DEFECT RESPONSES IN VISUAL WORLD MODELS: LONG-HORIZON PROPAGATION, GRADIENT INTERACTION, AND PLANNING OUTCOMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual world-model predictors are trained on local latent targets, whereas planners rank action sequences through composed predictions. Defect response is the separation between two rollouts that share future actions: one starts from the observed next latent, while the other replaces its non-action channels with the model prediction. We ask whether attenuating this response over long horizons improves planning. Defect-Response Fan (DRF) penalizes the terminal response and assigns predictor-parameter credit only through the injected prediction. A full-gradient all-suffix control (FG-AS), inspired by ACPC's same-action pairing, penalizes every suffix response and differentiates through both branches. We freeze every component except the predictor and compare both objectives with Native one-step training on PushT and PointMaze using five paired seeds, a training-inaccessible candidate bank, and matched 1,200-update and 1,800-second total-wall views. Under matched wall time, FG-AS has lower mean-regret point estimates than Native by 4.57% on PointMaze and 7.03% on PushT, despite reaching fewer updates; at 1,200 updates, the direction reverses on both tasks. All paired regret intervals include zero, so the prespecified cross-task, dual-budget confirmation gate fails. Across 40 method–Native contrasts, three response measures attenuate in 23–25 cases, but their Spearman correlations with true-regret improvement are -0.148, 0.052, and 0.032, with cluster-stratified intervals spanning zero. DRF and FG-AS retain 24.6% and 17.7% of Native optimizer throughput. In this setting, the objectives measurably alter response propagation, but response attenuation is not established as a reliable surrogate for planning quality.

1 INTRODUCTION

Visual world-model predictors are optimized on local latent targets, but planners rank candidate action sequences through repeated predictions. Learned dynamics support control through imagined interaction and model-predictive search (Ha & Schmidhuber, 2018; Chua et al., 2018; Schrittwieser et al., 2020); PlaNet and TD-MPC2 exemplify latent planning and scalable model-predictive control (Hafner et al., 2019; Hansen et al., 2024). I-JEPA establishes predictive visual representations, while DINO-WM, V-JEPA 2-AC, and JEPA-WM apply representation-space prediction to physical planning (Assran et al., 2023; Zhou et al., 2025; Assran et al., 2025; Terver et al., 2026). Yet planners act on relative costs after model composition, not average one-step accuracy. Rollout errors can therefore alter candidate rankings in ways that depend on the dynamics, data, and rollout protocol (Janner et al., 2019; Lambert et al., 2022); lowering an internal predictive quantity need not improve a decision.

This objective mismatch is established in model-based control (Lambert et al., 2020; Nair et al., 2020). Latent overshooting extends supervision across rollout steps (Hafner et al., 2019); Online World Modeling and Adversarial World Modeling use simulator-corrected or adversarial trajectories (Parthasarathy et al., 2025); MWM post-trains action-conditioned video rollouts against observed trajectories (Yan et al., 2026). Action-Conditioned Predictive Consistency (ACPC), by contrast, is a frozen-model diagnostic comparing clean and externally perturbed histories under shared recorded actions (An et al., 2026). None establishes that a lower internal response implies better ranking for the predictor-only, fixed-bank setting studied here. We therefore ask whether response attenuation

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

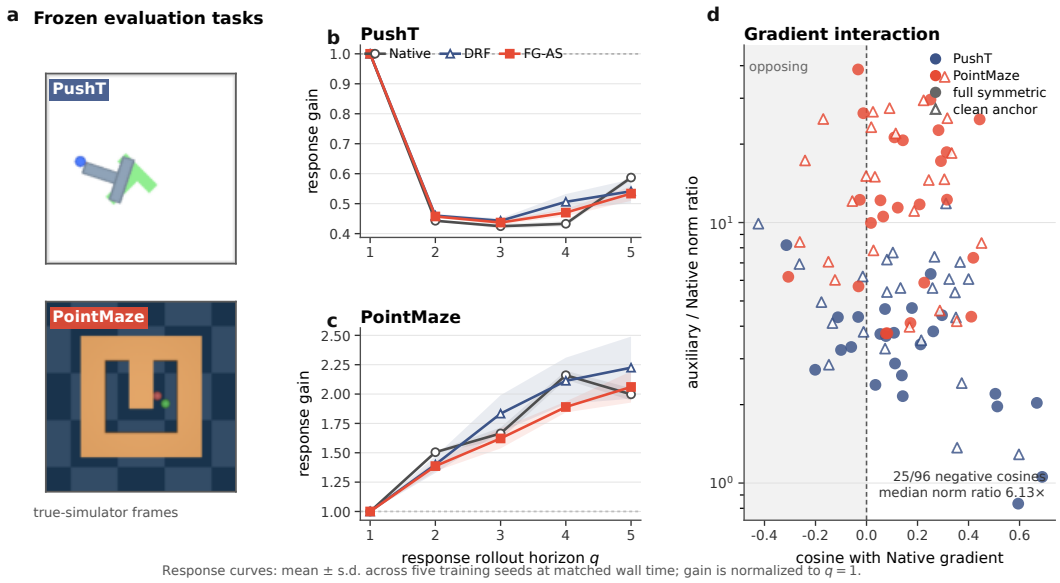

Figure 1: Tasks and frozen internal diagnostics. (a) True-simulator frames from the balanced matched-wall examples in Fig. 3, included only for task context. (b,c) Non-action response gain at the matched-wall checkpoints, normalized to response horizon $q = 1$ and shown as mean $\pm$ s.d. across five training seeds. Clean and defect-injected branches receive identical recorded future actions. (d) Equal-forward-value gradient-routing probe across 96 fixed batch–checkpoint–route measurements; shading marks negative cosine with the Native gradient. These diagnostics use no environment labels and are not planning outcomes.

coincides with lower true-simulator simple regret—the selected candidate's true cost minus the bank minimum—and non-inferior success under matched-update and matched-wall-clock evaluation.

We use the predictor's one-step error as an endogenous perturbation. The live prediction $\hat{x}_{t+1}$ replaces only the non-action channels of the observed next frame; observed-start and defect-start histories then receive the same recorded action suffix. Their non-action separation is the defect response. Defect-Response Fan (DRF) penalizes its terminal energy and assigns predictor-parameter credit only through the live injection, retaining suffix input Jacobians but detaching suffix parameter occurrences. The full-gradient all-suffix control (FG-AS), inspired only by ACPC's shared-action pairing, averages every suffix response and differentiates through both branches. The comparison therefore changes aggregation and parameter credit jointly. Only the predictor is optimized; encoders, decoder, planner, cached target latents, and recorded actions remain fixed. Figure 1 previews the frozen task and mechanism diagnostics; Fig. 2 makes the two training routes explicit.

Hyperparameters and the FG-AS candidate are selected on development data; Native, locked DRF, and FG-AS are then evaluated once on a prespecified confirmation panel. All arms share the released DINO-WM initialization (Zhou et al., 2025), five paired seeds per task, and a training-inaccessible bank of 96 roots with 128 candidates per root. We compare exactly 1,200 updates with a 1,800-second end-to-end budget that includes 100 calibration batches. FG-AS versus Native is primary. In each view, confirmation requires a Bonferroni-adjusted one-sided 97.5% regret lower bound above zero on both tasks and nonnegative mean improvement in every seed. A separate success guardrail requires a one-sided 95% lower bound above -3 percentage points. Intervals use 20,000 hierarchical-bootstrap replicates over seeds and paired roots; FG-AS–DRF inference is tested only if the primary gate passes.

Under matched wall time, FG-AS has lower mean-regret point estimates than Native by 4.57% on PointMaze and 7.03% on PushT, despite fewer updates, and PointMaze success rises by 5.00 points. At 1,200 updates, both regret directions reverse, and every regret interval includes zero; the registered gate fails. Across 40 objective–Native contrasts, response attenuation occurs in 23–25 cases, but its three Spearman correlations with regret improvement are -0.148, 0.052, and 0.032, with cluster-stratified intervals spanning zero. Native–auxiliary gradient cosine is negative in 25 of

96 routing probes, while DRF and FG-AS retain 24.6% and 17.7% of Native optimizer throughput. The intervention is measurable and trainable, but not established as a stable planning surrogate.

Contributions. We operationalize a residual-anchored, same-action response and compare DRF's terminal, injection-only predictor-parameter route with an all-suffix, full-gradient paired-response control. A five-seed, two-task, shared-bank protocol tests rather than assumes transfer to planning under matched-update and matched-wall views. Reporting planning outcomes together with propagation, gradient interaction, and throughput identifies a proxy–outcome boundary rather than claiming uniform improvement.

2 Related Work and Positioning

Latent world models for control. Learned dynamics support model-predictive search, model-generated policy data, and latent imagination (Ha & Schmidhuber, 2018; Chua et al., 2018; Janner et al., 2019; Hafner et al., 2019; Kaiser et al., 2020; Hafner et al., 2020; 2021; Schrittwieser et al., 2020). TD-MPC/TD-MPC2 learn task-oriented latent dynamics, while IRIS, Masked World Models, and DreamerV3 broaden visual control across Atari, robotics, and diverse domains (Hansen et al., 2022; 2024; Micheli et al., 2023; Seo et al., 2023; Hafner et al., 2025). Joint-embedding prediction avoids pixel reconstruction: I-JEPA supplies the representation-learning interface, and DINO-WM, V-JEPA 2-AC, and JEPA-WM apply it to physical planning (Assran et al., 2023; Zhou et al., 2025; Assran et al., 2025; Terver et al., 2026). We instead freeze the released representation and planner and post-train only the predictor.

Long-horizon training. TWISTER and TACO use action-conditioned contrastive objectives, whereas TD-JEPA adds a temporal-difference latent target (Burchi & Timofte, 2025; Zheng et al., 2023; Bagatella et al., 2026). MWM supervises action-conditioned video rollouts with observations; LeWorldModel trains an end-to-end JEPA, and Fast LeWorldModel parallelizes action-prefix prediction (Yan et al., 2026; Maes et al., 2026; Gao & Xu, 2026). PersistWorld applies reward-contrasted post-training to self-generated rollouts, and AdaJEPA adapts latent dynamics online (Bardhan et al., 2026; Wang et al., 2026). Our loss neither changes the representation nor introduces a target network or generated-data loop; it penalizes propagation of the predictor's own one-step defect under fixed recorded actions.

Planner-side changes. Hierarchical Planning with Latent World Models and FF-JEPA introduce subgoals or hierarchical search, while Sparse Imagination reduces planning-time token computation (Zhang et al., 2026; Masip et al., 2026; Chun et al., 2026). Value-guided JEPA, LEAP, and Traj-LeWM modify candidate scoring with value, energy, or path information (Destrade et al., 2025; Pham & Bera, 2026; Huang et al., 2026); HAUWM pre-trains horizon-calibrated uncertainty, and ACID adds inverse-dynamics consistency at decision time (Wan et al., 2026; Seo et al., 2026). DRF and FG-AS leave the candidate population, planning cost, and ranking rule unchanged.

Mismatch and response diagnostics. One-step prediction can be an imperfect control proxy (Lambert et al., 2020; Nair et al., 2020; Lambert et al., 2022); value equivalence further shows that useful decision structure need not reproduce every transition (Grimm et al., 2020). Online World Modeling, Adversarial World Modeling, and invariant visual planning use simulator-corrected trajectories, adversarial perturbations, or bisimulation objectives (Parthasarathy et al., 2025; Toso et al., 2026), while gradient surgery and phasic auxiliary optimization address generic multiobjective interference (Yu et al., 2020; Cobbe et al., 2021). Closest to our construction, ACPC diagnoses prediction-cost change and selection regret under shared-action clean/perturbed rollouts without updating weights (An et al., 2026). We place an endogenous response inside predictor post-training and test it against true-simulator candidate-bank outcomes. Appendix Table 4 summarizes representative fixed-dimension contrasts; because DRF and FG-AS change aggregation and predictor-parameter credit together, we treat them as bundled objectives rather than a factorial routing ablation.

3 DEFECT-RESPONSE TRAINING

3.1 PREDICTOR-ONLY ADAPTATION

We study a released DINO-WM predictor without changing its representation, action interface, or planner (Zhou et al., 2025). At model time t, one augmented latent frame is $z_t = [x_t, a_t] \in \mathbb{R}^{P\times(D_x+D_a)}$, where $x_t \in \mathbb{R}^{P\times D_x}$ contains the visual-patch and proprioceptive channels and $a_t \in \mathbb{R}^{P\times D_a}$ is the recorded action after encoding and patchwise tiling. Let π_x remove the action channels, let h be the history length, and write $C_t = (z_{t-h+1}, \dots, z_t)$ for the complete history state. The output of $f_\theta(C_t)$ is reshaped into h frame blocks; $[f_\theta(C_t)]_m$ denotes block m, not one flattened patch token. For any single prediction head, $\mathrm{MSE}(u)$ denotes the mean of u^2 over its batch, patch, and channel axes. When visual and proprioceptive heads are separate, the implementation sums their scalar MSE terms; the equations suppress this head index for readability.

The released one-step objective, which we call Native, supervises every causal output block. For a clip $(z_0, \dots, z_{T-1})$, every length-h context whose shifted targets remain in the clip contributes

$$\mathcal{L}_{\mathrm{nat}}(\theta) = \frac{1}{(T-h)h} \sum_{s=0}^{T-h-1} \sum_{m=0}^{h-1} \mathrm{MSE}(\pi_x[f_\theta(z_s, \dots, z_{s+h-1})]_m - x_{s+m+1}) \,. \tag{1}$$

Our experiments use $T = 8$ and $h = 3$, hence five contexts per cached clip. Response rollouts below append only the final output block. In all objectives, recorded actions are supplied as inputs rather than used as prediction targets. Only θ is trainable; the DINOv2 encoder, action encoder, proprioceptive encoder, and decoder remain frozen.

3.2 AN ENDOGENOUS, SAME-ACTION RESPONSE

From one observed history, we imagine two futures. The observed-start branch retains the recorded next latent. The injected branch replaces only its non-action channels with the predictor's own estimate. Both branches then receive exactly the same recorded actions.

Concretely, the live predictor forms

$$\widehat{x}_{t+1} = \pi_x[f_\theta(C_t)]_{-1}, \qquad \widetilde{z}_{t+1} = [\widehat{x}_{t+1}, a_{t+1}]. \tag{2}$$

Let $S(C, z)$ drop the oldest frame of C and append z. The two starts are defined together as

$$C^0_{t+1} = S(C_t, z_{t+1}), \qquad C^1_{t+1} = S(C_t, \widetilde{z}_{t+1}). \tag{3}$$

For $q \geq 2$, both branches are advanced with the same recorded action channel:

$$C^b_{t+q} = G_{\theta, a_{t+q}}(C^b_{t+q-1}), \qquad b \in \{0, 1\}, \tag{4}$$
$$G_{\theta,a}(C) := S(C,\, [\pi_x[f_\theta(C)]_{-1}, a]) \,.$$

The finite response is defined for every $q \geq 1$:

$$r_{t\to t+q}(\theta) = \pi_x\big([C^1_{t+q}]_{-1} - [C^0_{t+q}]_{-1}\big) \,. \tag{5}$$

For $q = 1$, Eq. (5) uses the starts in Eq. (3) and equals the one-step residual $\widehat{x}_{t+1} - x_{t+1}$; no suffix call is needed. For $q > 1$, no observed $x_{t+2:t+q}$ is a loss target. The quantity measures separation between two model rollouts, not prediction error against the observed suffix. Auxiliary rollouts disable predictor dropout, matching the deterministic deployed planner; the Native loss retains the released training mode.

The response name follows a forward-value identity on the lifted history state. At fixed θ, with dropout disabled and a common recorded-action suffix, let $Y^{(\theta)}_{j\to e}$ be the projected non-action endpoint obtained by predicting frame $j+1$ from observed history C_j and then freely rolling to e. Set $Y^{(\theta)}_{e\to e} = x_e$. Then

$$r^{(\theta)}_{j\to e} = Y^{(\theta)}_{j\to e} - Y^{(\theta)}_{j+1\to e}, \qquad Y^{(\theta)}_{k\to e} - x_e = \sum_{j=k}^{e-1} r^{(\theta)}_{j\to e}. \tag{6}$$

This is a telescoping finite-difference identity by construction. Only response vectors within a head telescope. Squared energies, sums of head-wise scalar losses, and the selective gradients below do not. The identity asserts neither independent defect contributions nor a monotone relation to planner ranking. DRF samples individual terminal-edge energies; it does not minimize the full sum in Eq. (6).

3.3 DRF: TERMINAL RESPONSE WITH INJECTION-ONLY CREDIT

At optimizer step n, for every length $\ell \in \{2,3,4,5\}$ we draw one injection index uniformly from $\{h-1,\dots,T-\ell-1\}$. The resulting four-edge set is $\mathcal{E}_n = \{(t_\ell, t_\ell+\ell)\}_\ell$; one sampled index is shared by all examples in the batch. Defect-Response Fan (DRF) averages terminal response energy:

$$\mathcal{L}_{\mathrm{DRF}}(\theta) = \frac{1}{|\mathcal{E}_n|} \sum_{(t,e)\in\mathcal{E}_n} \mathrm{MSE}\big(r^{\mathrm{sel}}_{t\to e}(\theta)\big)\,. \tag{7}$$

The superscript "sel" changes autodiff, not the forward response. Let $R_{t,e}(u;\phi)$ denote the paired endpoint response when the injected non-action frame is u and every suffix call uses predictor state ϕ. DRF evaluates

$$r^{\mathrm{sel}}_{t\to e}(\theta) := R_{t,e}(\widehat{x}_{t+1}(\theta); \mathrm{sg}(\theta))\,, \tag{8}$$

where $\mathrm{sg}(\theta)$ denotes a value-identical snapshot of predictor parameters and buffers that is constant to automatic differentiation. Forward values are unchanged, and derivatives with respect to suffix inputs remain live, but suffix parameter occurrences receive no auxiliary gradient. Consequently, the expression below is an implemented semigradient rather than the total derivative of a scalar function with all occurrences of θ live:

$$\widetilde{\nabla}_\theta \mathcal{L}_{\mathrm{DRF}} = \frac{1}{|\mathcal{E}_n|} \sum_{(t,e)\in\mathcal{E}_n} J_\theta \widehat{x}_{t+1}^\top \, \nabla_u \,\mathrm{MSE}(R_{t,e}(u;\mathrm{sg}(\theta)))\big|_{u=\widehat{x}_{t+1}}\,. \tag{9}$$

Thus terminal credit traverses the suffix input Jacobians back to the live injection. This is neither a truncated forward rollout nor ordinary full backpropagation through time.

3.4 FULL-GRADIENT ALL-SUFFIX PAIRED-RESPONSE CONTROL

ACPC compares clean and externally perturbed histories under shared actions as a frozen-model diagnostic (An et al., 2026). We use *ACPC-inspired* only for this paired, shared-action template. Our control replaces the external view perturbation with an endogenous one-step replacement, turns a diagnostic into predictor training, and minimizes paired response energy; it is not an ACPC reproduction or ablation.

The resulting full-gradient all-suffix (FG-AS) control uses the same sampled edges and actions as DRF. For an edge of length $\ell = e - t$, it averages every shared-action suffix response:

$$\mathcal{L}_{\text{FG-AS}}(\theta) = \frac{1}{|\mathcal{E}_n|} \sum_{(t,e)\in\mathcal{E}_n} \frac{1}{e-t-1} \sum_{q=2}^{e-t} \mathrm{MSE}(r_{t\to t+q}(\theta))\,. \tag{10}$$

Suffix states are uniform within an edge, and the four edges retain equal outer weight; the $q=1$ injected start is not itself a loss term. Gradients pass through the live injection and every predictor occurrence in both observed-start and injected suffixes.

Both objectives compare two model rollouts rather than a rollout with observed future latents. Ordinary overshooting or full-BPTT rollout loss instead uses observed future latents as targets and differentiates through every rollout occurrence. FG-AS has a full parameter gradient but remains a paired-response objective. The confirmation comparison also changes two coupled choices—terminal versus all-suffix aggregation and selective versus full parameter credit—so it compares bundled objectives rather than identifying a causal routing effect.

Figure 2 makes these two coupled choices explicit while preserving the common forward construction.

One edge of length ℓ requires one injection call and two calls for each of its $\ell-1$ suffix states, or $2\ell-1$ forward predictor evaluations. The per-batch auxiliary total is therefore $\sum_{\ell\in\{2,3,4,5\}}(2\ell-1) = 24$ for either route. Equal forward-call count does not imply equal backward cost, motivating separate matched-update and matched-wall-clock views.

3.5 GRADIENT CALIBRATION AND UPDATE

For $A \in \{\mathrm{DRF}, \text{FG-AS}\}$, training uses

$$\mathcal{L}^{\mathrm{train}}_A(\theta) = \mathcal{L}_{\mathrm{nat}}(\theta) + \lambda_A \mathcal{L}_A(\theta). \tag{11}$$

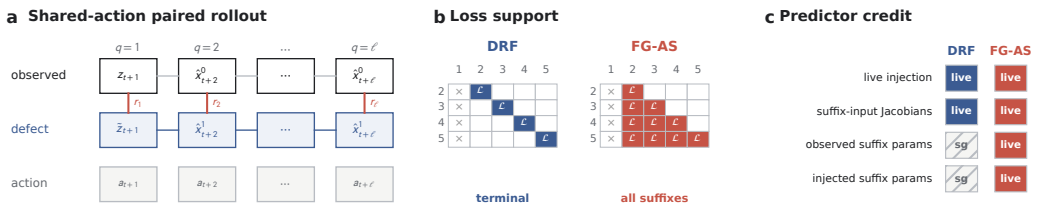

Figure 2: Objective support and predictor-parameter credit. (a) Observed and defect starts share actions; r_q is their non-action separation. (b) DRF penalizes only $q = \ell$, whereas FG-AS averages $q = 2, \ldots, \ell$; $q = 1$ is excluded. (c) Both keep the injection and suffix-input Jacobians live; DRF stop-gradients suffix parameter occurrences, whereas FG-AS differentiates both branches. Forward trajectories and call counts match; aggregation and credit differ jointly.

In the frozen confirmation runs, every arm explicitly uses $K = 100$ no-update calibration batches. Let $g_{\text{nat}}^{(k)}$ be the predictor-parameter gradient of the Native loss on batch k. Let $g_A^{(k)}$ be the corresponding unscaled auxiliary direction, using $\widetilde{\nabla}$ for DRF and the ordinary gradient for FG-AS. Before global clipping, we compute

$$\bar{g}_{\text{nat}} = \frac{1}{K}\sum_{k=1}^{K} \|g_{\text{nat}}^{(k)}\|_2, \qquad \bar{g}_A = \frac{1}{K}\sum_{k=1}^{K} \|g_A^{(k)}\|_2, \qquad \lambda_A = 0.25\,\frac{\bar{g}_{\text{nat}}}{\bar{g}_A}, \tag{12}$$

with finite $\bar{g}_A > 0$ required. This calibrates the mean pre-clipping auxiliary norm, not the norm of a mean gradient or the realized fraction of an AdamW update. Native performs 100 corresponding no-update Native-gradient batches. All arms then reset data order and random state before optimization.

We update predictor parameters only with AdamW using learning rate 5×10^{-6}, default $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$, zero weight decay, and batch size 8. After accumulating Native and auxiliary gradients, we clip the predictor-wide global ℓ_2 norm to 1. Calibration time counts toward the equal-total-wall budget. Within each task and training seed, all arms start from the same released predictor and use the same cached clips, data-order stream, and edge-sampling stream; arm order is counterbalanced across the five seeds.

The intervention changes neither the planner nor its cost. It changes only predictor training. Our evaluation therefore measures internal response and held-out true-simulator planning outcomes separately.

4 Experimental Setup

We compare Native, DRF, and FG-AS on PushT and PointMaze through the released DINO-WM interface (Zhou et al., 2025). FG-AS is the single ACPC-inspired objective selected from development evidence; all confirmation settings were fixed before the candidate bank was generated. Table 1 summarizes the design, while Appendix A gives construction details, timing boundaries, and hashes.

Table 1: Frozen confirmation protocol. Both resource views, all roots, and all paired seeds are fixed.

Axis	Design
Study	PushT and PointMaze; three arms; five paired seeds
Trainable state	Predictor only; frozen encoders, decoder, planner, and task checkpoint
Training cache	4,096 latent clips/task; shared initialization, order, and random-state reset
Optimization	AdamW, 5×10^{-6}, batch 8; 100 calibration batches; clip 1
Resource views	1,200 updates; $\geq$1,800 s total wall (first completed update)
Blind endpoint	96 roots $\times$ 128 candidates/task; same bank for all 30 checkpoints
Inference	20,000 hierarchical paired bootstraps; Bonferroni over tasks; both views required

Auxiliary weights are fixed by Eq. (12); Native executes the corresponding Native-gradient pass. Total wall time includes loading, integrity checks, calibration, optimization, logging, and periodic checkpoint I/O. Thus matched updates control update count but not computation, whereas matched wall time permits different update counts.

Blind fixed-bank planning endpoint. The bank is built with the common initial predictor and is inaccessible to training. Each candidate contains 25 normalized primitive actions packed into five model steps. Checkpoint s selects $\widehat{j}_{s,r} = \arg\min_j \widehat{c}_{s,r,j}$ at root r (first-index ties); the simulator supplies terminal cost $c_{r,j}$ and success. We report

$$R_{s,r} = c_{r,\widehat{j}_{s,r}} - \min_j c_{r,j}, \qquad S_{s,r} = \mathbf{1}\{\widehat{j}_{s,r} \text{ is successful}\}, \tag{13}$$

where lower regret and higher success are better. PushT success requires full-state error below 20 and wrapped angular error below $\pi/9$; PointMaze uses XY distance below 0.5. All 96 PushT roots and 75 of 96 PointMaze roots contain a successful candidate. This endpoint isolates ranking, not online CEM, receding-horizon control, or planning latency.

Paired inference and decision rule. For FG-AS (F) against $C \in \{\text{Native}, \text{DRF}\}$, $\Delta_R = R_C - R_F$ and $\Delta_S = S_F - S_C$; positive values favor FG-AS. Each replicate resamples five seeds and then 96 roots within a selected seed row, applying identical indices to both methods. The seed is the replication level, not the shared 5×96 cells. FG-AS versus Native is primary. Both views require a Bonferroni-adjusted 97.5% lower bound for Δ_R above zero on both tasks, nonnegative mean improvement in every seed, and a one-sided 95% lower bound for Δ_S above -0.03. FG-AS versus DRF is inferentially eligible only after this gate passes.

Mechanism, efficiency, and external strata. Let q denote response rollout horizon, distinct from the history length h. An overlap-free 64-window cache per task measures non-action RMS R_q, gain $G_q = R_q/R_{q=1}$, and the reported discrete gain AUC $\bar{G}_{2:5} = \frac{1}{4}\sum_{q=2}^{5} G_q$; this is the uniform average over response horizons 2–5, not a continuous integral. Attenuation is Native minus method. Separate eight-batch probes compare Native gradients with two equal-forward-value routes; they are diagnostics, not trained arms. Response–planning association keeps all 40 contrasts and resamples seed clusters within task–budget strata with method rows paired (20,000 bootstrap and permutation replicates; Appendix B.2). Efficiency reports updates, throughput, memory, and predictor calls. Protocol-separated AWM and JEPA-WM checks enter no pooled comparison (Parthasarathy et al., 2025; Terver et al., 2026).

5 Results

5.1 Planning effects reverse across budget views

Table 2 reports every registered task–budget cell; Fig. 3a visualizes the paired FG-AS effects after dividing each task–budget estimate and interval by its Native regret. Under matched wall time, FG-AS has lower mean-regret point estimates than Native by 0.151 on PointMaze (4.57%) and 0.921 on PushT (7.03%), with positive seedwise effects in four and five seeds; PointMaze success rises by 5.00 percentage points (paired 95% interval $[1.46, 8.54]$), with positive effects in all five seeds. At 1,200 updates, FG-AS has higher mean-regret point estimates by 0.103 and 0.915, with one positive seedwise effect in each task. DRF shows the same directional split. Every FG-AS regret interval includes zero, so the prespecified dual-budget, cross-task gate is not met: the wall-time pattern is favorable but not confirmed.

Table 2: Frozen confirmation panel. R is true-simulator simple regret; S is success (%). $\Delta R = R_{\text{Native}} - R_{\text{FG-AS}}$ with paired 95% intervals; k counts seeds with positive mean ΔR. Bold marks the best row value, not significance.

		Native		DRF		FG-AS			
Task	Budget	R	S	R	S	R	S	ΔR [95% CI]	$k/5$
PointMaze	1,200 updates	**3.124**	66.46	3.244	**67.50**	3.228	**67.50**	-0.103 $[-0.256, 0.046]$	1
PointMaze	1,800 s	3.309	62.92	3.209	66.25	**3.158**	**67.92**	0.151 $[-0.026, 0.323]$	4
PushT	1,200 updates	**11.570**	**92.50**	12.863	**92.50**	12.485	92.08	-0.915 $[-2.193, 0.316]$	1
PushT	1,800 s	13.088	92.50	12.920	**93.12**	**12.168**	92.92	0.921 $[-0.125, 1.952]$	5

Paired success effects. In table order, $\Delta S = S_{\text{FG-AS}} - S_{\text{Native}}$ [95% CI] is $+1.04$ $[-2.71, 5.00]$, $+5.00$ $[1.46, 8.54]$, -0.42 $[-2.08, 1.25]$, and $+0.42$ $[-1.46, 2.29]$ percentage points; positive seed counts are $3/5$, $5/5$, $0/5$, and $2/5$.

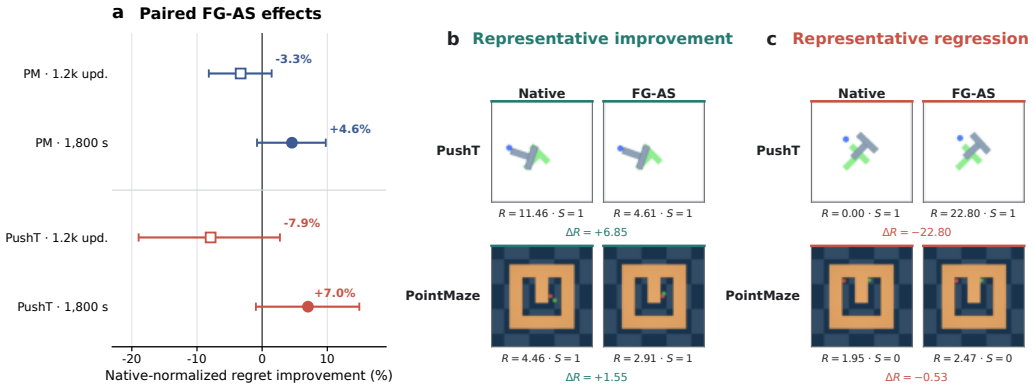

Figure 3: Planning effects and true-simulator outcomes. (a) FG-AS–Native regret differences, normalized by Native regret for display; bars are paired 95% intervals, with squares for 1,200 updates and circles for 1,800 s. (b,c) Frozen matched-wall Native/FG-AS pairs selected without images or success labels: within each task and sign, the median root-level seed-mean effect and then a direction-consistent seed. Pairs share root and bank; both directions are shown; frames are simulator states.

5.2 Internal attenuation does not predict planning reliably

Across all 40 method–Native contrasts, attenuation occurs in 23 cases for gain AUC, 25 for horizon-5 gain, and 23 for horizon-5 response RMS; regret improves in 18, and only 10, 12, and 10 contrasts show both. The corresponding Spearman correlations are -0.148, 0.052, and 0.032. Cluster-stratified intervals span zero for all three, with paired-cluster permutation $p \geq 0.283$ (Fig. 4). Task–budget strata vary in sign (Appendix B.2), so the pooled pattern is not monotonic.

The equal-forward-value routing probes have a mean auxiliary-to-Native gradient-norm ratio of 9.773 (median 6.130) across 96 batch–checkpoint–route measurements; 25 cosines are negative. These diagnostics establish scale mismatch and occasional local opposition, not a cause of planning outcomes.

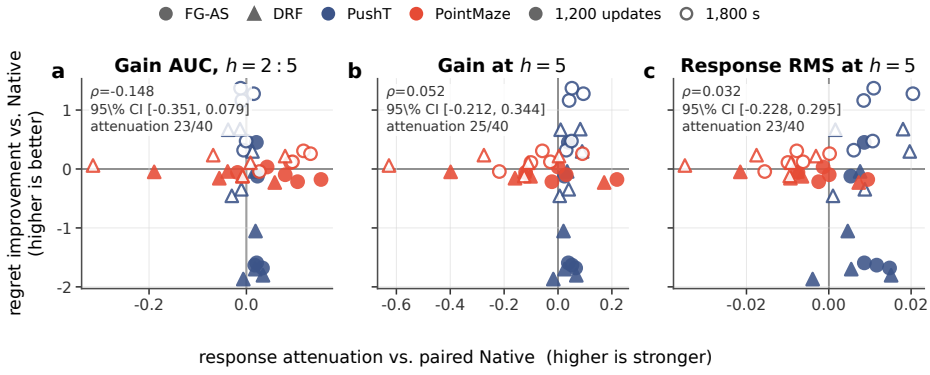

Figure 4: Response attenuation versus true-regret improvement for all 40 method–Native contrasts. Method is encoded by shape, task by color, and budget by fill. Each panel reports Spearman ρ and a cluster-stratified 95% interval. Both favorable and unfavorable planning quadrants contain attenuated responses.

5.3 Favorable wall-time estimates incur substantial training cost

Within 1,800 seconds, FG-AS reaches 1,289.3 updates (16.6% of Native) and retains 17.7% throughput while adding 1.20 GiB peak memory; DRF retains 24.6%. Both use 24 auxiliary calls per batch. Thus the favorable wall-time estimates do not reflect equivalent optimization work (Appendix Table 6).

A development control gives DRF no cross-task advantage over matched overshooting. AWM is favorable only on PushT regret, while JEPA-WM succeeds on 63/96 PushT and 84/96 PointMaze episodes under its own protocol (Parthasarathy et al., 2025; Terver et al., 2026). Appendix B gives protocol-separated values and provenance; none is pooled with Table 2.

6 Discussion

The response is measurable and trainable, but its planning effect is budget-dependent. Both objectives reverse regret direction between matched updates and matched wall time, where FG-AS reaches only 16.6% of Native's update count; the resource views are not interchangeable, and attenuation is not a monotonic planning surrogate. DRF and FG-AS also couple response aggregation with parameter credit, so their contrast cannot identify routing as the cause of a planning difference. Equal-forward-value probes expose local gradient changes, not outcome causation. Factorial routing–aggregation tests, planner-aware targets, and cheaper response approximations remain next steps.

7 Limitations

The panel covers two tasks and one architecture; its fixed bank isolates ranking rather than online planning, and five seeds do not establish architecture-level generality. A Wall transfer stopped before training. The mechanism analysis remains descriptive, while AWM and JEPA-WM are protocol-separated.

8 Conclusion

We introduced a residual-anchored same-action response and compared two predictor-training routes. FG-AS has favorable wall-time regret point estimates, but the matched-update direction reverses, the confirmation gate fails, and response attenuation does not track regret reliably. Changing long-horizon response is therefore not, by itself, evidence that a visual world model will plan better; future objectives should establish that link under both statistical and computational controls.

Reproducibility statement

Sections 3 and 4 specify the objectives, gradient routes, optimization, evaluation endpoint, and statistical decision rule. Appendix A gives cache provenance, bank construction, timing boundaries, and source hashes. The supplementary material contains the immutable experiment specifications, source manifests, plotting code, checkpoint-bound evaluation records, and unit tests used to verify the selective-gradient and full-gradient implementations.

Ethics statement

This work uses simulated control tasks and released model checkpoints. It involves no human participants, personal data, or deployment. The principal research-integrity risk is overinterpreting a descriptive mechanism signal; we address it by separating development from confirmation, reporting the failed confirmation gate, and keeping protocol-incompatible external results out of pooled comparisons.

AI use statement

Generative AI tools were used for three purposes: retrieving and discovering relevant literature; drafting and organizing sections of the paper; and aiding or polishing the writing, including title and text editing and reference formatting. They were not used to generate experimental observations or synthetic datasets, or to formulate or prove mathematical claims. The authors reviewed the AI-assisted text, checked numerical claims against hash-bound experiment records and citations against official sources, inspected every figure and table, and retain final responsibility for the paper's text, claims, and artifacts.

REFERENCES

Guo An, Zijing Wu, Honghua Dong, Yuhao Yan, Zixuan Gui, Haochong Chen, Shanzhao Ruan, Xiang Wang, Yurong Ling, and Qi Tian. Diagnosing JEPA world models with action-conditioned predictive consistency. *arXiv preprint arXiv:2608.12939*, 2026. URL https://arxiv.org/abs/2608.12939.

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

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

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3d158f054ff0cb83397367234899db07-Paper-Conference.pdf.

Jai Bardhan, Patrik Drozdik, Josef Sivic, and Vladimir Petrik. Persistent robot world models: Stabilizing multi-step rollouts via reinforcement learning. *arXiv preprint arXiv:2603.25685*, 2026. URL https://arxiv.org/abs/2603.25685. Accepted at ECCV 2026.

Maxime Burchi and Radu Timofte. Learning transformer-based world models with contrastive predictive coding. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://iclr.cc/virtual/2025/poster/29267.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Junha Chun, Youngjoon Jeong, and Taesup Kim. Sparse imagination for efficient visual world model planning. In *International Conference on Learning Representations*, volume 2026, pp. 102160–102188, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a750d52284ff70c6d6bab8072c392d74-Paper-Conference.pdf.

Karl W. Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2020–2027. PMLR, 2021. URL https://proceedings.mlr.press/v139/cobbe21a.html.

Matthieu Destrade, Oumayma Bounou, Quentin Le Lidec, Jean Ponce, and Yann LeCun. Value-guided action planning with JEPA world models. *arXiv preprint arXiv:2601.00844*, 2025. URL https://arxiv.org/abs/2601.00844. Presented at the World Modeling Workshop 2026.

Yuntian Gao and Xiangyu Xu. Fast LeWorldModel. *arXiv preprint arXiv:2606.26217*, 2026. URL https://arxiv.org/abs/2606.26217.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552. Curran Associates,

Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. doi: 10.5281/zenodo.1207631. URL https://arxiv.org/abs/1803.10122.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *The Eighth International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *The Ninth International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2. URL https://doi.org/10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://iclr.cc/virtual/2024/poster/18722. Spotlight.

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

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

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłoś, Błażej Osiński, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for Atari. In *The Eighth International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

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

Nathan Lambert, Kristofer Pister, and Roberto Calandra. Investigating compounding prediction errors in learned dynamics models. *arXiv preprint arXiv:2203.09637*, 2022. URL https://arxiv.org/abs/2203.09637.

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

Sergi Masip, Jonathan Swinnen, Yutong Hu, Renaud Detry, and Tinne Tuytelaars. FF-JEPA: Long-horizon planning in world models with latent planners. *arXiv preprint arXiv:2606.09311*, 2026. URL https://arxiv.org/abs/2606.09311.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb. Oral.

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

Arjun Parthasarathy, Nimit Kalra, Rohun Agrawal, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning. *arXiv preprint arXiv:2512.09929*, 2025. URL https://arxiv.org/abs/2512.09929.

Phu Pham and Aniket Bera. Latent energy action planning with world models. *arXiv preprint arXiv:2609.03294*, 2026. URL https://arxiv.org/abs/2609.03294.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4. URL https://doi.org/10.1038/s41586-020-03051-4.

Gawon Seo, Dongwon Kim, and Suha Kwak. ACID: Action consistency via inverse dynamics for planning with world models. *arXiv preprint arXiv:2607.02403*, 2026. URL https://arxiv.org/abs/2607.02403.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1332–1344. PMLR, 2023. URL https://proceedings.mlr.press/v205/seo23a.html.

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. URL https://openreview.net/forum?id=cHZn5Gdh8e. Accepted; OpenReview forum cHZn5Gdh8e.

Leonardo F. Toso, Davit Shadunts, Yunyang Lu, Nihal Sharma, Donglin Zhan, Nam H. Nguyen, and James Anderson. Learning invariant visual representations for planning with joint-embedding predictive world models. *arXiv preprint arXiv:2602.18639*, 2026. URL https://arxiv.org/abs/2602.18639.

Shenghua Wan, Le Gan, and De-Chuan Zhan. Learning to be uncertain: Pre-training world models with horizon-calibrated uncertainty. In *International Conference on Learning Representations*, volume 2026, pp. 106107–106127, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ad222e6e38e26e3f00d85829b9e02f43-Paper-Conference.pdf.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: An adaptive latent world model. *arXiv preprint arXiv:2606.32026*, 2026. URL https://arxiv.org/abs/2606.32026.

Han Yan, Zishang Xiang, Zeyu Zhang, and Hao Tang. MWM: Mobile world models for action-conditioned consistent prediction. *arXiv preprint arXiv:2603.07799*, 2026. URL https://arxiv.org/abs/2603.07799.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

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

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 48203–48225. Curran Associates, Inc., 2023. doi: 10.52202/075280-2092. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/96d00450ed65531ffe2996daed487536-Paper-Conference.pdf.

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

Appendix

A Protocol and Reproducibility Details

A.1 Primary confirmation design

Table 3: Frozen design of the primary confirmation panel.

Axis	Registered setting
Scope	PushT and PointMaze; Native, DRF, and FG-AS; five paired training seeds
Trainable state	Predictor only; visual, action, and proprioceptive encoders plus decoder frozen
Cache	4,096 latent clips per task; eight frames; history length 3; frameskip 5
Optimizer	AdamW; learning rate 5×10^{-6}; batch 8; weight decay 0; global clip 1
Calibration	100 no-update batches; fixed mean pre-clipping auxiliary-gradient ratio 0.25
Budget views	Exactly 1,200 optimizer updates; first completed update at or beyond 1,800 s total wall time
Planning endpoint	96 roots $\times$ 128 candidates per task; same bank for all 30 task checkpoints
Inference	20,000 hierarchical paired-bootstrap replicates; Bonferroni across two tasks

PushT draws one legal eight-frame window from each of 4,096 eligible training episodes. PointMaze draws 4,096 unique episode–start windows without replacement from 1,800 episodes; the selected windows span 1,624 episodes. Both caches store `float16` tensors of shape $8\times196\times404$. The matched-update checkpoint follows 800 calibration examples and 9,600 optimization examples. The wall-clock timer includes data and checkpoint loading, integrity checks, calibration, optimization, logging, periodic checkpoint I/O, and snapshot publication. Final serialization occurs after the stopping decision.

PushT candidates comprise 48 demonstration perturbations at noise scale 0.25, 48 at 0.50, and 32 standard-normal sequences; the exact demonstration is excluded. PointMaze includes 64 standard-normal sequences and the 64 proposals from the last of three CEM rounds produced by the common initial predictor. Candidate actions are packed from 25 primitive steps into five model steps. The training process cannot open the bank. Evaluation records are bound to bank and checkpoint hashes before analysis.

A.2 Primary result provenance

The absolute values in Table 2 are read from the two task-level confirmation analyses bound by the original run's `result.sha256`; paired gates come from the frozen decision file. A deterministic diagnostic view supplies the efficiency aggregates. The extraction cross-checks the two views and asserts $R_{\text{Native}} - R_{\text{FG-AS}} = \Delta_R$ and $S_{\text{FG-AS}} - S_{\text{Native}} = \Delta_S$ in every task–budget cell. Secondary narrative audits are not analysis inputs. Source SHA-256 values are:

```
PushT: f24a4d2d47f68cae0e360ff107971d5d
       d9fd66f53aac0d647452afcbe46c4173
PointMaze:  a568313997bc3e9aa0c96952eb8288c1
            66178112804f537a0bfd715c71716543
decision:  af00c1221713f27bcf99dca436bd969c
           eec35496f115b8a3fb15fcfd0875ae95
```

Decision-gate replay. From the root of the anonymous supplementary audit artifact, `bash CHECK_ALL.sh` verifies the package hashes, reconstructs the four registered result cells and frozen gate, checks the 40 mechanism contrasts, 96 gradient measurements, external-protocol records, and terminal-case selection, and runs 33 implementation tests. Large upstream checkpoints, datasets, simulator assets, latent caches, and candidate banks are identified by immutable records but are not redistributed; end-to-end retraining requires those external resources. This boundary does not affect the hash-bound audit of the submitted values.

Table 4: Representative mechanism positioning against cited methods. "Labels" denotes simulator-derived supervision; "planner" denotes a changed candidate-generation or scoring procedure. This is a mechanism comparison, not an empirical leaderboard.

Work	Predictor updated	Simulator labels	Planner changed	Paired source	Training credit
ACPC (An et al., 2026)	no	no	no	exogenous view	diagnostic only
AWM (Parthasarathy et al., 2025)	yes	yes	no	adversarial state–action	full next-state loss
TD-MPC2 (Hansen et al., 2024)	yes	reward/value	yes	latent trajectories	end-to-end control
Sparse Imagination (Chun et al., 2026)	yes	task reward	yes	sparse imagined tokens	end-to-end control
DRF / FG-AS (ours)	yes	no	no	endogenous defect	selective / full pair

A.3 Fixed-dimension closest-work comparison

B Additional Diagnostics and Baselines

B.1 Gradient interaction

Across 96 fixed batch–checkpoint–route measurements, the unscaled auxiliary-to-Native gradient-norm ratio has mean 9.773, median 6.130, and range $[0.834, 38.692]$; the cosine is negative in 25 cases. The count is two tasks $\times$ three predictor states (initial, matched-update, matched-wall) $\times$ eight fixed batches $\times$ two equal-forward-value all-suffix routes (full symmetric credit and a stop-gradient clean anchor). These are routing probes, not three trained arms. The diagnostic is separate from calibration: calibration estimates one fixed scalar from 100 other batches, whereas these probes record realized batch-level scale and direction.

B.2 Mechanism association by task and budget

Table 5: Descriptive mechanism–planning heterogeneity. Each stratum contains ten method–Native contrasts (two methods and five paired seeds). Entries are Spearman ρ; b counts contrasts with both positive attenuation and positive regret improvement.

Task	Budget	Regret +	Gain AUC ρ / b	Gain h=5 ρ / b	RMS h=5 ρ / b
PointMaze	1,200 updates	1/10	-0.527 / 1	-0.455 / 1	-0.455 / 0
PointMaze	1,800 s	8/10	0.612 / 6	0.576 / 2	0.515 / 1
PushT	1,200 updates	1/10	0.006 / 1	-0.103 / 1	-0.103 / 1
PushT	1,800 s	8/10	0.115 / 2	0.527 / 8	0.455 / 8

The pooled correlations in Fig. 4 are therefore not summaries of a homogeneous relation: the stratum-level signs and attenuation frequencies vary. The cluster-aware bootstrap resamples training-seed clusters within each task–budget stratum and carries both objective rows together; it does not treat the 40 rows as independent runs.

B.3 Training efficiency

Table 6: Efficiency averaged over tasks and seeds. Updates are reached in the 1,800-s total-wall view; throughput excludes setup and calibration.

Objective	Updates at 1,800 s	Updates/s	Peak GiB	Aux. calls/batch
Native	7,747.9	4.385	18.161	0
DRF	1,829.3	1.077	18.161	24
FG-AS	1,289.3	0.775	19.356	24

B.4 Protocol-separated external checkpoints

Table 7: External evidence strata. AWM is evaluated on a common fixed bank; JEPA-WM follows its released online protocol. The rows are not a cross-method leaderboard.

Protocol	Task	Model outcome	Reference outcome	Scope
AWM common bank	PushT	$R = 10.582$, $S = 90.63$	Native $R = 13.652$, $S = 92.71$	$\Delta R = 3.071$ $[0.716, 5.403]$
AWM common bank	PointMaze	$R = 3.299$, $S = 54.17$	Native $R = 3.258$, $S = 64.58$	$\Delta R = -0.041$ $[-0.437, 0.355]$
JEPA-WM official	PushT	63/96 success (65.63%)	—	Wilson 95% CI $[55.69, 74.36]$
JEPA-WM official	PointMaze	84/96 success (87.50%)	—	Wilson 95% CI $[79.41, 92.70]$

The AWM uncertainty is root-level because only one released checkpoint is available per task. Its official repository is pinned at commit `c97ac5f6113347e5ecc93ffb01220c05d686748b`; converted PushT and PointMaze checkpoint SHA-256 values begin `7735e1ab` and `f7f7521b`. The JEPA-WM run pins `https://github.com/facebookresearch/jepa-wms` at commit `13cf1d9c7e476f53c17714d2e0f1dc239a883ce0`; its PushT and PointMaze checkpoint hashes begin `9beca3ea` and `a01d99c4`. Full hashes, frozen specifications, per-root AWM scores, JEPA-WM evaluation records, and decision files are included in the supplement. JEPA-WM uses different goals, candidate generation, and online control, so its success rates verify execution under the released protocol rather than provide a direct comparison with Table 2.

B.5 Matched overshooting

A separate four-seed development study reuses the same task caches and blind banks to compare DRF with standard multistep overshooting under matched wall time and gradient budget. DRF shows no cross-task advantage over this stronger baseline, and all relevant paired intervals include zero. The prespecified claim that DRF survives matched overshooting therefore fails. This result is development-only and is not pooled with the five-seed confirmation panel.

C Development Ledger

Development-only studies covered full-BPTT and overshooting controls; local and multiscale responses; anchored, projected, trust-region, and adaptive gradients; planner reranking; response prediction and curricula; and periodic auxiliary updates. None met its frozen cross-task promotion criterion. The supplementary ledger records each question and outcome; no branch is promoted after observing the confirmation bank.

C.1 Wall transfer attempt

A preregistered Wall transfer panel produced no training or method result. Successive attempts stopped during cache construction, held-out-data validation, and import recovery. The final recovery reached bank construction but failed the frozen candidate-coverage eligibility threshold before smoke evaluation or optimization. The panel is therefore an infrastructure and endpoint-coverage record, not evidence for or against DRF or FG-AS.

D Implementation Checks

The release includes unit tests for the telescoping forward identity, selective suffix-gradient equivalence, full-gradient reference sums, deterministic multiscale edges, cache tamper rejection, gradient calibration, wall-clock stopping, and process cleanup. The confirmation source, data manifests, banks, checkpoints, evaluation records, and decisions are hash-bound. Figure 4 is generated from all 40 registered rows without filtering; its plotting script and JSON inputs are included in the supplementary material.