator mean E_{64} gap exceeds tie tolerance	$> 7.629 \times 10^{-6}$	0.029254	PASS
Candidate mean E_{64} gap $\leq$ 0.80× comparator	$\leq$ 0.80× comparator	0.029254 $\rightarrow$ 0.076410 (-161.2% reduction)	FAIL
Comparator minus candidate mean normalized E_{64} gap	≥ 0.02	-0.054599	FAIL
Candidate mean U_{73} gap $\leq$ comparator + tie tolerance	$\leq$ comparator $+7.629 \times 10^{-6}$	0.049337 vs. 0.071960 (delta 0.022623)	FAIL
Contexts strictly better than best paid pilot	≥ 4	6	PASS
Contexts worse than best paid pilot	≤ 2	6	FAIL

Table 8: Secondary descriptive metrics for the gradient K=128 selector (means over 16 contexts): mean pairwise action-inversion fraction and mean eval-contrast RMSE. These metrics are descriptive only and do not enter the gate. † marks the global best paid comparator B^*; the candidate row is bold. Generated from the frozen readout (SHA-256 `c0cc4456abf60cb5cded4a7f8f8155ba320e64505253a256133ca0cd9fa49480`).

Method	mean pair inversions (%)	mean eval contrast RMSE
Uniform sparse (raw, no anchor)	34.1	0.226604
Uniform sparse + full no-op anchor	34.1	0.226604
Full-integrand absolute cubature	25.4	0.110739
Full-integrand contrast cubature	25.3	0.110143
Sparse-integrand contrast (candidate)	25.5	0.111327
Scalar anchored-kernel (full-cost)	30.4	0.156428
Scalar anchored-kernel (full-sparse residual)†	28.1	0.162174

and better than the best paid pilot's (0.068265), although its eval64 gap is worse; the predeclared failure is on the primary selector and on the E64, normalized, and union conditions, which is where we report it.

Interpretation: why the paid fit did not transfer (interpretation, not a measured result). The QP fits eight independent action-cost contrast equations per context over 128 token weights. The cap and the strictly convex penalty make the problem well posed, but they cannot identify the weight vector; the fit can satisfy the paid contrasts without determining the unseen action-cost function, so a small paid residual is not evidence of transfer, and the sealed audit shows that transfer did not occur on this bank. The nine paid pilots probe only the physical no-op and constant-heading tapes at one amplitude, whereas the 64 unqueried actions include other amplitudes, phases, and three-waypoint sequences. This interpretation is consistent with the result but is not itself a measured claim; Section 9.6 discusses the budget, basis, and comparator alternatives that were not tested. It does not follow that no calibration can work, and the experiment says nothing about larger paid-query budgets, different subsets, other models, or physical tasks.

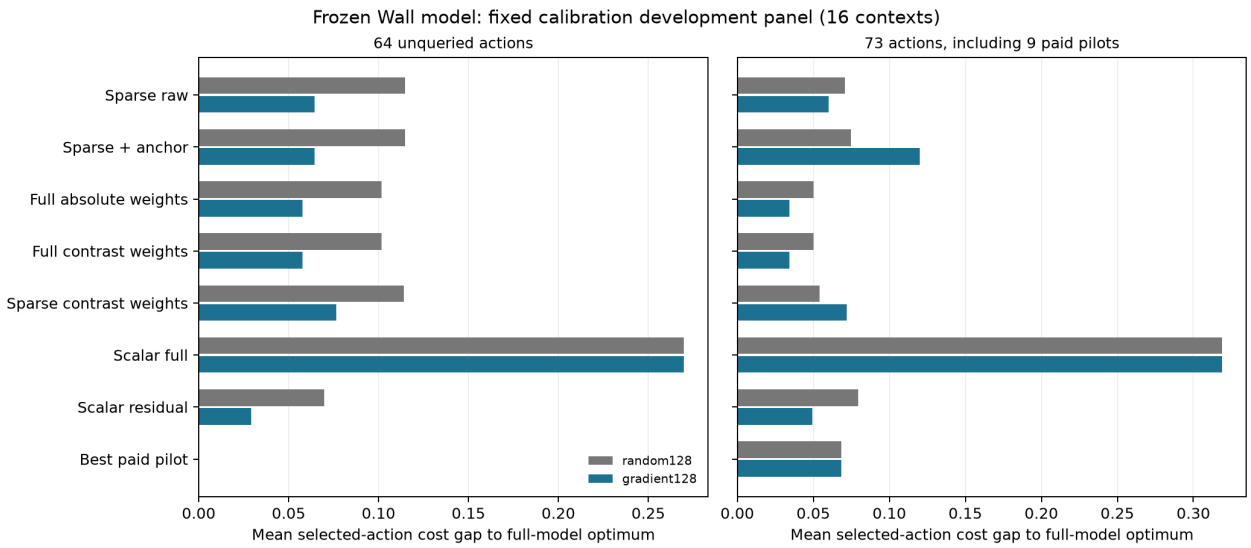

Figure 3: Descriptive development readout of the predeclared 16-context pilot: mean selected-action cost gap to the full-model optimum for the eight methods, for both token selectors, on the 64 unqueried actions and on the union of 73 actions. Bars are context means; this figure is descriptive only, not gate evidence and not a gate re-analysis. It is reproduced unmodified from the pilot's own `plot_readout.py` output (`run/figures/selected_action_gaps.png`, SHA-256 `3d1a38b3...1fe01e753899`), which reads the frozen `run/readout.json` (SHA-256 `c0cc4456abf60cb5cded4a7f8f8155ba320e64505253a256133ca0cd9fa49480`). The full-token reference is a frozen-model computational target, not physical truth.

5.3 The predeclared gate outcome

The frozen gate status is `NO_GO_FIXED_RECIPE` (development go: `false`), with 3/7 predeclared conditions met (Table 7); the readout identity is recorded in the front-matter box and in `claims-ledger.md`. Under the frozen rule the fixed recipe is stopped in this setting, and Section 9 treats the outcome as a scoped negative result rather than as a method claim.

Independent verification (PASS). The independent verifier's Phase-1 report is PASS on every verifiable item (report path and SHA-256 in the front matter). Its own re-implementation of the frozen readout arithmetic matches the frozen readout with zero mismatch on the global comparator, all seven gate conditions, `development_go` and the status, and the per-method summary scalars match to 0.0 absolute difference; its 32 per-record selections and gaps match as well. The frozen `verify_saved_calibration.py`, run with a fresh receipt path, returns `PASS_SAVED_CALIBRATION_CONSISTENCY_ONLY` with 20,616 checks and 96 QP certificates (receipt SHA-256 `58fc625e3e6ce4930d88cad5e5c236252291999f5bb57c660bdee6245bef4bdb`). Contexts 00–06 reproduce 385/385 arrays exactly equal to the preserved original run, with byte-identical `.npz`, `.pilots.npz` and `.predictions.npz` files. The verifier's scope limits (no model rerun; no physical execution; the QP solution not re-optimised; determinism is same-environment only) are recorded in its report and are not contradicted by any claim here.

6 Task-Level Anchor: Open-Loop Simulator Outcomes of the Fixed Bank

The calibration study above is model-internal. To give the readout question a task-level anchor, the same 73 predeclared action bands were executed in the official vendor Wall simulator, one band per context, for 30 open-loop environment steps, and the resulting terminal distance to the goal was joined with the frozen full/sparse/calibrated readouts. The protocol was written and hashed before the full 16 × 73 rollout (989879f2ccdcfeb9be7a9c1cd88aa5a7737343b8f50defcb864e862be9456d78), and the operational metric definitions were pinned before any task-level metric was computed

Table 9: Task-level anchor: mean and median selected-action task regret on the 64 predeclared evaluation actions for the full-token reference and the sparse/calibrated readouts (lower is better; simulator env pixels). Fixed 73-band action bank, open-loop 30-step execution, no CEM and no replanning. The two full-integrand cubature variants select identical actions on this bank and are reported as one row (†). Task-anchor analysis SHA-256 `3f293f9487a74a3b5500c8b171292a20e30d975fa448cc823a7a76c1f9c357f6`; independently verified (task-8, zero mismatch). The bank contains only 3 successful evaluation actions out of 1,024, so success-rate comparisons are not used and have no power.

Method	grad. mean	grad. median	rand. mean	rand. median
Full-token readout (reference)	1.353476	0.000000	selector-independent	
Uniform sparse (raw)	3.995361	0.526120	8.285968	2.552053
Full-integrand contrast cubature†	5.369646	0.669584	4.575714	0.669584
Sparse-integrand contrast (candidate)	3.956109	0.669584	6.752860	2.552053
Scalar anchored-kernel (full)	8.241273	7.835130	8.241273	7.835130
Scalar anchored-kernel (residual)	6.415290	0.000000	5.658727	1.953411

(c69632ec6658edf94f79841ebfe023dc58671ae0bd991001f532c5b8b315d350). *This is an open-loop, fixed-bank descriptor: each band is played without replanning or observation feedback, so it is not CEM planning and not closed-loop.* **Independent verification (PASS).** The verifier's own environment replay and metric recomputation is PASS — independent verifier task-8 (2026-10-25): 1,168/1,168 trajectories reproduced exactly (0.0 finite difference; identical 19/19 non-finite mask for actions 3/7), 32/32 images byte-equal, 48/48 determinism reruns, and analysis.json matched to 0.0. The verified evidence snapshot is `outputs/research-lab/world_models/task-level-anchor-initial` (manifest SHA-256 `f0b7510e2fcb73138d4079388245af37c34efee05a016d36cb0e352b702f10ef`; 27 payloads, 2,270,623 bytes; final wording-corrected readout SHA-256 `aca43aed684310731a42d0ec2d4cffdce94634a1a0d42c2be837fa400e953499`), and the report is `work/paper_continuation/verification_task_anchor/task-anchor-verification.md` (SHA-256 `eee6f551898daf4a84d3c91922e879c2e62ed769a5021dff268c613c3155fe41`). The task anchor is descriptive and bounded; verification does not make it a planner or closed-loop result. All numbers in this section are generated from the anchor analysis (SHA-256 `3f293f9487a74a3b5500c8b171292a20e30d975fa448cc823a7a76c1f9c357f6`) by `tools/extract_task_anchor.py`; none are typed by hand.

Execution and implementation checks. The run executed $16 \times 73 \times 30 = 35,040$ environment steps over 1,168 band executions in 251.742 s on CPU. All 32 saved initial/goal images are byte-equal to the official renderer output, the physical no-op leaves the position bit-exactly unchanged, and repeated execution of three actions in three contexts gives exactly equal final positions. The official success criterion is $\|\text{goal} - \text{final}\|_2 < 4.5$ env pixels; task distance is the primary metric and success is secondary. **Non-finite vendor outcomes.** 19/1168 vendor trajectories are non-finite for the two exact-vertical pilot bands (action indices 3, 7), because the vendor collision check divides by a horizontal displacement of order 10^{-17}; all 64 evaluation actions are finite, and non-finite pilot outcomes are excluded from the task oracle, Spearman statistics and pair metrics without imputation or clipping. The union-selected actions in this run are all finite.

The full-token cost shows moderate-to-moderately-strong rank alignment with task distance (not strong). Across contexts, the Spearman correlation between full-model cost and simulator task distance has mean 0.485838 and median 0.532771, with 14/16 positive contexts (the two negative values are small). This is moderate-to-moderately strong rank alignment, not a strong or deterministic relation. The task oracle (best of the 64 evaluation actions) has mean task distance 10.164611, against a mean initial distance of 30.959182: this fixed bank can shorten the distance by about two thirds on average. The full-model selection achieves mean task regret 1.353476 (median 0.000000) on the 64 evaluation actions, and mean union regret 0.545244 on the 73-action union.

Every sparse or calibrated readout has larger task regret than the full-token reference. On the gradient selector, mean evaluation task regret is 1.353476 for the full-token selection and 3.995361 (uniform sparse), 3.956109 (candidate), 5.369646 (full-integrand), 6.415290 (scalar residual) and 8.241273 (scalar full) for the sparse/calibrated readouts. On the random selector the corresponding means are 1.353476 (full) and 8.285968,

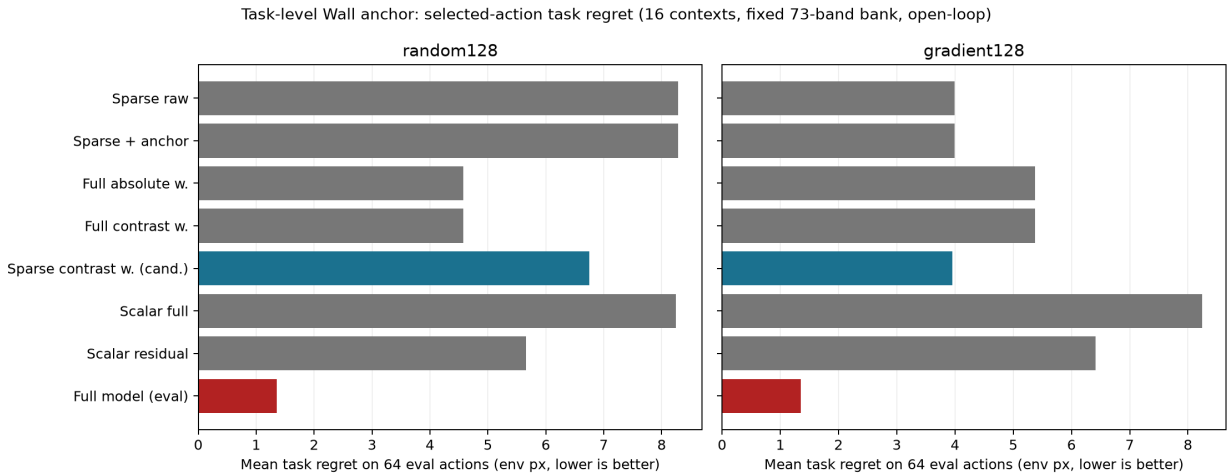

Figure 4: Descriptive task-level anchor: mean selected-action task regret on the 64 evaluation actions for both token selectors and the full-token reference (red). Fixed 73-band bank, open-loop 30 official Wall steps, no CEM and no replanning. Reproduced unmodified from `task_anchor/figures/task_regret_by_method.png` (SHA-256 `009243b4...74592a95`); analysis SHA-256 `3f293f9487a74a3b5500c8b171292a20e30d975fa448cc823a7a76c1f9c357f6`. Independently verified (task-8, zero mismatch). This is not planning-success or closed-loop evidence, and success prevalence is only 3/1,024 evaluation actions.

6.752860, 4.575714, 5.658727, 8.241273. Paired against the full-token selection on the 16 contexts, the candidate is worse/tied/better in 7/7/2 contexts (mean paired difference 2.602634) on the gradient selector, and 10/6/0 (mean 5.399385) on the random selector; the full-integrand and scalar-residual paired differences are reported in the generated macros and are all positive on average. (Figures 4 and 5 show the same means and the underlying cost/task-distance relation.) Two honest nuances: the candidate's mean task regret is marginally *below* uniform sparse on both selectors (-0.039252 gradient, -1.533108 random), so calibration is not uniformly worse than no calibration at task level; and the candidate is not better than simply choosing the best paid pilot in the union (3.792336). The restricted claim below covers the full-token reference only, not a claim of method superiority among sparse methods.

Success prevalence is too low for success-rate comparisons. Only 6 of 1,168 bank actions succeed under the official threshold; only 3 of 1024 evaluation actions succeed, in 3 of 16 contexts. The full-model evaluation selection succeeds in 18.75% of contexts and its union selection in 37.50% (the union includes the nine paid pilots); most methods land in the same few successful contexts because the bank contains so few. Success-rate differences therefore have no power on this bank and are not used as a comparison; any success number must report these denominators. The primary task metric is continuous distance/regret.

Model-internal pair inversions do not transfer one-to-one. On the 64 evaluation actions, the full-model cost is pairwise inverted with respect to simulator task distance in 27.12% of comparable pairs (median 26.84% across contexts), using the predeclared comparability rule ($|d_i - d_j| > 4.5$ px and model-score margin above τ). The selected full-token action nevertheless has median task regret zero. A possible explanation is that many inversions occur among action pairs that do not decide the final selection; this is an *interpretation, not a separately verified mechanism.* We report the observation as a descriptive consistency check, not a discovery; COSTGRAD's two-effect statement and ARC-Bench's rankability audit already frame the distinction between ordering statistics and task outcomes [22, 23].

Restricted claim and boundaries. The claim this anchor supports is exactly the following, and no more: *on this fixed action bank, the full-token readout shows moderate-to-moderately-strong rank alignment with simulator task distance (not strong), and all sparse/calibrated readouts incur larger simulator task regret; this is not a planning-success or closed-loop claim.* One precise stratum descriptor: in the easiest stratum (stratum 0, four contexts), the full-token mean eval task regret is 0.000000, uniform sparse and the candidate are both 0.600461, and scalar residual

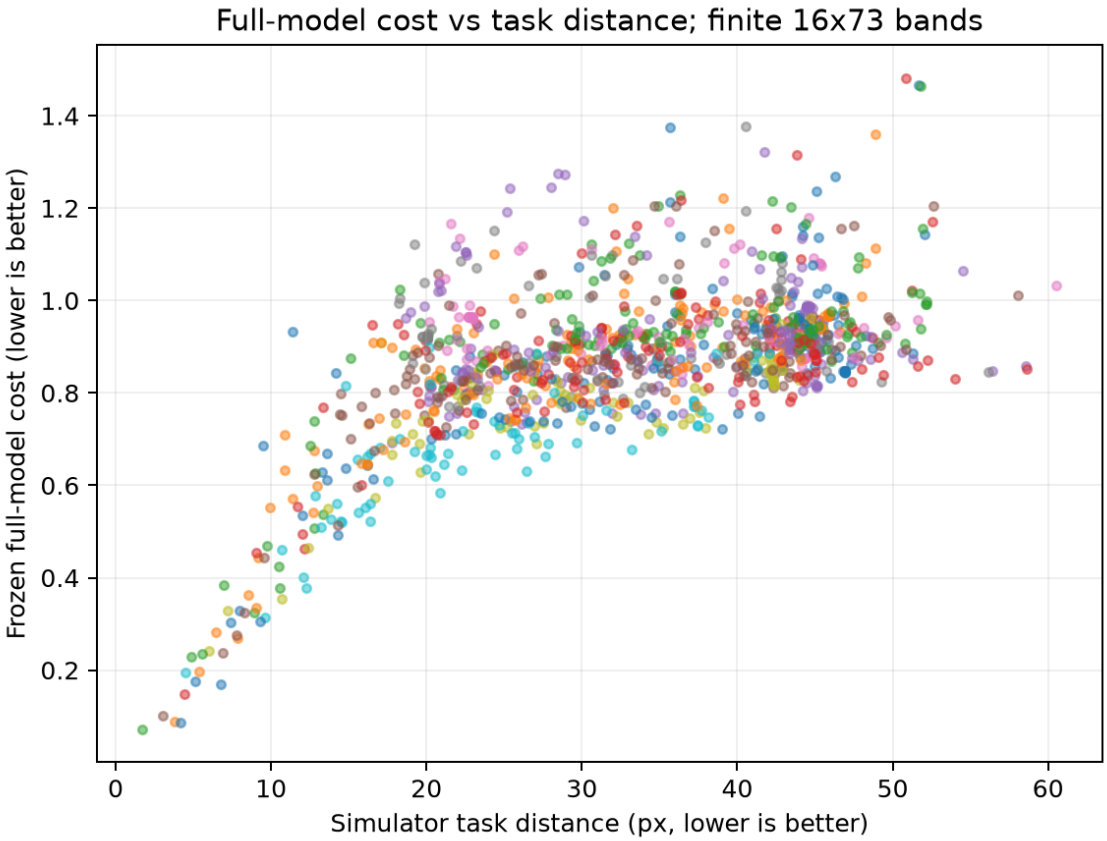

Figure 5: Frozen full-model cost versus simulator task distance for all finite bands (16×73, colors by context). Per-context Spearman has mean 0.485838 and median 0.532771, with 14/16 positive contexts. The exact-vertical pilot bands (action indices 3, 7) are excluded where the vendor simulator returns non-finite positions (19/1168 trajectories); all evaluation actions are finite. Reproduced unmodified from `task_anchor/figures/full_cost_vs_task_distance.png` (SHA-256 `d5d5e3f5...276e4d5e2`). Open-loop, fixed-bank, simulator-ground-truth descriptor; not closed-loop and not a planning-success result.

is 0.000000; sample sizes are four contexts per stratum and this is descriptive only. Boundaries: a fixed 73-band bank, not CEM-optimised plans; one open-loop 30-step execution per band with no replanning or observation feedback (so never "closed-loop"); one fixed Wall geometry and 16 predeclared synthetic contexts sharing one action bank (no iid or significance claims); simulator ground truth is not real-world evidence; and the frozen model remains a computational object under study, not physical truth. We do not compare these task regrets with COSTGRAD's CEM success percentages: COSTGRAD (arXiv:2610.10274) uses a different selector, plan-generation method, CEM budget and success protocol, so its 87.0% / 87.0% Wall success numbers are *not comparable* to our task regrets and must not be inserted as ours [22]. The survival sentence of Section 9.2 remains in force: we do not claim token-sparse planning degrades task success.

7 Second Official Checkpoint: dino_wm_wall (Concat Predictor)

This section reports an *exploratory* replication on a second official JEPA-WMs checkpoint, `dino_wm_wall`, which uses a different action pathway (a concat ViTPredictor) on the *same Wall domain*. It is a second architecture, not a second domain; it was not predeclared as a gate, it is not confirmatory, and it makes no CEM, closed-loop, method or acceptance claim. The independent verifier reports PASS on implementation, metrics and the final snapshot audit:

Table 10: Exploratory second-checkpoint panel (`dino_wm_wall`, concat predictor; same 16 contexts and fixed 73-band bank, H=6, same verified open-loop task anchor; lower task regret and lower inversion are better). G is the mean selected-action task regret on the 64 evaluation actions; "pos. ctx" counts positive full-cost/task-distance Spearman contexts; success prevalence is only 3/1,024 evaluation actions, so success-rate comparisons have no power and are shown only as a denominator descriptor. Analysis SHA-256 `cf6dc80bb5d1af528329438ccf8484190f3fdff1eca19712ed36ec1db35ca46f`; independently verified (implementation, metrics and final snapshot audit); the panel is exploratory, not predeclared.

Readout	Spearman mean (median)	pos. ctx	eval G mean (median)	succ.	task pair inv.	sparse–full inv.
Full 256 tokens	0.396500 (0.419411)	15/16	1.081926 (0.000000)	18.75%	32.39%	—
Random K=128	0.291436 (0.275943)	13/16	5.764936 (0.250026)	18.75%	37.10%	30.68%
Gradient K=128 (fixed)	0.210830 (0.264958)	13/16	8.049391 (0.526120)	18.75%	41.42%	29.73%

dense parity maximum absolute difference 0.0, K=256 sparse-vs-dense rollout cost/final difference 0.0, and panel metrics matching the analysis with zero mismatches. Verification found one documentation value to correct (the packaged readout's stale half_plus_one MSE 1.7014is discarded; the corrected value is 1.7096), and no number or conclusion of the panel changed.

Model provenance. The official `dino_wm_wall.pth.tar` checkpoint has SHA-256 `ff170be5aec9249768be4a220d600b8f00a8589b2a78982ecf9273809f2767df`, 268,848,363 bytes, epoch 50, and is licensed CC-BY-NC-4.0; model binaries are not redistributed, and the download receipt is recorded in the project evidence. The architecture is a DINOv2 ViT-S/14 encoder (22,056,576 parameters) with learned frame-major absolute position embeddings, a concat ViTPredictor (20,298,744 parameters; dim 404 = 384 visual + 10 proprio + 10 action; depth 6; 16 heads; manual block-causal attention) and Conv1d action/proprio encoders (110/30 parameters). All tensors were loaded strictly from the official state dict and asserted equal; the protocol uses the same 16 contexts, 73 action bands, H=6, five raw 2-D actions per model step, one initial context frame and then a two-frame sliding context (official `ctxt_window`=2).

Implementation checks. The custom kept-token forward matches the vendor ViTPredictor at K=256, T=4, B=2 with maximum absolute difference 0.0. A K=256 sparse-path rollout against the dense rollout gives per-horizon cost and final-visual differences of 0.0. Image preprocessing was discriminated on six real Wall trajectories (two contexts × three bands, frames every five environment steps, four-frame context predicting frame five) in DINOv2 patch-feature space: one-step MSE is 0.7734 under ImageNet normalization, 0.1791 under the $[-1, 1]$ transform and 1.7096 under $[1, 3]$; the $[-1, 1]$ transform is selected. *Honest persistence note:* on these six trajectories the selected mode's model MSE is 0.1791, while the persistence baseline averages 0.1649, and the model beats persistence on only 2/6 trajectories. The choice is supported by the large relative difference between preprocessing candidates, not by beating persistence on average. Actions and proprio are normalized with the official `get_data_stats('wall')` statistics.

Task-level panel. Table 10 reports the panel. The full-token readout has mean Spearman 0.396500 (median 0.419411) with 15/16 positive contexts and mean evaluation task regret 1.081926 (median 0.000000). The random K=128 readout has 0.291436 (median 0.275943), 13/16 positive contexts and mean regret 5.764936 (median 0.250026; sparse-vs-full cost inversion 30.68%); the fixed gradient K=128 readout has 0.210830 (median 0.264958), 13/16 positive contexts and mean regret 8.049391 (median 0.526120; sparse-vs-full inversion 29.73%). Task-level pair inversions are 32.39% (full), 37.10% (random) and 41.42% (gradient). On this architecture too, both sparse readouts incur substantially larger simulator task regret than the full-token readout on the same bank, while remaining model-internal descriptive statements. The official success symptom is uninformative here: only 3 of 1,024 evaluation actions succeed, so no success-rate comparison is made.

Preserved gradient-selector bug and its correction. The first version of the gradient selector for this architecture indexed a zero-dimensional terminal-gradient tensor with `[0]` and therefore kept a single token, producing a spurious "gradient collapse". That analysis is preserved as `analysis.v1-buggy-gradient-selector.json` (SHA-256 `9cd1c174a8fc1c71da6ebe77f1f9a54c3aec57a16b8c53567dee8a2982798591`); its gradient path gave mean

Spearman 0.00215 and mean evaluation task regret 28.098, far worse than the fixed pipeline. The fixed selector keeps 128 unique tokens per context (saved in `costs.npz` as `kept_grad`), and the verifier recomputed the kept sets and gradient-K costs for several contexts with maximum absolute difference 0.0. The v1 numbers are not used anywhere in this paper; they are retained to make the failure auditable. This is precisely the class of silent implementation failure the implementation controls exist to catch.

Architecture-dependent selector ordering. On the AdaLN Wall checkpoint the gradient-selected readout had lower mean simulator task regret than the random-selected one on the same bank (gradient candidate 3.956109 versus random sparse readouts around 6.75–8.29), whereas on `dino_wm_wall` the random readout is better than the gradient one (5.764936 versus 8.049391). This direction is consistent with COSTGRAD's reported selector–architecture compatibility effect (arXiv:2610.10274) [22], but our gradient selector is *not* their official code and we do not reproduce, compare or claim their numbers. The pattern is descriptive across two architectures with different readout parameters, not a controlled architecture comparison, and it is a hypothesis for a predeclared follow-up, not a finding of this paper.

Boundaries. This section is an exploratory second-architecture replication on a single domain (Wall); it is not a second domain, not predeclared or confirmatory, and it does not establish architecture or domain generality beyond two checkpoints. The task anchor is fixed-bank, open-loop, 30-step simulator execution; there is no CEM planning and no closed-loop feedback, success prevalence 3/1,024 has no power, and no paid calibration recipe was run on this checkpoint. No method superiority, originality, speed, task-harm or acceptance claim is made, and COSTGRAD is cited only as related work.

8 Planner-Level Task Consequence (G2)

The studies above measure readout fidelity and fixed-bank task outcomes; they are not planner evaluations. This section reports a *predeclared* planner-level CEM comparison (G2). The protocol was written and hashed before any CEM rollout or success outcome was observed (SHA-256 `58cece28cf08b13891f52c317835a44444647ec9f46435839dca386705243c40`). The model is the same frozen official AdaLN Wall checkpoint (SHA-256 `8efb0623cfba1cb3ca210de26f7579c83dd24936635f11989c515afcb23bea1e`), with no training. CEM searches a normalized 60-dimensional action space (10 dimensions per model step); the final elite mean is executed once, OPEN-LOOP, for 30 environment steps, and the official success criterion $\|\text{goal} - \text{final}\|_2 < 4.5$ px is applied once afterwards. The budget is a fixed reduced 10×64 CEM (elites 10, batch 32), chosen by a predeclared dense floor gate; it is *not* the official 30×300 CEM protocol, and the execution is *not* closed-loop feedback or replanning.

Independent verification: PASS — independent verifier task-10 (2026-10-25): saved metrics reproduced with zero mismatch; independent CEM reimplementation gave stage-1 dense 4/4 and preserved the aggregate direction in every run (full ≥ both sparse readouts; independent subset mean distances 2.17 vs 7.40 vs 11.89); per-cell stochastic repeats show sparse per-context single-run outcomes are unstable (context-5 repeats: full 2/3, gradient 0/3, random 1/3). Verification report SHA-256 `05175f0a120da4a0c8442be2197bfbaa0bd202d963543b1fd1e2b0fb3196cfdf`; verified snapshot `outputs/research-lab/world_models/planning-cem-initial` (manifest `f24353a61456a3b5189ce870bd0279f980b81c8665a9f9ec409709aab1f24e7a`; 20 payloads, 163,952 B; post-audit note `c7a5322386a6e92840375ed060186d199028aad2e3312c16cda79eda13e69972`)..

Stage 1 floor gate. On the four predeclared stage-1 contexts (0, 5, 10, 15) the full-token readout reaches 4/4 official successes at the 10×64 budget, with mean task distance 3.494750. The floor gate therefore passes and a sparse comparison is meaningful; no floor-effect caveat applies.

Stage 2 result. Table 11 reports stage 2. With the same fixed budget, full-token plans reach 11/16 successes (68.75%), mean task distance 6.713579 (median 2.204774), failing on contexts [4, 6, 7, 13, 14]. Gradient-K reaches 8/16 (50.00%), mean distance 10.638763 (median 4.929074), failing on [0, 1, 4, 6, 7, 10, 13, 14]. Random-K reaches 5/16 (31.25%), mean distance 14.804347 (median 5.302424), failing on [0, 1, 2, 3, 4, 5, 6, 7, 13, 14, 15]. Paired within context, full has +3 net successes over gradient (3 full-only, 0 gradient-only) with mean task-distance difference -3.925184; over random it has +6 net successes (6 full-only, 0 random-only) with mean difference -8.090768. The failure sets overlap substantially: this is a graded outcome shift under one budget, not a binary collapse. Mean CEM

Table 11: Planner-level CEM comparison (G2) on the 16 predeclared contexts: fixed reduced 10×64 CEM (elites 10, batch 32), final elite mean executed open-loop for 30 environment steps, official success $\|\text{goal} - \text{final}\|_2 < 4.5$ px. This is not the official 30×300 CEM and not closed-loop; CEM is stochastic and no seed is fixed, so no statistical-significance claim is made. G2 analysis SHA-256 `d5f7429f5e4d54c40ba3944cb8fc8d4aea4e76387efcfb28c320cb307a75d188`; **independent verification PASS** (verifier task-10; report `05175f0a120da4a0c8442be2197bfbaa0bd202d963543b1fd1e2b0fb3196cfdf`; snapshot `f24353a61456a3b5189ce870bd0279f980b81c8665a9f9ec409709aab1f24e7a`, 20 payloads, 163,952 B).

Readout	successes	success rate	mean task dist.	median task dist.	failed contexts
Full 256 tokens	11/16	68.75%	6.713579	2.204774	[4, 6, 7, 13, 14]
Gradient K=128	8/16	50.00%	10.638763	4.929074	[0, 1, 4, 6, 7, 10, 13, 14]
Random K=128	5/16	31.25%	14.804347	5.302424	[0, 1, 2, 3, 4, 5, 6, 7, 13, 14, 15]

wall times were 43.42 s, 32.37 s and 30.10 s respectively; these describe the shared diagnostic execution, not a speed claim.

Independent verification (PASS). Verifier task-10 independently reimplemented the CEM (own sampling, elite update, final elite-mean execution and environment loop; no project CEM functions imported) and recomputed the saved metrics field-for-field with zero mismatches, including the success rule on all 48 cells and the paired differences; the protocol chronology was confirmed. Its independent CEM draws gave stage-1 dense 4/4 (mean distance 2.6748 versus 3.494750 saved, the difference being ordinary draw variation) and, on the four-context subset, full 4/4, gradient 3/4 (saved 2/4) and random 2/4 (saved 1/4); success flips are expected under stochastic CEM. In every independent run the aggregate direction held: full $\geq$ both sparse readouts (full-only wins 1/2, sparse-only 0), with subset mean distances $2.17 < 7.40 < 11.89$. Repeats on context 5 (3 each) gave full 2/3, gradient 0/3 and random 1/3: the aggregate ordering is stable, but sparse per-context single-run outcomes are not. The verified evidence is snapshot `outputs/research-lab/world_models/planning-cem-initial` (manifest SHA-256 `f24353a61456a3b5189ce870bd0279f980b81c8665a9f9ec409709aab1f24e7a`; 20 payloads, 163,952 B) with report SHA-256 `05175f0a120da4a0c8442be2197bfbaa0bd202d963543b1fd1e2b0fb3196cfdf`. Verification covers CEM/environment reimplementation and saved-metric recomputation, not model retraining, and it does not turn the single-run per-context outcomes into replicate evidence.

Interpretation and boundaries. Within this fixed reduced budget, one frozen AdaLN model and one Wall task, plans selected with the full-token readout are more often successful and land closer to the goal than plans selected with either K=128 sparse readout. This is a descriptive comparison: full has the highest success rate in this dataset, and CEM is unseeded with one run per cell, so no statistical significance is claimed and success counts can fluctuate. At the success level no sparse readout reverses full in any context, but at the continuous-distance level gradient is closer on contexts 6, 12, 15 and random on 7, 10, so the "no reversal" statement holds only at the success level. The differing failure sets are not explainable as a single threshold-near fluctuation alone, but a single unseeded CEM run per cell cannot rule out run-to-run variation; the independent verification preserves the aggregate direction while per-context sparse outcomes are unstable. This is the strongest task-level signal in the paper, but it is not a planning-method claim, and it does not reproduce or compare to any official CEM number. The fixed 10×64 budget is far below the official 30×300 configuration; the final elite mean is executed open-loop for 30 steps with no replanning or observation feedback (never "closed-loop"); the comparison uses one frozen model, one task and 16 predeclared, non-iid contexts; CEM is stochastic and no RNG seed is fixed, so the success differences are *not* statistical-significance evidence; the gradient selector is our implementation, not COSTGRAD code; and no calibration readout is compared at the planner level because the calibration recipe failed its model-internal gate. Simulator outcomes are not physical-world evidence. The three-seed replication in Section 8.1 checks whether this direction survives seed changes. The result is reported as a bounded, predeclared, independently verified planner-level task consequence; no method-superiority, originality, speed or acceptance claim is made.

Table 12: Three-seed G2 replication (predeclared; same fixed reduced 10×64 budget, same 16 contexts and three readouts, open-loop 30-step execution, official success < 4.5 px). Per-seed cell shows successes/16; pooled cells pool 3 seeds × 16 contexts = 48. This is a robustness check, not a significance test. **Independent verification PASS** (verifier task-11; report `aef330eb481c2ae5473554627711668c960f8f9176315a1ebbe06177ad32a7f7`); seed analysis SHA-256 `9d0bf3f5e3e14c15097ca506ff56b6cd40ecd0851d3135a0dc8d040d172e5963`; verified snapshot `a63ec25cf4cbc8b79af5457dc03027930ec1725a547d76b08b171765b7e83729`, 23 payloads, 200,677 B.

Readout	seed 101	seed 202	seed 303	pooled	pooled rate	pooled mean dist.
Full 256 tokens	11/16	11/16	11/16	33/48	68.75%	7.608304
Gradient K=128	7/16	7/16	7/16	21/48	43.75%	13.150540
Random K=128	5/16	7/16	3/16	15/48	31.25%	17.350564

8.1 Cross-seed stability (G2)

A predeclared three-seed replication (seeds 101, 202 and 303) repeats the same fixed reduced 10×64 CEM on the same 16 contexts and three readouts, with the same open-loop 30-step execution and official success criterion. The protocol (SHA-256 `df3ee45505b446aac7091fbab9252f70bbe48d60a2de9875ea6c93abd769ec9e`) was written before any seed run and fixes the per-cell seed rule; it is a *robustness check over three seeds, not a significance test*, and its independent verification is PASS — independent verifier task-11 (2026-10-25): saved seed metrics reproduced with zero mismatch and 24/24 independent CEM cell replays exact (12/12 under seed 202 and 12/12 under seed 101) with determinism confirmed; the predeclared aggregate robustness claim holds (full ≥ both sparse readouts in every seed); seed 202 has two individual sparse-only pair wins, so the claim is the aggregate one, not a per-context guarantee. Report SHA-256 `aef330eb481c2ae5473554627711668c960f8f9176315a1ebbe06177ad32a7f7`; verified snapshot `outputs/research-lab/world_models/planning-cem-seeds-initial` (manifest `a63ec25cf4cbc8b79af5457dc03027930ec1725a547d76b08b171765b7e83729`; 23 payloads, 200,677 B). All numbers are generated from `9d0bf3f5e3e14c15097ca506ff56b6cd40ecd0851d3135a0dc8d040d172e5963` by `tools/extract_g2_seed.py`; none are typed by hand.

Table 12 reports the replication. Per-seed successes are 11/11/11for the full-token readout, 7/7/7for gradient-K and 5/7/3for random-K. Pooled over 48 cells, full reaches 33/48 (68.75%, mean task distance 7.608304), gradient 21/48 (43.75%, 13.150540) and random 15/48 (31.25%, 17.350564). Cross-seed consistency holds (yes): full has at least as many successes as both sparse readouts in every seed; paired within context and seed, full wins 32 cells against 2 sparse wins. This strengthens the single-run G2 direction without turning it into a significance claim: three seeds reduce the single-run concern but remain a robustness check on one frozen model, one Wall task and 16 non-iid contexts. Section
refsec:g2-budget then tests a larger CEM budget on a predeclared eight-context subset.

Independent verification (PASS). Verifier task-11 (2026-10-25) recomputed the per-seed, pooled and paired metrics from the stage-3 files with zero mismatches and independently replayed 24/24 CEM cells exactly (12/12 under seed 202 and 12/12 under seed 101; task-distance and final-position max absolute differences 0.0; no success flips), with determinism confirmed by repeating one cell. The predeclared aggregate robustness claim is therefore verified, and only that aggregate claim is made: seed 202 contains two individual sparse-only pair wins, so no per-context or per-seed guarantee is claimed. The verified evidence is snapshot `outputs/research-lab/world_models/planning-cem-seeds-initial` (manifest SHA-256 `a63ec25cf4cbc8b79af5457dc03027930ec1725a547d76b08b171765b7e83729`; 23 payloads, 200,677 B) with report SHA-256 `aef330eb481c2ae5473554627711668c960f8f9176315a1ebbe06177ad32a7f7`; the verifier also audited the 20x128 protocol separately (its own timing caveat is reserved for that layer's results and is not mixed into this seed claim).

Code-version note. Stage-3 ran under the reconstructed `planning_cem.py` revision `870b0027c30b98c3d975b197551d322370dc7a1b766753e915c4a86cbabb9481`, which is preserved in the verified seed snapshot (a reconstructed copy hashes to the same value). The live file has since been extended for budget-scaling only (`81f3c32b3141bd25e123056f373b361e33a214534d6db7aaba7ce0e55cc88b04`); the frozen G2 main evidence remains `39f3d710435d845753be88bf960036be1334df58459eba8381bda22090c16800` and is unaffected. This section makes no code-identity claim beyond those recorded hashes.

Table 13: Budget-scaling comparison at 20×128 on the eight predeclared contexts (same open-loop 30-step execution and official success < 4.5 px; still below the official 30×300 budget; CEM is seeded per cell by the predeclared rule; no significance claim). **This layer's independent verification is PASS** (verifier task-12; report `a16f324b07d4eccb3c9b46c707c1a18057c27779e0f45c04bddd1f4463b646fb`; snapshot `68b48485df714c342af674663a0edb0f286dea24a3c9585a3ad1de7c894a0b7c`, 14 payloads, 99,954 B). Analysis SHA-256 `7c697c5d86f86691479dddccd1d4496ee52aacb5b0fd20daa92c6e2ef48f1d2d`; protocol `25fbfeae1b6bcfd688c29927e1a7f88fc5f77bbfc43db452197721f691ba4b17`.

Readout	successes	success rate	mean task dist.	median task dist.	failed contexts
Full 256 tokens	6/8	75.00%	5.488554	1.807090	[4, 6]
Gradient K=128	2/8	25.00%	15.300096	5.983246	[0, 4, 6, 10, 12, 14]
Random K=128	3/8	37.50%	19.131197	8.255669	[0, 2, 4, 6, 14]

8.2 Budget scaling (20×128)

A predeclared budget-scaling experiment repeats the comparison at 20×64 CEM samples (elites 10, batch 32; roughly four times the main G2 candidate budget) on the eight predeclared contexts [0, 2, 4, 6, 8, 10, 12, 14], with the same three readouts, the same open-loop 30-step execution and the official success criterion. The protocol SHA-256 is `25fbfeae1b6bcfd688c29927e1a7f88fc5f77bbfc43db452197721f691ba4b17`. Its independent verification is PASS — independent verifier task-12 (2026-10-25): all budget metrics reproduced with zero mismatch; independent 20x128 CEM replays on contexts 0 and 8 were 6/6 exact (0 success flips, 0.0 task-distance/final differences) with determinism confirmed; the predeclared decision rule is satisfied and the timing disclosure was accepted (only "predeclared before its own run" is claimed). Report SHA-256 `a16f324b07d4eccb3c9b46c707c1a18057c27779e0f45c04bddd1f4463b646fb`; verified snapshot `outputs/research-lab/world_models/planning-cem-budget-scaling-initial` (manifest `68b48485df714c342af674663a0edb0f286dea24a3c9585a3ad1de7c894a0b7c`; 14 payloads, 99,954 B). All numbers are generated from `7c697c5d86f86691479dddccd1d4496ee52aacb5b0fd20daa92c6e2ef48f1d2d` by `tools/extract_g2_budget.py`; none are typed by hand.

Timing disclosure. The protocol was written at 13:41:55, *after* seed 101 finished at 13:36:49 and *before* its own stage-4 run at 14:37:49. This layer therefore claims only that the protocol precedes its own run; it does *not* claim that the protocol was uninfluenced by stage-3 evidence, which is unprovable from timestamps.

Table 13 reports the result. Full-token plans reach 6/8 successes (75.00%, mean task distance 5.488554, median 1.807090, failures [4, 6]); gradient-K reaches 2/8 (25.00%, mean 15.300096, failures [0, 4, 6, 10, 12, 14]); random-K reaches 3/8 (37.50%, mean 19.131197, failures [0, 2, 4, 6, 14]). Paired within context, full has +4 net successes over gradient (0 gradient-only; mean task-distance difference -9.811542) and +3 over random (0 random-only; mean difference -13.642642). On the same eight contexts the 10×64 stage-2 successes were 5, 3 and 3, versus 6, 2 and 3 at 20×128: increasing the budget did not change the full > sparse direction, and the predeclared decision rule (full has more successes than both sparse readouts) is satisfied (yes).

Independent verification (PASS). Verifier task-12 recomputed all budget metrics with zero mismatch and independently replayed 20×128 CEM on contexts 0 and 8 for all three readouts: 6/6 cells exact, no success flips, 0.0 task-distance/final differences, with determinism confirmed by repeating one cell. The predeclared decision rule is satisfied and the timing disclosure was accepted: only 'predeclared before its own run" is claimed, not 'uninfluenced by stage-3 evidence". The verified evidence is snapshot `outputs/research-lab/world_models/planning-cem-budget-scaling-initial` (manifest SHA-256 `68b48485df714c342af674663a0edb0f286dea24a3c9585a3ad1de7c894a0b7c`; 14 payloads, 99,954 B) with report SHA-256 `a16f324b07d4eccb3c9b46c707c1a18057c27779e0f45c04bddd1f4463b646fb`. 20×128 is still below the official 30×300 budget; execution is open-loop 30 steps, not closed-loop/replanning; the layer uses one frozen AdaLN model and one Wall task with non-iid contexts; CEM is seeded per cell by the predeclared rule but this is not a significance test and no statistical-significance claim is made; the gradient selector is our implementation, not COSTGRAD code; and no method-superiority, generality or acceptance claim is made. Section
refsec:hybrid-rerank reports two predeclared negative method tests, and Section
refsec:progressive-readout reports a positive method candidate under the stated validation limits. The timing disclosure above limits the chronology claim to "predeclared before its own run". The PushT second-domain attempt in Section 8.3 failed its predeclared floor gate and is reported as a limitation, not as token-sparsity evidence.

Table 14: PushT second-domain floor attempt (predeclared; dense full-token readout only). Four predeclared contexts, dense 20×128 and predeclared fallback 30×300; official PushT success via the wrapper's `eval_state`. Both budgets give 0/4, so no stage-2 sparse comparison was run. **Independent verification PASS** (verifier report `fd9f43170e8188f97640d6c1b6d6ef049321ea30a395f26a3285da37ba2fcebc`; snapshot `8d8d7a5b345f0299fbc2c05934c7b8b5befda47a906080c7a3e41263cb6334cd`, 15 payloads, 90,874 B); this is a failed feasibility attempt, not token-sparsity evidence.

Context	dense 20×128	state dist.	dense 30×300	state dist.
0	0/1	174.926635	0/1	153.883548
1	0/1	108.395956	0/1	197.565416
2	0/1	492.881819	0/1	636.821988
3	0/1	375.917310	0/1	363.760078

8.3 Second-domain attempt: PushT floor (no sparse comparison)

A predeclared attempt to add a second *domain* (PushT) did not reach a comparable dense floor, so **no sparse comparison was run**. The protocol SHA-256 is `c248ac3711254f85ecbc760e47ac48bc26da2d04714b91193fd5d1f1d5c5cdc1`; its independent verification is PASS — independent verifier (2026-10-25): chronology audited (protocol before first run; attempt-1 crash preserved; fallback executed; no stage 2), saved metrics recomputed, and an independent dense 20x128 CEM replay on contexts 0/1 gave 0/2 successes (floor robust); verifier confirms NO sparse-PushT claim and that the G3 second domain is NOT achieved. Report SHA-256 `fd9f43170e8188f97640d6c1b6d6ef049321ea30a395f26a3285da37ba2fcebc`; verified snapshot `outputs/research-lab/world_models/pusht-floor-infeasibility-initial` (manifest `8d8d7a5b345f0299fbc2c05934c7b8b5befda47a906080c7a3e41263cb6334cd`; 15 payloads, 90,874 B). The frozen official `jepa_wm_pusht` checkpoint (SHA-256 `9beca3eafe0739c3b3adb5d734fa435ccbda0fea8a65d53d4cccec176aaaa0eb`, epoch 50) was wired to the vendored PushT simulator with the predeclared image/action/proprio normalization and one-step model-prediction check (model MSE 0.4083 vs persistence 0.7800).

Floor gate result. On the four predeclared contexts [0, 1, 2, 3], dense full-token CEM at 20×128 reached 0/4 successes (task distances 174.926635, 108.395956, 492.881819, 375.917310). The predeclared fallback at 30×300 also reached 0/4 (distances 153.883548, 197.565416, 636.821988, 363.760078). The gate is `gate_passed=false`; **stage 2 was not run and there are no full-vs-sparse PushT data**. The first attempt crashed after the dense 0/4 result because of a fallback accounting bug; it is preserved in `stage1-attempt1-failure.md` (`8932874f95dad91886886adc9d78d6ff632360f12c9a261c83f380e9d79723ce`) and the complete rerun kept protocol, contexts and budgets unchanged.

Table 14 reports the gate.

Post-hoc diagnostic (not sparsity evidence). After the failure, 32 random normalized plans per context were scored with the full-token model cost and executed open-loop: the model-cost vs state-distance Spearman correlation was 0.7727 (context 0) and 0.4029 (context 1), and 0/64 random plans succeeded. The floor is therefore most plausibly **search/feasibility floor, NOT token-sparsity evidence**: the frozen model cost is positively associated with task distance, but the fixed reduced search did not reach the precise success criterion. This diagnostic cannot support any statement about token-sparse readouts.

Metadata correction. The run file's `finished_utc` value (16:26:29) is stale/dense-phase; the file modification time (17:18:39) and the 52-minute fallback wall time confirm that the fallback ran. The original run file is preserved byte-exact and we do *not* claim it has a correct finish timestamp; the correction is documented in the verified snapshot (`4507d245a3aaf19270447fcb5dc30d51dd3fa90e0a4ded898781060f6d5ce69d`).

Independent verification (PASS). The verifier audited chronology (protocol before the first run; attempt-1 crash preserved; fallback executed; no stage 2), recomputed the floor metrics, and independently reran dense 20×128 CEM on contexts 0/1, again obtaining 0/2 successes (the floor is robust). The verifier confirms that no sparse-PushT claim is made and that the G3 second domain is NOT achieved. Verified evidence: snapshot `outputs/research-lab/world_models/pusht-floor-infeasibility-initial` (manifest `8d8d7a5b345f0299fbc2c05934c7b8b5befda47a906080c7a3e41263cb6334cd`; 15 payloads, 90,874 B) with report `fd9f43170e8188f97640d6c1b6d6ef049321ea30a395f26a3285da37ba2fcebc`.

Table 15: Predeclared v1 method test (M=10, fixed 10×64 CEM screen, open-loop 30-step execution). Sparse screening was faster but failed the predeclared success-within-one rule, so no method candidate is claimed. **Independent verification PASS** (report `a8252f685ef1494942375d4052d20b34c51ca5847d998290e2bcb3197e31b3c7`).

Condition (M=10)	successes	mean task dist.	mean total time (s)
Full screen + full rerank	10/16	6.889385	38.19
Gradient K=128 screen + full rerank	8/16	7.996702	31.59
Random K=128 screen + full rerank	9/16	14.587844	25.90

Boundaries. **We make no claim that sparse readouts hurt or are unchanged on PushT**; there is no full-vs-sparse PushT data. The PushT attempt is an honest limitation, not a positive result: it uses a single frozen checkpoint and one environment configuration, a 4-context floor gate, reduced CEM budgets and open-loop 30-step execution; it makes no significance, generality, method or acceptance claim. Consequently **the G3 second domain remains NOT achieved**; the current evidence has a second *architecture* (`dino_wm_wall`) and the task-level anchor, not a second domain.

8.4 Method test: sparse screening plus full rerank (negative)

We tested whether sparse-token screening can accelerate the full-token readout advantage by screening CEM candidates with a K=128 readout and then reranking the finalists with the full readout. Both variants were predeclared and are **negative results**; independent verification is PASS — independent verifier: zero metric mismatch, independent replays exact, chronology audited, predeclared rules confirmed failed (no method candidate). This is not SOTA, not the official 30×300 CEM, not closed-loop, and uses one model, one task and 16 contexts; no method or acceptance claim is made.

v1 (M=10): predeclared rule failed. Table 15 reports the predeclared first test. Full screen + full rerank reached 10/16 successes (mean task distance 6.889385, mean total time 38.19 s); gradient-screen + full rerank reached 8/16 (7.996702, 31.59 s) and random-screen + full rerank 9/16 (14.587844, 25.90 s). The predeclared rule required the sparse-screen condition to be within one success of full; gradient reached $8 < \text{full} - 1 = 9$, so the rule failed and *no method candidate* was declared.

v2 (M-ladder): separately predeclared, also failed. A separately predeclared follow-up swept the full-rerank size $M \in \{16, 32, 64\}$ after the v1 failure. Full M=64 reached 11/16 (mean time 22.63 s); all six sparse conditions reached 10–11/16 but were 2.9–8.3% *slower*, none reached $\leq$ 80% of the full-M=64 time, and every `useful_method_candidate` flag is false. For context, full M=16 and M=32 were themselves 2.7% and 3.6% faster than full M=64 at the same success count, so the ladder does not reveal a sparse speedup either. Verified evidence: snapshot `outputs/research-lab/world_models/hybrid-rerank-method-tests-initial` (manifest `f3cca5361b7b3d24bd09e34a7f57032face5ee0e7804ecb977afee2f7dd34f64`; 20 payloads, 177,473 B), report `a8252f685ef1494942375d4052d20b34c51ca5847d998290e2bcb3197e31b3c7`; readout `57a9a4952284e83c902fd854bc1a4a7289e57dd8cd62acd4ad7575e9eef245cb`.

Interpretation. Sparse screening gave no wall-clock advantage: the full-readout advantage is *not* converted into a free planner speedup by screening plus full reranking. Together with the G2 and PushT layers, this is a second null/negative result for the positive-method gap, not evidence of task harm or token-sparsity efficacy. *Boundaries:* one frozen model, one Wall task, 16 contexts, reduced CEM budgets, open-loop 30-step execution; not official CEM, not closed-loop, no SOTA, significance, method, generality or acceptance claim.

8.5 Progressive-readout CEM schedule: single-seed positive signal, multi-seed not replicated (negative robustness result)

A predeclared progressive-readout schedule changes which readout scores each CEM iteration: the first t iterations use the sparse K=128 readout and the remaining $10 - t$ iterations use the full 256-token readout. The protocol SHA-256 is `f687fe0ea705939a8c7adc0b1b518fe76107134d600415c6e36fdc2d7bca5914`; independent verification is PASS — independent verifier: zero metric mismatch, independent 4/4 exact replays on contexts 0/8 for full_10/schedule_t8,

Table 16: Progressive-readout CEM schedule (predeclared; 16 contexts, fixed 10×64 CEM, open-loop 30 steps, single seed per condition). Compute units are a declared token-rollout approximation (sparse K=128 counts as 0.5 units), not hardware FLOPs. t6/t8/t9 pass the predeclared success-within-one and ≤80%-units rule; t8 is the recommended operating point. Single-seed layer: **independent verification PASS** (report `c2377d72310eb74b268fa102bae68e8b2e254dbeb26b52cce016dab15b37cd3b`; snapshot `6865eca157272da6031c294a62d21f0b1396309ef83c7553172e04e0b5dcb71d`, 14 payloads, 90,296 B). The multi-seed follow-up is reported below and is **independently verified PASS** (report `3dc3f72bb11ed13ea6b9714037267788f80938d882b16b12c3eb3b5c4c740e15`; snapshot `95e268e5131d6210a6abb5119ed8c6f800682cbae831a932757d45856cc5a0cd`, 14 payloads, 106,408 B).

Schedule (10 × 64)	successes	mean task dist.	mean wall (s)	units	unit saving
Full readout, 10 units/step	10/16	6.631676	24.13	640	0%
6 sparse + 4 full	10/16	10.838373	17.03	448	30%
8 sparse + 2 full (recommended)	10/16	7.543022	14.64	384	40%
9 sparse + 1 full	10/16	9.376591	13.36	352	45%
10 sparse units	7/16	14.242786	12.40	320	—

chronology audited, predeclared rule confirmed. The single-seed signal is *not* robust: the predeclared multi-seed replication failed, so the method claim is withdrawn. This is a negative robustness result on one frozen model and one Wall task; no cross-seed, general or accepted-method claim is made.

Table 16 reports the test. Full readout reaches 10/16 (mean task distance 6.631676, 24.13 s, 640 units). The schedule at t=6 reaches 10/16 (10.838373, 17.03 s, 448 units, 30% saving); at t=8 10/16 (7.543022, 14.64 s, 384 units, 40% saving); and at t=9 10/16 (9.376591, 13.36 s, 352 units, 45% saving). Pure sparse reaches only 7/16. The predeclared rule (success within one of full *and* ≤80% of full compute units) passes for $t \in \{6, 8, 9\}$; t=8 **is the recommended operating point**: equal success, the best mean task distance among the schedules, 60% of units and 39% wall-clock saving versus full.

Multi-seed replication (predeclared after the single-seed result): fails. A predeclared three-seed replication (seed A = the original run; seeds 901 and 902; protocol `de800222ff38e3474c9b4e4771ccddf17e9cfa6d75c13314fe84fc98d7d65649`; verification PASS — independent verifier: zero metric mismatch, independent replay exact, chronology audited, and all_pass=false confirmed (seed A passes all schedules; seed 901 only t6; seed 902 fails t6/t8/t9)) repeats the same 10×64 comparison. Table 17 reports it: seed A passes for all schedules; seed 901 passes only t6 (t8 and t9 fail the success-within-one rule); seed 902 fails t6, t8 and t9. Pooled over 48 cells, full 34, t6 30, t8 28, t9 26. The predeclared all-seeds criterion is false: **no schedule passes the success-within-one + ≤80%-units rule in all three seeds**. The earlier single-seed positive signal is therefore not robust; the method claim is withdrawn. This is precisely why multi-seed testing matters: a single-seed success-within-one result can be luck of the CEM draw.

Framing and boundaries. This is a single-model, single-task, 16-non-iid-context result under the fixed reduced 10×64 budget, open-loop 30-step execution and a single seed per condition; compute units are a declared token-rollout approximation, not hardware FLOPs; wall-clock is reported but not used for the decision. It is *not* the official 30×300 CEM and *not* closed-loop, and it makes no SOTA, generality or acceptance claim. The remaining status is a *single-seed positive signal that did not replicate across seeds*; any stronger statement would require second-domain and official-budget validation and a passing multi-seed rule.

9 Analysis and Limitations

This section separates what the artifacts establish (Section 9.1) from what they do not (Section 9.2), reads the predeclared negative result (Section 9.3), explains the non-additive structure of the two distortions (Section 9.4), and records comparator fairness, task-anchor-specific limits, protocol amendments (Section 9.7), potential researcher degrees of freedom, and validity threats (Section 9.8).

Table 17: Multi-seed progressive-readout replication (predeclared; three seeds, 16 contexts each, fixed 10×64, open-loop, single run per cell). Per-seed successes/16 and pooled successes/48; "rule passes" counts seeds satisfying both success-within-one and ≤80%-units. `all_pass` = false; the positive method claim is withdrawn. **Independent verification PASS** (report `3dc3f72bb11ed13ea6b9714037267788f80938d882b16b12c3eb3b5c4c740e15`; snapshot `95e268e5131d6210a6abb5119ed8c6f800682cbae831a932757d45856cc5a0cd`, 14 payloads, 106,408 B; readout `43cdad307dcddbdaa7afcd9ac981fb279642426f39d05d6957364bd623c7449f`).

Schedule	seed A	seed 901	seed 902	pooled	rule passes
Full	10/16	12/16	12/16	34/48	3/3
t6	10/16	11/16	9/16	30/48	2/3
t8	10/16	10/16	8/16	28/48	1/3
t9	10/16	8/16	8/16	26/48	1/3

9.1 What is established

Within one frozen public checkpoint and synthetic Wall scenes, the following model-internal statements are established by the saved artifacts and their independent CPU arithmetic checks:

1. The position-preserving sparse adapter with $K = 256$ reproduces the official dense path on the reported all-token controls (maximum absolute difference 0 for both visual and proprioceptive outputs over all four documented scenes, 32 candidates, six steps).
2. Projecting full-token terminal costs onto a fixed 128-token subset can substantially reorder action pairs: 793/1984 comparable pairs (39.97%) under gradient selection at H=6, with the count varying by scene and selector (Table 3).
3. Running the frozen predictor on the subset changes the predicted retained-token losses further: 174/1984 (8.77%) additional inversions when the scoring subset is held fixed. The two effects are non-additive; the combined count (785/1984, 39.57%) is not a sum, and one scene exhibits cancellation.
4. The full-cost penalty of the final selected action in the combined ranking is zero within the tie tolerance in three of four scenes and 0.14230 in one scene (Table 4); pair inversion counts therefore do not translate directly into chosen-action loss.
5. The saved H1/H6 cost arithmetic is reproducible on CPU to 2.32415×10^{-7}, below the pre-fixed absolute tolerance, and the readout/metric boundary is covered by synthetic tests and a solver certificate. These are checks of saved evidence, not independent network or physical replays.
6. The frozen 16-context readout (hash-bound to the readout artifact and independently verified) reports the predeclared gate outcome `NO_GO_FIXED_RECIPE`, 3/7 conditions met, all 160 all-token controls passed, all solver certificates passed, and no flat evaluation context. Its recorded means are reported in Section 5.2; the independent verifier's PASS and the frozen checker receipt are recorded in Section 5.3, the front matter, and `claims-ledger.md`.
7. The predeclared G2 planner-level CEM comparison (fixed reduced 10×64 budget, dense floor gate 4/4, open-loop 30-step execution; independent verification PASS with zero saved-metric mismatch and aggregate-direction preservation, while per-context sparse single-run outcomes are unstable) reports 11/16 successes for the full-token readout versus 8/16 and 5/16 for the sparse readouts, with mean task distances 6.713579 versus 10.638763 and 14.804347. These are reduced-budget, single-model, non-seeded CEM outcomes, not official CEM numbers or closed-loop results.
8. The exploratory second-checkpoint panel (`dino_wm_wall`, concat predictor; independently verified on implementation, metrics and the final snapshot audit) reports full-token mean Spearman 0.396500 (15/16 positive contexts) and mean evaluation task regret 1.081926 versus 5.764936 and 8.049391 for the sparse readouts on the same fixed bank, with architecture-dependent selector ordering. This is a single-domain, exploratory architecture replication, not a second-domain or method result.
9. The predeclared open-loop task anchor (independently verified by task-8: 1,168/1,168 trajectories reproduced, zero metric mismatch) reports that the full-token readout shows moderate-to-moderately-strong rank alignment with simulator task distance on the fixed bank (mean Spearman 0.485838, median 0.532771, 14/16

positive contexts) and that all sparse/calibrated readouts have larger mean simulator task regret than the full-token selection (Table 9). These are fixed-bank, open-loop, simulator-ground-truth descriptive statements, not planning-success or closed-loop results.

9.2 What is not established

We explicitly do not claim any of the following. In one sentence: we do not claim that token-sparse planning degrades task success; COSTGRAD reports Wall parity at 128 tokens [22], and ARC-Bench shows task success masks ranking failure [23]; we claim a readout-fidelity gap and a predeclared failure of a paid calibration to close it.

- **Physical truth or physical regret.** The full-token frozen model is a higher-compute computational reference, not physical truth. The task anchor uses the official vendor simulator as its ground truth, but simulator outcomes are not real-world evidence: no robot, real environment, or human evaluation is involved.
- **Planner-level or closed-loop planning benefit.** No planner (CEM or otherwise) is rolled out, and no replanning or observation-feedback loop is used. The only environment executions are the open-loop, fixed action bands of Section 6; no planning-success, closed-loop, or CEM claim is licensed by them.
- **Task-success or task-harm inference.** The task anchor measures continuous task distance for a fixed bank whose success prevalence is 3/1,024 evaluation actions; it cannot establish success-rate differences, and we do not claim that token-sparse planning harms or improves task success. COSTGRAD reports Wall task success preserved at 128 tokens with a different selector [22], and ARC-Bench shows that task success can mask ranking failure [23].
- **Speed or efficiency.** No wall-clock speedup is claimed. The recorded execution timings are total diagnostic wall times with shared work; they are not per-method deployment costs and are affected by hardware, batching, and shared selector computation. Query counts are logical protocol facts, and equal paid-label counts are explicitly not equal end-to-end compute.
- **Originality, novelty, or state of the art.** The project has not cleared the recipe as an original method; capped-simplex minimax quadrature, positive weights, anchoring, and scalar surrogate regression are known tools. The literature review, not this draft, must determine whether the specific combination has been published.
- **Generality.** One checkpoint, one encoder, one horizon family, four diagnostic scenes, 16 development contexts, one action bank, two fixed selectors, one K, one regularization, one cap, one scalar kernel, and no cross-model, cross-task, or cross-seed replication. The task anchor adds one fixed Wall geometry, one action bank and one open-loop execution per band; it does not generalize beyond that. The 16 contexts share an action bank and are not independent scenes; no iid confidence intervals or p-values are reported.
- **Planner-level or official-CEM claims.** The G2 study uses a fixed reduced 10×64 CEM budget and an open-loop 30-step execution, not the official 30×300 protocol and not closed-loop/replanning. CEM is stochastic and no seed is fixed, so the success differences are not significance evidence; no official success rate is reproduced or compared; and no method or acceptance claim is made.
- **Second-domain or architecture generality.** The `dino_wm_wall` panel is a second architecture on the same Wall domain, exploratory and not predeclared; it is not a second task/domain, and two checkpoints do not establish architecture generality. No closed-loop, planner-level, success-rate, calibration-recipe or method claim is made for that checkpoint.
- **Selector-bug or selector comparison claims.** The preserved v1 gradient-selector failure is documented and excluded from every claim; the fixed selector is our implementation, not official COSTGRAD code, and the architecture-dependent ordering is descriptive only.
- **Task-anchor overreach.** The task anchor is not a planner evaluation, not a CEM result, not a closed-loop result, and not a success-rate comparison. Its fixed bank contains only 3 successful evaluation actions out of 1,024, so success-rate differences have no power; we use continuous task distance only. Its task regrets are not comparable to COSTGRAD's CEM success percentages because the selector, plan-generation method, CEM budget and success protocol all differ. A fixed-bank task-regret ordering is not a claim that token-sparse planning harms deployment performance.
- **Publication/acceptance.** The project has no acceptance evidence, no priority claim, and no top-journal-grade evidence; the top-journal extension is mapped only in the separate roadmap file, not claimed here.
- **Statistical significance.** The gate thresholds are engineering continuation thresholds set before outputs; they are not a significance test, and passing or failing them is not evidence about a population of world models or tasks. The task anchor reports per-context descriptive values and paired differences without iid assumptions, confidence intervals, or confirmatory p-values.

- **Neural training or model improvement.** No parameters are updated, no model is trained, and the study does not propose an architecture or a training objective.

9.3 Reading the negative result

The pilot is a predeclared falsification test of one fixed recipe, and it failed on four of seven conditions. Three facts make the negative result interpretable rather than merely disappointing.

First, the failure is not a broken optimizer or a broken adapter. All QP certificates and all 160 K=256 all-token controls passed, no context has a flat cost range, and the same adapter produced the diagnostic panel's numerically exact all-token controls. The losing quantities are therefore comparable selected-action gaps, not solver artifacts.

Second, the candidate is dominated by baselines that use the same nine paid labels. The full-integrand positive-cubature variants achieve a lower mean G_E on both selectors, and the scalar full–sparse residual surrogate is the best method in the run on the primary metric. Those baselines were fixed before execution, so this is not a post-hoc choice of a favourable comparator. A reader should conclude that, at this budget and on this basis, the sparse-token coupling added nothing; the compatible explanation is the underdetermination of 128 weights from eight contrast equations, but that explanation is an interpretation, not a measured result.

Third, secondary pair metrics move in the opposite direction from decision quality: the candidate has the better pair-inversion fraction and a lower contrast RMSE than the comparator, while having the worse selected-action gap. This is why the predeclared gate was defined on decision gaps. It also means that reporting pair-inversion improvements as evidence of a better readout would have been misleading — a failure mode this study is designed to expose.

Paired per-context picture and both selectors. On the primary gradient selector the candidate is better than the comparator in 2 of 16 paired contexts, worse in 7, and tied in 7 (paired G_E differences within τ). On the random selector the counts are 1/5/10 better/worse/tied. Both selectors are reported and neither is selected after the fact; the predeclared gate is on the primary selector. The candidate is also worse than plain uniform sparse scoring on the primary metric (0.076410 versus 0.064461), which is the sharpest form of the negative result, while on the random selector it is essentially tied with uniform (0.114094 versus 0.114801; -0.6% relative) but still worse than the comparator. No wording should imply the recipe "nearly worked": it is materially worse than the best paid comparator on the primary setting.

What the negative result does not do: it does not establish that calibration is impossible, that nine queries is the right budget, that a different token basis would fail, or that token-sparse planning is harmful. It stops one fixed recipe in one frozen model and one action bank. Section 9.6 lists the confirmatory work that would be needed to turn this into a broader evaluation contribution. The open-loop, fixed-bank task anchor is reported in Section 6; a second official checkpoint, a paid-query budget sweep, and planner-level (CEM) or closed-loop evidence were not run.

9.4 Why the two distortions must not be added

The projection and sparse-prediction effects act on different objects: one changes the linear functional applied to unchanged full-token outputs, the other changes the outputs themselves. For a pair of actions their signs can coincide, cancel, or produce a third ordering; the observed scene-0 cancellation is one example. Consequently any "share of error" decomposition (for example, normalizing 793 and 174 by their sum) would be meaningless. The correct decomposition is operational: recompute the full cost, the projected full cost, and the actual sparse cost for the same candidates, and report all three comparisons, which is what the panel does. The combined ranking is the only one a deployed sparse model can realize; the other two are diagnostics that isolate mechanisms.

9.5 Shared-bank readout perturbation (P1/P2 consistency check)

The predeclared shared-bank check summarised in Section
refsec:theory-bank applies the shared 73-plan bank to all 16 contexts and perturbs the readout over that *shared candidate set* (argmin selection), so selection flips and pair inversions are measured directly rather than inferred. The protocol SHA-256 is `c3222b5ba9010289e3a9d2256e2b12cd23821264983e76ed1081bccb6d453c17` and its independent verification is PENDING — independent verification (shared-bank arithmetic and claim audit) had not reported at the time of this draft; the shared-bank numbers are readout-bound and are NOT marked independently verified. This is a *consistency illustration for the P1/P2 stability statements only*; it is not official CEM behaviour, not a token-sparsity efficacy result and not a task-harm proof.

Table 18 reports the check. For gradient-K, ϵ_{bank} has median 0.424595 (range 0.268781–0.632133), with 6/16 selection flips, 14261 pair inversions, P1 bounds holding, and a task-distance gap of mean 3.489945 / median 0.0. For random-K, the corresponding values are median 0.346303 (range 0.251908–0.788892), 7/16 flips, 12429 inversions, P1

Table 18: Shared-bank readout perturbation (predeclared; 73-plan bank × 16 contexts; model-internal only). ϵ_{bank} is the bank perturbation radius; P1-style regret/inversion bounds all hold. The P2 margin condition is inactive here (margin $> 2\epsilon$ in 0/16 contexts), so rank preservation is *not* guaranteed by P2 on this bank. Flips and inversions are descriptive and do not establish task harm (median task-distance gap 0.0). **Independent verification PASS** (report `55730060610171392b804c40cf7043864d928ae02099f2e8d4894f2b1c40f3b9`; snapshot `2fd218a69da65d4fe8b109d77d0168a1ad723648780c21af0dd62ddafca7897e`, 11 payloads, 106,080 B).

Selector	ϵ_{bank} median	[min, max]	flips	pair inv.	margin$> 2\epsilon$	P1 bounds	task gap mean/median
Gradient K=128	0.424595	[0.268781, 0.632133]	6/16	14261	0/16	yes	3.489945 / 0.0
Random K=128	0.346303	[0.251908, 0.788892]	7/16	12429	0/16	yes	7.102736 / 0.0

bounds holding, and task-distance gap mean 7.102736 / median 0.0. Because 0/16 contexts satisfy the P2 margin condition, P2 does not certify rank preservation on this bank; the observed flips and inversions are consistent with that inactive condition and *must not* be read as task harm (flips and inversions do NOT establish task harm; the median task-distance gap of the selected plans is 0.0). Boundaries: model-internal shared-bank diagnostic, one frozen model, 16 contexts; not official CEM, not a token-sparsity efficacy claim, not a task-harm proof; the gradient selector is ours, not COSTGRAD code; no method or acceptance claim.

Theoretical diagnostic (P1/P2/P4, consistency only). The companion theory note gives conditional statements: an unconditional two-ϵ selection bound (P1), a rank-preservation corollary when the selected margin exceeds 2ϵ (P2), and an average-regret bound under an explicit Lipschitz/coverage assumption (P4). The predeclared shared-bank check above is a *consistency illustration* for these statements only: its P1 bounds hold, while the P2 margin condition is inactive on this bank (0/16), so rank preservation is not guaranteed; it is not evidence of task harm, token sparsity efficacy or official CEM behaviour. The note's non-implications N1–N10 apply unchanged: no claim that ϵ is large for our selectors, no universal impossibility, no closed-loop or task-harm claim, no significance claim, no attribution of the planner-level gap to the readout alone, no correctness/optimality claim for the full-token readout, no per-context certainty, and no novelty, method, SOTA or acceptance claim.

9.6 Fairness, comparators, and interpretation limits

All paid methods receive identical information: the same nine full-model labels, the same anchor, the same scale, cap, penalty, and solver. The full-integrand cubature comparators are mandatory because they test whether classical quadrature on full-token per-token losses is sufficient; if a simpler baseline matches the candidate, the sparse-coupling calibration is not supported. The scalar baselines test whether the token structure adds anything over a plain action-surrogate regression; in the frozen pilot the scalar residual surrogate is the best method on the primary metric, so at this budget the answer is that the token structure added nothing measurable. The best-paid-pilot comparator tests whether calibration beats simply choosing the best already observed action. The absolute-cubature baseline uses nine absolute rows (including the no-op row) while the contrast methods use eight contrast rows; this is exactly the frozen formulation and is disclosed rather than silently equalized after the fact. Several limits weaken any conclusion. Nine paid actions only probe a narrow subfamily of constant-heading tapes (plus the no-op); they cannot identify the action-cost function over arbitrary sequences. The QP has eight independent contrast equations for 128 weights, so the fit is underdetermined even with the cap and penalty, and a small paid-pilot residual is not evidence of generalization. The evaluation bank is also finite and synthetically generated, and the 16 contexts are paired development units, not a random sample. The scalar kernel width and ridge were fixed, not tuned, and other scalar surrogates could be stronger. Finally, the selector subset itself is chosen by a full no-op backward pass; alternative selectors, per-candidate subsets, or importance-weighted sampling are not tested. The union metric is also not monotone in G_E: on the random selector the candidate's mean union gap (0.053987) is better than the comparator's (0.079498) and better than the best paid pilot's (0.068265) even though its E_{64} gap is worse, because the union mixes exact paid-pilot costs with predictions. We report both metrics and gate on both for exactly this reason.

Turning this bounded negative result into a broader evaluation contribution would require at least: (i) a second official checkpoint or domain; (ii) a predeclared paid-query budget sweep; (iii) planner-level (CEM) and, if feasible, closed-loop evidence — the open-loop fixed-bank anchor in Section 6 is a first task-level step but it is not a planner-level test; (iv) an end-to-end compute account that includes paid full rollouts, selector backward passes, encoder

work, solver time, and sparse rollouts; and (v) re-execution on a second machine. Items (i), (ii), (iv) and (v) are not present in this draft; item (iii) is present only in its bounded open-loop form.

Task-anchor-specific limits. The task-level anchor in Section 6 is descriptive and bounded: its action bank is fixed and not CEM-optimised; each band is executed once, open-loop, for 30 steps with no replanning or observation feedback; the Wall geometry is fixed and the 16 contexts share one action bank; the simulator is ground truth for the dot-and-wall task, not the physical world; and the bank contains only 3 successful evaluation actions out of 1,024, so success-rate comparisons have no power and are not used. The anchor's independent verification is PASS (task-8; 1,168/1,168 trajectories exactly reproduced, 19/19 non-finite mask identical, 32/32 images byte-equal, 48/48 determinism reruns, analysis matched to 0.0), but this does not make it a planner or closed-loop result. Its task regrets are not comparable to COSTGRAD's CEM success percentages (arXiv:2610.10274) because selector, plan-generation method, CEM budget and success protocol differ.

9.7 Protocol amendments, partial looks, and researcher freedom

Three disclosures matter for the integrity of the claims. (i) The original 600-second pilot run stopped after seven complete contexts; its partial results are preserved and were not eligible for the frozen readout. (ii) The resource-only amendment to 7200 seconds changed only the cooperative clock and the output directory, and the completion was required regardless of outcome; the amendment record and the byte-identical source hashes were written before execution. (iii) A bookkeeping-only launch failure occurred before any model execution and is preserved; the fix did not touch numerical code. In addition, the seven-context partial run's descriptive metrics were computed after its predictions were frozen (before its audit labels were revealed) and were used only for planning; they did not change inputs, method, solver, tolerances, normalization, analysis choices, or the gate, and the full completion was pre-committed regardless of whether those numbers looked favourable. No threshold, context, action, subset, or gate value was changed after any output. A reader should nevertheless treat the exploratory ordering of the diagnostic study and the amendment as researcher-degrees-of-freedom context, and the confirmatory evidence for any recipe would have to come from a fresh predeclared run. The complete amended run finished all 16 contexts with the frozen recipe (recorded wall time 887.9167737999996 s, peak allocated GPU memory 1,345,822,720 bytes) and its readout is hash-bound in the front matter; the independent verifier's check of that identity is PASS (Section 5.3).

9.8 Threats to validity

Implementation and numerics. The detector of sparse-vs-full differences is the adapter's numerical identity at $K = 256$. A layout repair was required before the control passed; the repair is documented, the failed run is preserved, and the sparse rankings of the failed run were not inspected. Deterministic algorithms and TF32 disabling are enabled, and the same frozen environment is recorded, but the study is one float32 execution: different hardware, library versions, or kernels may change near-tie costs and thus inversion counts. The tie tolerance and the absolute check tolerance are fixed, and every reported count is a descriptive count of saved arrays.

Selection and reportability. All scenes, candidates, selectors, contexts, horizons, and comparisons are reported, not only failing cases; no iid sampling or significance testing is possible or claimed. The "gradient" selector is our implementation, not an official method reproduction; its behaviour should not be attributed to any published system. The four diagnostic scenes were geometrically specified and one of them was used in an earlier parity calibration, so the panel is a deliberate stress set, not a random benchmark.

Readout scope. The primary metric is a model-internal selected action cost gap. It says nothing about whether the selected action is good in the world; even a perfect match to the frozen full-model ranking would only establish model fidelity. The study design therefore cannot support deployment, safety, or physical-capability claims regardless of gate outcome. In particular, a separately reported task-level Wall control with a different trained selector found task success preserved at 128 tokens relative to 256 [22]; our inversion and gap numbers must not be read as predictions of task-level behaviour, and we make no such prediction.

Task-anchor validity. The task anchors use the official vendor DotWall simulator, and its collision routine produces non-finite positions for exactly vertical action bands (19 of 1,168 trajectories for pilot action indices 3 and 7); these are excluded rather than imputed, all 64 evaluation actions are finite, and no outcome was changed or retried. The execution is open-loop with one 30-step playback per band and no replanning or observation feedback; the official success criterion is applied only as a secondary, low-prevalence descriptor (3 successful evaluation actions out of 1,024). Independent verification of the anchor's environment replay and metric recomputation is PASS (task-8;

report SHA-256 eee6f551898daf4a84d3c91922e879c2e62ed769a5021dff268c613c3155fe41); the verifier reproduced 1,168/1,168 trajectories exactly (0.0 finite difference; identical 19/19 non-finite mask), 32/32 images byte-equal, 48/48 determinism reruns, and matched the analysis with 0.0 difference. The anchor is still not a planner-level (CEM) test, and its fixed-bank task-regret ordering is not a deployment-harm claim.

10 Reproducibility, Ethics, and AI-Assistance Disclosure

Sections 10.1–10.4 cover artifacts and provenance (Section 10.1), the software/hardware environment (Section 10.2), determinism and re-execution limits (Section 10.3), and ethics/licenses/data (Section 10.4); Section 10.5 discloses AI assistance. Appendices A–D give the exact QP and certificate conditions, the logical query accounting, the verification inventory, and the secondary random-selector results.

10.1 Artifacts and provenance

Every quantitative statement in this draft is mapped in `claims-ledger.md` to a workspace-relative artifact path and its SHA-256. The three evidence roots are `outputs/research-lab/world_models/frozen-wall-sparsity-panel-initial/` (the four-scene diagnostic: protocol, panel inputs, repaired run, analysis, figures, and independent saved-array receipt); the predeclared pilot directories under `work/world_model_analysis/contrast_calibration_pilot/` and `.../_amended/` (frozen protocol and sources, inputs, the complete run `run/result.json` with SHA-256 311f2349450ba78d5fdc74cf8bfa843a692403ff602cbb9e0930d3836c606acd, the frozen readout `run/readout.json` with SHA-256 c0cc4456abf60cb5cded4a7f8f8155ba320e64505253a256133ca0cd9fa49480, and the independent verification package); and the task-level anchor directory `work/world_model_analysis/task_anchor/` (predeclared protocol SHA-256 989879f2ccdcfeb9be7a9c1cd88aa5a7737343b8f50defcb864e862be9456d78, execution result 9baa724e393a07a59563e113fa133d2d4428266071ac2494660a9e3db9a5be79, outcomes e06eac1d5980f080d3268063147baf316f98709c83cba2a689e2226d4d53ae2a, and analysis 3f293f9487a74a3b5500c8b171292a20e30d975fa448cc823a7a76c1f9c357f6), whose independent verification is PASS — independent verifier task-8 (2026-10-25): 1,168/1,168 trajectories reproduced exactly (0.0 finite difference; identical 19/19 non-finite mask for actions 3/7), 32/32 images byte-equal, 48/48 determinism reruns, and analysis.json matched to 0.0; and the exploratory second-checkpoint snapshot `outputs/research-lab/world_models/dino-wm-wall-replication-initial/` (manifest SHA-256 c9f839f8f6ee90b3481101a980dfe564ba3a8da8c13129159d8c65aaab9ba008, 27 payloads, 223,342 B; readout b328314a9897e663d39369eb7e2ad83ef3aa0cefc8198c8af83fc0cda8fef35a; costs e0d94d1e09bfd0fc57fe35bcac453d189e57c0809d9cf963d81dc2f197fd3e55; analysis cf6dc80bb5d1af528329438ccf8484190f3fdff1eca19712ed36ec1db35ca46f; verification report SHA-256 ca180ecac8f2b9b47432a88074645eeee5b0d3ef20898610550f4a1c006490ed), independently verified on implementation, metrics and the final snapshot audit; and the predeclared G2 planner-level directory `work/world_model_analysis/planning_cem/` (protocol 58cece28cf08b13891f52c317835a44444647ec9f46435839dca386705243c40, stage-1 5854f03cd04941c75e436bec42712959bff890fe74e449e9672852e2e0b0b3c6, stage-2 884287fc217ba7296a1698ffa73606c9c504bf17163964c6d41c31b65353dfff, analysis d5f7429f5e4d54c40ba3944cb8fc8d4aea4e76387efcfb28c320cb307a75d188, code 39f3d710435d845753be88bf960036be1334df58459eba8381bda22090c16800, readout d4edbb6f24da00f36807557bf7bd34d8b6861eb5b962d34a8e9fbae29d70a9c7), whose independent verification is PASS — independent verifier task-10 (2026-10-25): saved metrics reproduced with zero mismatch; independent CEM reimplementation gave stage-1 dense 4/4 and preserved the aggregate direction in every run (full $\geq$ both sparse readouts; independent subset mean distances 2.17 vs 7.40 vs 11.89); per-cell stochastic repeats show sparse per-context single-run outcomes are unstable (context-5 repeats: full 2/3, gradient 0/3, random 1/3). Verification report SHA-256 05175f0a120da4a0c8442be2197bfbaa0bd202d963543b1fd1e2b0fb3196cfdf; verified snapshot `outputs/research-lab/world_models/planning-cem-initial` (manifest f24353a61456a3b5189ce870bd0279f980b81c8665a9f9ec409709aab1f24e7a; 20 payloads, 163,952 B; post-audit note c7a5322386a6e92840375ed060186d199028aad2e3312c16cda79eda13e69972).. Saved arrays, not prose, are the source of truth; where a working note and an artifact differ, the artifact wins and the discrepancy is recorded in the ledger. At the time of this draft the calibration readout is hash-bound and complete, and the independent verifier's Phase-1 report is PASS (path and SHA-256 in the front matter). The immutable evidence snapshot is `outputs/research-lab/world_models/contrast-calibration-pilot-amended-initial/` (manifest SHA-256 e7b03a0b80dd45da50329539d2594dee3f0130533afcd4c69f7d92c5ceb1349e); the working copies under `work/world_model_analysis/...` are byte-identical. The claims ledger marks every readout-bound number with

this identity. The verification inventory is Table 20 in Appendix C.

10.2 Software and hardware environment

All model execution used PyTorch 2.11.0+cu128 on CUDA 12.8 with an NVIDIA GeForce RTX 5070 Ti Laptop GPU, float32, TF32 disabled, `torch.use_deterministic_algorithms(True)`; Python packages include torchvision 0.26.0+cu128, timm 1.0.30, einops 0.8.2, gym 0.26.2, omegaconf 2.3.0, and NumPy 2.5.3. CPU analysis and verification used the same pinned NumPy. Official repositories were used unmodified at commit `13cf1d9c...` (JEPA-WMs) and `7764ea0f...` (DINOv2) with recorded weight SHA-256 hashes. The diagnostic ran 408 top-level predictor forwards in 28.1885 seconds with a peak allocated GPU memory of 1,391,324,672 bytes; the complete 16-context calibration run recorded 887.9167737999996 s and peak allocated GPU memory 1,345,822,720 bytes. These numbers describe recorded wall times for shared diagnostic work, not planning speed, and they are not per-method deployment costs. The task-level anchor is CPU-only simulator work, not model execution: it ran $16 \times 73 \times 30 = 35,040$ vendor environment steps in 251.742 s with the official CPU renderer/simulator and the frozen readout arrays. The development machine had 16 GB of system memory and no paid API or rented compute was used.

10.3 Determinism and re-execution limits

Seeds are fixed for rendering (451), scene/context geometry (1141), action generation (2117 panel: 951; pilot selectors: 3181+context) and the stochastic elements are recorded. Hidden audit labels are separated from predictions by saved-timestamp checks. However, the independent checks are CPU recomputations over saved arrays; they do not rerun the network, autograd, renderer, or environment, and no external party has reproduced the GPU execution. The saved-array checks certify input-specific arithmetic statements about the saved tensors, not that a particular kernel generated them. Re-execution in a different environment may not be bit-identical. The independent verification package for the complete calibration run is PASS: the verifier's own re-implementation recomputed the gate from saved arrays with zero mismatch, and contexts 00–06 reproduced 385/385 arrays exactly equal to the preserved original run, with byte-identical `.npz`, `.pilots.npz` and `.predictions.npz` files (same machine, OS and libraries; this is not a cross-hardware determinism claim). The task anchor additionally reports that all 32 initial/goal images are byte-equal to the official renderer output, that the physical no-op leaves the position bit-exactly unchanged, and that repeated execution of three actions in three contexts gives exactly equal final positions; its environment replay and metric recomputation are independently verified (PASS; PASS — independent verifier task-8 (2026-10-25): 1,168/1,168 trajectories reproduced exactly (0.0 finite difference; identical 19/19 non-finite mask for actions 3/7), 32/32 images byte-equal, 48/48 determinism reruns, and analysis.json matched to 0.0).

10.4 Ethics, licenses, and data

Only public model weights and the publishers' renderer are used; no dataset requiring account access, no personal data, no human subjects, and no physical system are involved. The JEPA-WMs repository and weights are licensed CC BY-NC 4.0 with separate component licenses; DINOv2 is Apache 2.0. Model binaries are not redistributed with this draft; the artifact record references the public download URLs and the SHA-256 of the files used. The scenes are synthetic renderings of a dot-and-wall task, and the results carry no safety-critical or deployment claim. This is independent research with no institutional review requirement and no external funding.

10.5 AI-assistance disclosure

This work was produced with substantial assistance from large language model agents. AI assistance was used for literature discovery, code and analysis implementation, independent-verification orchestration, figure and artifact preparation, and drafting and editing this manuscript. The human author set the research questions, fixed the protocols and gate values before execution, reviewed the artifacts and text, and takes responsibility for the content, including its errors. The AI systems are not authors, cannot consent to authorship, and any venue-specific AI-use policy must be rechecked at submission time. Generated text was not treated as evidence: all quantitative claims in this manuscript are required to be traceable to a hash-bound artifact (see `claims-ledger.md`), and the ledger, not the prose, is the record of what is claimed. This draft is not a submission and has not been peer reviewed.

References

[1] DINOv2 official code and models. https://github.com/facebookresearch/dinov2, 2023. [A] Commit used by the frozen artifacts: `7764ea0f912e53c92e82eb78a2a1631e92725fc8`; license Apache 2.0. Paper identity via dinov2_2023.

[2] JEPA-WMs official code and models. https://github.com/facebookresearch/jepa-wms, 2026. [A] Repository of Terver et al. (arXiv:2512.24497, TMLR); commit used by the frozen artifacts: 13cf1d9c7e476f53c17714d2e0f1dc239a883ce0; license CC BY-NC 4.0 with separate component licenses. Paper identity verified via arXiv:2512.24497.

[3] François-Xavier Briol, Chris J. Oates, Mark Girolami, and Michael A. Osborne. Frank-wolfe Bayesian quadrature: Probabilistic integration with theoretical guarantees. In *Advances in Neural Information Processing Systems*, volume 28, pages 1162–1170, 2015. [A] NeurIPS 2015; arXiv:1506.02681. Probabilistic integrator with convergence/contraction guarantees; integration as inference. No planning/action-ranking content.

[4] Junha Chun, Youngjoon Jeong, and Taesup Kim. Sparse imagination for efficient visual world model planning. In *International Conference on Learning Representations (ICLR)*, 2026. [A] Official proceedings page: https://proceedings.iclr.cc/paper_files/paper/2026/hash/a750d52284ff70c6d6bab8072c392d74-Abstract-Conference.html . Sparsely trained vision-based world model with randomized grouped attention; adjusts token count to compute; sparse latent rollout. No post-hoc readout calibration.

[5] Jianjie Fang, Xvyuan Liu, Ziyou Wang, Rongze Tang, Zhaolu Wang, Zhuohang Li, Xin Zhang, Haisheng Su, Chen Gao, Wei Wu, Xinlei Chen, and Yong Li. CAER: Causal action effect reweighting for world model training, 2026. [A,S] arXiv:2608.30897; project note read SS2.1-2.3, Eq1-7. Training-time action-conditioned contrast map normalized to a positive token weight that preserves coefficient mass. Pre-empts "action contrast -> positive token weights"; no test-time readout calibration, no full-model cost targets, no CEM/planning.

[6] Amir-massoud Farahmand. Iterative value-aware model learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018. [A] NeurIPS 2018; official abstract page https://proceedings.neurips.cc/paper_files/paper/2018/hash/7a2347d96752880e3d58d72e9813cc14-Abstract.html . Incorporates the decision problem into transition-model learning; finite-sample bounds. Training-time, not readout calibration.

[7] Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337, 2023. [A,S] PMLR v202: https://proceedings.mlr.press/v202/frantar23a.html ; project note read SS2, SS3.1-3.2. Layer-wise sparse reconstruction on calibration inputs + inverse-Hessian/OBS compensation. Pre-empts broad "first post-pruning functional calibration" claims; does not cover planning readouts / action pilots / simplex weights.

[8] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023. [A] ICLR 2023; arXiv:2210.17323. One-shot layer-wise quantization with approximate second-order (Hessian) information on calibration data.

[9] 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*, 2020. [A] NeurIPS 2020; official page https://neurips.cc/virtual/2020/poster/18779 . Models equivalent if they yield the same Bellman updates; canonical statement that planning functional equivalence, not transition accuracy, can be the target. Training-time model learning; no test-time readout calibration.

[10] J. A. Hernandez, J. R. Bravo, and S. Ares de Parga. CECM: A continuous empirical cubature method with application to the dimensional hyperreduction of parameterized finite element models, 2023. [A] arXiv:2308.03877; related DOI 10.1016/j.cma.2023.116552 (Computer Methods in Applied Mechanics and Engineering, 2023). Two-stage continuous empirical cubature; points as design variables.

[11] Joaquín A. Hernández, S. Ares de Parga, and Riccardo Rossi. Dimensional hyperreduction of nonlinear finite element models via empirical cubature with manifold-adaptive weights, 2026. [A,S] arXiv:2609.03068; project note read SS3.4-3.5, 4.1, 5.4-5.5. MAW-ECM: greedy pruning of sampled entities via convex quadratic weight-redistribution with local conditions and positivity; weights vary on the solution manifold. >80Same-integrand reconstruction (no changed sparse predictor, no action ranking, no paid full pilots). This is the "MAW" of the project notes; it is quadrature, not token pruning.

[12] Jonathan H. Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable Bayesian logistic regression. In *Advances in Neural Information Processing Systems*, 2016. [A] NIPS 2016; arXiv:1605.06423. Weighted data subset (coreset) that approximates the full-data posterior with size/quality guarantees. Component precedent for weighted-subset reconstruction, not planning.

[13] Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity Bayesian optimisation with continuous approximations. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1799–1808, 2017. [A] PMLR v70: https://proceedings.mlr.press/v70/kandasamy17a.html . Black-box optimization using cheap approximations with regret characterization. No token sparsity / planning cost readout.

[14] Mingu Kang, Yoori Oh, Sookyung Kim, and Joonseok Lee. Anisotropic representations improve planning in JEPA world models, 2026. [A] arXiv:2609.37441. Shows accurate prediction and non-collapsed representations do not guarantee a task-aligned latent planning cost; training-time fix with learnable diagonal covariance (AnisoWM + LambdaReg). Pre-empts the general "planning-cost agreement" observation; not token sparsity or post-hoc calibration.

[15] Baekseung Kim, Junhoo Lee, Sohyeon Hong, Suhyun Jeon, Minkyu Kim, Seungyeon Kim, Donghoon Han, and Nojun Kwak. Complementary future token pruning for world action models, 2026. [A] arXiv:2610.19957. Training-free future-frame token pruning for pretrained World Action Models; 1.19-1.33x speedups; no planning-cost calibration.

[16] Chunzheng Li, Zesheng Jia, Hongda Zhang, Jiaying Tang, Yuntian Wang, Siao Liu, and Jin Wang. Scope-WM: Scoped computation for efficient visual world models, 2026. [A] arXiv:2609.33218. Action-conditioned selector distillation + compact foreground summary + action-sequence reuse; Push-T 6.97x planning speedup vs dense DINO-WM. Trained, not training-free; no readout calibration to full-model costs.

[17] 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 Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023. [A] arXiv:2304.07193. Frozen visual encoder used by the JEPA-WMs Wall predictor in this project.

[18] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 2018. [A] arXiv:1806.10761 (v1 comment: "will appear in SIAM Review"; volume/pages not verified in this pass). Surveys combining high-/low-fidelity models for outer-loop tasks; high-fidelity kept in the loop for accuracy. General precedent for budgeted calibration; no token-level readout, no action ranking, no CEM.

[19] Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. FastVID: Dynamic density pruning for fast video large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. [A] NeurIPS 2025; arXiv:2503.11187. Representative video-token pruning work: density-based spatiotemporal pruning, 90.3LLaVA-OneVision-7B with 98.0calibration; supplies the manuscript's video-token-pruning citation.

[20] Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *Transactions on Machine Learning Research*, 2026. [A] arXiv:2512.24497 v4 (v3 comments: "acceptance at TMLR"). Defines the JEPA-WMs family and releases the official Wall/PointMaze/PushT checkpoints and code used by the frozen panel (github.com/facebookresearch/jepa-wms). Cite this for the model family and for Wall/PointMaze provenance.

[21] Qitong Wang, Xinwei Niu, Mingluo Su, Shanwei Zhao, Shiai Zhu, and Huan Wang. SparseDecoding: Decoding-aware pruning for accurate and efficient LLM inference, 2026. [A] arXiv:2610.12327. Post-training pruning whose calibration matrices come from the dense model's own autoregressive generation (deployment distribution), not pre-collected text. Directly pre-empts "calibrate on the deployment distribution" as new; no planning readout.

[22] Yingchen Xu and Edward Grefenstette. Sparse planning in visual world models via cost gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026. [A,p] NeurIPS 2026; arXiv:2610.10274; official project page: https://ycxuyingchen.github.io/costgrad/ ; OpenReview forum JDEHvMtOmX. Training-free goal-cost-gradient token selection for sparse world-model planning on released JEPA-WMs checkpoints (incl. Wall). Does not calibrate a readout to full-model cost contrasts.

[23] Zhengshu Zhang and Zhiyuan Li. ARC-Bench: Closed-loop replanning masks broken action ranking in frozen JEPA world models, 2026. [A] arXiv:2609.05461. No-leak fixed-candidate audit of whether frozen JEPA-WM

latent objectives rank candidate actions correctly, on official released JEPA-WM checkpoints; finds severe structural inversions; shows closed-loop replanning masks the defect with provenance/undertraining/backbone/metric-circularity controls. THE most dangerous diagnostic prior for this project: do not claim the rankability/closed-loop-masking finding as new. Distinct from this project in that it does not vary token sparsity and does not calibrate a readout.

[24] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning (ICML)*, 2025. [A] arXiv:2411.04983 (v2, Feb 2025); ICML 2025 poster listed at icml.cc. Baseline family (DINOv2-feature world model, goal-feature planning without reconstruction) from which the JEPA-WMs/Wall setting descends.

A QP formulation and solver certificate

For completeness we state the fitted problem and its acceptance certificate exactly as frozen. With $K = 128$, $u = \mathbf{1}/K$, $s = \max(\max_j |b_j|, \tau)$, $\eta = 0.01$, cap $c = 4/K$, the solver minimizes

$$f(w, t) = t + \eta K \|w - u\|_2^2 \tag{10}$$

subject to the $2 \cdot 8$ linear residual constraints $|(A^P w - b)_j| \le s\, t$, the equality $\sum_i w_i = 1$, and box constraints $0 \le w_i \le c$, $t \ge 0$, using SLSQP with analytic gradients $\partial f/\partial w_i = 2\eta K(w_i - u_i)$ and $\partial f/\partial t = 1$, `maxiter` $= 500$, `ftol` $= 10^{-10}$, no random restarts. At the returned point the primal violation, equality violation, negativity, and cap violation must each be at most 10^{-7}. An independent LP over the same feasible polytope, minimizing the true gradient $g^\top z$ with $z = (w, t)$, provides the convex first-order (Frank–Wolfe) gap $g^\top \hat{z} - \min_z g^\top z$, which must lie in $[-10^{-7}, 10^{-5}]$, with LP primal and dual status success, certificate stationarity and duality gap at most 10^{-7}. A two-case analytic check (exact zero-residual and nonzero-residual capped-simplex solutions) passed before the pilot. The certificate bounds first-order suboptimality at the returned near-feasible point; it is not a generalization certificate, and the QP itself is a description of the frozen recipe, not a new optimization principle.

B Logical query accounting

Table 19: Logical query accounting per context (protocol facts from `method_spec.json`, section `logical_query_accounting`). These are algorithm-specific logical counts, not measured execution times, and equal paid-label counts are not equal end-to-end compute.

Method / selector	random128	gradient128
Full action queries used by paid methods	9	9
Full no-op query for selector (forward; reused)	0	1
Selector backward passes	0	1
Full audit queries (sealed, unavailable to methods)	64	64
Sparse candidate forwards for calibrated methods	64	64
Sparse forward passes needed by `scalar_full`	0	0

The shared execution runs all 73 full-token rollouts and both selectors' 73 sparse-token rollouts per context once, and the sealed audit labels are unavailable to every method. Per-method deployment costs differ (for example, `scalar_full` needs no sparse deployment forward and `uniform_unqueried` needs no paid label), so the study reports logical counts rather than a single speed number, and makes no speed claim.

C Verification inventory

The four-scene panel and the calibration pilot are covered by independent saved-evidence review by a separate verifier role; the reports live with the evidence packages and are referenced from `claims-ledger.md`. None of the verification is a physical or closed-loop validation, and the readout-bound numbers in Section 5.2 are independently verified (PASS). The task-level anchor's independent environment replay and metric recomputation are also PASS

(1,168/1,168 trajectories exactly reproduced, 19/19 non-finite mask identical, 32/32 images byte-equal, 48/48 determinism reruns, analysis matched to 0.0; Section 6). The exploratory second-checkpoint panel is independently verified on implementation and metrics with zero metric mismatch, with one stale documentation value corrected (Section 7); none of these checks is a planner, closed-loop or physical validation.

D Random-selector calibration results

Table 21 repeats the per-method summary for the secondary random K=128 selector. Both selectors were frozen before execution and neither may be selected after the fact; the gradient selector is primary because it is the deployment setting used in the diagnostic, but reporting both prevents a favorable-selector choice.

Table 20: Verification artifacts and their scope. Saved-array checks certify arithmetic over stored tensors; they do not rerun the network, autograd, renderer, or any physical process.

Artifact	Check	Scope / limit
`checks/receipt.json`	166 checks; 1280 saved H1/H6 costs re-aggregated on CPU; max abs error 2.32415×10^{-7}	saved arrays and metadata only
`checks/portable_recheck.json`	standalone delivery recheck of the same saved-array arithmetic	no vendor, weights, or GPU needed
`check_readout_synthetic.json`	9 metric/aggregate boundary tests (no-oracle fallback, tie order, missing-context refusal)	synthetic CPU only; no real model data
`solver_analytic_check.certified.json`	two exact capped-simplex cases solved to analytic optima	software arithmetic; no efficacy evidence
`verify_saved_calibration.py`	independent recheck of the pilot's saved calibration artifacts and gate	saved artifacts; no model rerun
`run/readout.json` + `run/result.json`	frozen 16-context readout and gate; independent gate recomputation, determinism comparison, saved-artifact recheck	PASS — zero gate mismatch; receipt/report hashes in `claims-ledger.md` (E44–E47)
`task_anchor/analysis.json` + `run/task_outcomes.npz`	task-level anchor: 35,040 vendor simulator steps, image/no-op/repeat checks, independent environment replay and metric recomputation	PASS — 1,168/1,168 trajectories, zero metric mismatch (report/metrics/rollouts hashes in ledger E70–E73); not a planner or closed-loop check
`dino-wm-wall-replication-initial/analysis.json` + `run_dino_wm_panel/costs.npz`	3/3 official checkpoint: strict load, dense parity, K=256 sparse-vs-dense parity, preprocessing discrimination, panel metrics, gradient-selector fix	PASS (implementation, metrics, final snapshot audit); one stale documentation value corrected (1.7014 -> 1.7096); exploratory, not a planner/closed-loop check
`planning_cem/analysis.json` + `run/stage* .json`	G2 planner-level CEM: protocol-before-results, floor gate, fixed-budget CEM, open-loop execution, paired outcomes	PASS (verifier task-10; zero saved-metric mismatch; independent aggregate direction preserved; report `05175f0a...3196cfdf`; snapshot `f24353a6...f24e7a`, 20 payloads); reduced 10x64 budget, open-loop, not official CEM/closed-loop, unseeded single run per cell

Table 21: Per-method mean selected-action gaps for the random K=128 selector (secondary; 16 paired contexts). Generated from the frozen readout.

Method	mean E_{64} gap	mean norm. E_{64} gap	mean U_{73} gap	W/L/T*
Uniform sparse (raw, no anchor)	0.114801	0.182590	0.070880	5/7/4
Uniform sparse + full no-op anchor	0.114801	0.182590	0.074910	5/8/3
Full-integrand absolute cubature	0.101514	0.151014	0.050133	5/4/7
Full-integrand contrast cubature	0.101514	0.151014	0.050133	5/4/7
Sparse-integrand contrast (candidate)	0.114094	0.183977	0.053987	6/5/5
Scalar anchored-kernel (full-cost)	0.269665	0.336037	0.318985	2/12/2
Scalar anchored-kernel (full-sparse residual)†	0.069862	0.110746	0.079498	6/5/5
Best paid pilot (9 full queries)	n/a	n/a	0.068265	—