When Do Updates Matter? Counterfactual Update Redundancy in 4D Gaussian Splatting

Anonymous authors
Paper under double-blind review

Abstract

Dynamic Gaussian splatting optimizes tens of thousands of explicit primitives throughout training, yet the value of individual optimizer updates remains poorly understood. We introduce a counterfactual measurement framework that evaluates the one-step value of a candidate Gaussian update from a fixed training state using training-only probe views and exact rollback. On 4D Gaussian Splatting, update utility becomes sharply concentrated in mid-to-late training: the top 25% of candidate updates account for 97.2–99.8% of positive utility. Broader probe averaging improves identity agreement, but does not recover a context-invariant ranking: from 1 to 24 probe views, mean cross-probe Spearman rises from 0.019 to 0.381 while top-25 positive-utility coverage remains 96.9–100%. Under strict post-densification fixed-identity comparisons across five result-blind paired Train-Opt contexts, temporal rank correlation remains near zero (mean Spearman 0.010 ± 0.020, range -0.018 to 0.036). In a 48-cell stage-by-budget study normalized by empirical three-seed Full-run variation, 33/48 cells satisfy the frozen selection-sensitive rule and 20/48 are budget-starved; across ten threshold-adjacent cells, the sensitive/non-sensitive classification is consistent across all three seeds in 9/10 cells. Removing the systematically weak Motion policy reduces selection-sensitive cells from 33/48 to 27/48 and median policy spread from 1.590 to 0.359 dB. Targeted semantic bridges on both 4D-GS and Deformable-3D-GS further show that selective optimizer updates and active-only backward are different interventions because shared deformation gradients couple primitives. The concentration and path-dependence patterns also recur on the second backbone. Together, the experiments show that one-step utility is highly concentrated, while sustained selective-update outcomes are separately shaped by scene, stage, sparsity path, and shared-gradient execution semantics.

1 Introduction

3D Gaussian Splatting (3DGS) represents scenes with differentiable Gaussian primitives that can be optimized efficiently and rendered in real time (Kerbl et al., 2023). Dynamic extensions retain this explicit representation while introducing time-conditioned deformation (Wu et al., 2024; Yang et al., 2024). These models repeatedly update thousands of Gaussian parameters, often together with a shared deformation representation. Because the primitives are explicit and heterogeneous, their optimization demand need not be uniform across space or training time. We ask a basic question: *what is the counterfactual value of executing a particular Gaussian optimizer update now, and when does selecting updates materially change training?*

Existing work reduces Gaussian cost through structured representations and densification (Lu et al., 2024; Kheradmand et al., 2024), optimizer decoupling (Ding et al., 2026), pruning and training-system optimization (Hanson et al., 2025; Hahlbohm et al., 2026), and renderer-level sparsity (Yang et al., 2026). More broadly, coordinate descent updates subsets of variables (Nesterov, 2012), while sparse-backpropagation and gradient-compression methods select or transmit only part of the gradient/update signal (Sun et al., 2017; Lin et al., 2018). These lines design sparse update rules for convergence, computation, or communication; they do not directly measure the frozen-state counterfactual value of

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.

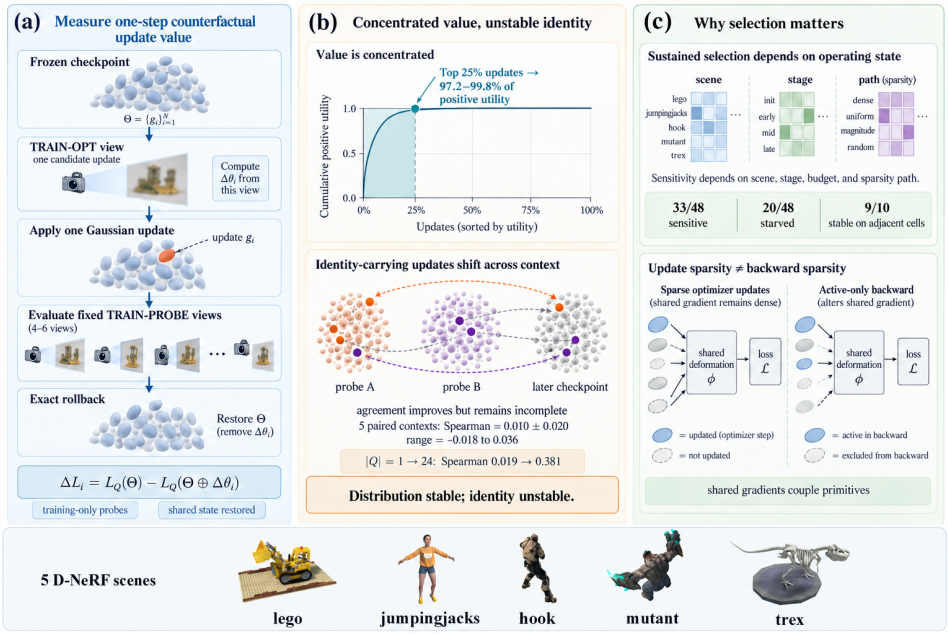


Figure 1: Study overview. (a) Paired counterfactual measurement with training-only probes and exact rollback. (b) Utility is concentrated while valuable-update identities shift across context and time. (c) Sustained selection depends on operating state, while shared gradients make optimizer-update sparsity distinct from sparse backward execution. Bottom: five D-NeRF scenes.

an admissible explicit primitive update. A Gaussian can participate in rendering while contributing little useful optimization progress, and the same primitive can become important later.

We therefore construct a paired counterfactual oracle. At a frozen checkpoint, we obtain optimizer proposals from a training view, evaluate a fixed set of training-only probe views, apply exactly one candidate Gaussian update, re-evaluate the identical probes, and restore parameters and optimizer state before evaluating the next candidate. The resulting one-step loss reduction separates two properties that are often conflated: *distribution-level redundancy*, which asks how concentrated useful update value is, and *identity-level predictability*, which asks whether the updates carrying that value are stable enough to localize.

The results reveal a sharp concentration–predictability gap. In mid-to-late 4D-GS training, the top 25% of candidate updates capture 97.2–99.8% of positive counterfactual utility even within visible-only candidate pools, and the concentration recurs on Deformable-3D-GS. Valuable-update identities, however, remain strongly context dependent. Four six-view probes at `jumpingjacks` 14k have mean pairwise Spearman 0.147. Increasing the probe set from 1 to 24 views raises agreement from 0.019 to 0.381, but top-25 overlap remains only 0.267 while positive-utility concentration stays above 0.969. A strict post-densification comparison fixes the same six probe views and 800 Gaussian rows from 16k to 20k. Across five result-blind paired Train-Opt contexts spanning normalized times 0.065–0.799, temporal Spearman is 0.010 ± 0.020 (range -0.018 to 0.036), with Jaccard@25 0.241 ± 0.012. Cheap local proxies provide limited ranking signal, while small learned rankers fit within-checkpoint structure but do not generalize to held-out checkpoints.

This shifts the question from *which primitive is important* to *when selection matters*. A controlled stage-by-budget study over five D-NeRF scenes and 48 operating cells reveals scene-, stage-, budget-, and path-dependent operating states. Using scene variation measured from three independent Full runs, 33/48 primary-grid cells satisfy the frozen selection-sensitive rule and 20/48 are budget-starved, with 15 cells satisfying both. Ten threshold-adjacent cells are repeated across three seeds; the sensitive/non-sensitive classification agrees across all three seeds for 9/10 cells. Selection remains asymmetric: under the frozen 0.10 dB descriptive band, 37/48 cells are risk-sensitive (risk-only) whereas RobustBenefit exceeds the band in only 8/48. Removing Motion reduces selection-sensitive

cells from 33/48 to 27/48, risk-sensitive cells from 37/48 to 29/48, and median policy spread from 1.590 to 0.359 dB. These counts describe the tested grid and targeted robustness audit rather than seed-marginal population frequencies.

Finally, update redundancy does not imply equivalent backward-compute redundancy. Dynamic Gaussian models couple primitives through shared deformation parameters: suppressing an explicit Gaussian optimizer step is different from deleting that primitive's contribution to the shared backward pass. On both tested backbones, restoring dense shared-deformation gradients nearly recovers the selective-update prototype trajectory, whereas active-only backward produces a distinctly different trajectory.

Our contributions are: (i) a controlled counterfactual measurement framework that exposes a concentration–predictability gap in dynamic Gaussian optimization; (ii) a scene-, stage-, budget-, and path-dependent characterization of when selective updating changes full training, including a decomposition showing that sensitivity is more often downside risk than robust benefit; and (iii) a mechanistic separation between optimizer-update sparsity and sparse backward execution under shared deformation-gradient coupling. Appendix A provides the detailed measurements, tables, and implementation evidence moved from the main paper for space.

2 Related Work

Dynamic Gaussian representations. 3DGS introduced an explicit anisotropic-Gaussian representation with differentiable rasterization and adaptive density control (Kerbl et al., 2023). Dynamic extensions model time-varying geometry with different deformation mechanisms: 4D-GS couples 3D Gaussians with a 4D neural voxel representation and deformation predictor (Wu et al., 2024), while Deformable 3D Gaussians learns a shared deformation field over canonical Gaussians (Yang et al., 2024). Spacetime Gaussian Feature Splatting (Li et al., 2024) and SC-GS (Huang et al., 2024) provide additional dynamic formulations. We use these systems as optimization substrates rather than redesigning their scene representation.

Efficient Gaussian optimization. Scaffold-GS and 3DGS-MCMC alter how primitives are parameterized, created, or relocated (Lu et al., 2024; Kheradmand et al., 2024); Speedy-Splat prunes sparse pixels and primitives (Hanson et al., 2025). AdamW-GS studies optimizer-step and state coupling and introduces Sparse Adam (Ding et al., 2026); Faster-GS coordinates algorithmic and systems optimizations (Hahlbohm et al., 2026); FastGS uses multi-view Gaussian importance to regulate densification and pruning and demonstrates acceleration on dynamic reconstruction (Ren et al., 2026); SparseOIT exploits renderer-specific sparse dependencies and optimizes an active set (Yang et al., 2026). Our question is orthogonal: from a fixed model state and an admissible candidate proposal, what loss reduction results from paying for that update now? This also motivates our later distinction between suppressing an explicit optimizer step and removing a primitive's contribution to the shared deformation gradient.

Selective and sparse optimization. Randomized coordinate methods update only selected variables to reduce per-iteration cost (Nesterov, 2012). meProp sparsifies backpropagation by retaining top-k gradient coordinates and updating only the corresponding parameters (Sun et al., 2017), while Deep Gradient Compression sparsifies communicated gradients with momentum-aware correction for distributed training (Lin et al., 2018). Our setting differs in objective and semantics: we use a paired counterfactual intervention to measure the marginal value of one explicit scene-primitive update at a fixed state, then separately test how sustained selection changes training and how shared deformation gradients break the equivalence between update sparsity and backward sparsity.

3 Measuring Counterfactual Update Value

At iteration t, let the optimization state be

$$\Theta_t = \{\theta_1, \ldots, \theta_N, \phi\}, \tag{1}$$

where θ_i denotes the explicit parameters of Gaussian i (position, rotation, scale, opacity, and appearance coefficients) and ϕ denotes shared deformation or feature-field parameters. A candidate update

$b = (i, k)$ optionally selects an attribute group k; unless stated otherwise, we intervene on all explicit parameter groups of Gaussian i while keeping ϕ fixed during the candidate evaluation.

At a frozen checkpoint, one independently sampled training view (TRAIN-OPT) is used to construct the normal optimizer proposals. A fixed set Q of training views (TRAIN-PROBE) defines the probe objective. The one-step counterfactual gain is

$$\Delta L_b^t = L_Q(\Theta_t) - L_Q(\Theta_t \oplus \Delta\theta_b^t), \tag{2}$$

where $\oplus$ applies only candidate b. For each candidate we snapshot the affected parameters and optimizer state, apply the proposal, re-evaluate the same Q without gradient tracking, and exactly restore the pre-intervention state. Training iteration, camera sampling, learning-rate schedule, and densification do not advance inside the loop. We verify parameter, optimizer-state, and probe-loss equality after rollback, and the oracle never loads test views.

Unless stated otherwise, Q contains six fixed training views and uses mean L1 loss. The principal concentration experiment restricts candidates to Gaussians visible under the probe context, removing the trivial explanation that utility only separates rendered from non-rendered primitives. Because updates may be neutral or harmful, define non-negative relevance

$$r_b = \max(\Delta L_b, 0), \qquad C_q = \frac{\sum_{b \in \text{Top-}q} r_b}{\sum_b r_b + \epsilon}, \tag{3}$$

where Top-q is the highest-relevance fraction q of candidates. Positive-utility coverage uses r_b, whereas the reported Gini coefficient is computed on $|\Delta L_b|$; signed gain is retained for rank correlation and sign analyses. We summarize identity stability with Spearman correlation and top-q Jaccard overlap, and predicted rankings with NDCG@q and top-q utility capture. Because gains close to floating-point resolution can change sign without changing their ranking, Appendix A.2 additionally reports a precision-derived near-zero band. The oracle is a measurement instrument rather than a deployable pre-backward scheduler. Full protocol and metric details are in Appendix A.1.

4 CONCENTRATED UTILITY, CONTEXT-DEPENDENT IDENTITY

We first apply the oracle to 4D-GS (Wu et al., 2024) on D-NeRF (Pumarola et al., 2021), sampling 800 visible Gaussians per checkpoint. On `jumpingjacks`, the top 25% of candidate updates capture 99.8%, 98.6%, and 97.2% of total positive gain at iterations 7k, 14k, and 20k, with absolute-gain Gini coefficients 0.85, 0.82, and 0.78. The corresponding raw positive-gain rates are 28.8%, 34.0%, and 38.3%, so concentration describes how positive utility is distributed rather than how frequently it occurs. At 3k, coverage is 67.8% with an 88.3% positive rate; `hook` similarly rises from 59.7% coverage at 3k (91.6% positive) to 98.0% at 14k (30.5% positive; Gini 0.70). On Deformable 3D Gaussians (Yang et al., 2024), four tested scene/checkpoint settings yield 96.4–96.8% top-25 coverage and Gini 0.71–0.77.

High concentration is not accompanied by a context-invariant identity ranking. At the 4D-GS `jumpingjacks` 14k checkpoint, four independently sampled six-view probe sets preserve the distributional pattern (95–99% coverage; Gini 0.76–0.82), but mean pairwise Spearman correlation is only 0.147 and top-25 Jaccard overlap is 0.266. A dedicated probe-size sweep strengthens the distinction: as $|Q|$ increases from 1 to 24 training views, mean pairwise Spearman rises monotonically from 0.019 to 0.381 and Jaccard@25 from 0.017 to 0.267, while top-25 positive-utility coverage remains 1.000 to 0.970. Broader averaging therefore improves agreement without yielding a stable top-update identity.

Temporal non-stationarity persists under stricter identity and proposal-context control. We continuously resume a `jumpingjacks` trajectory through the last densification event (14.9k) and compare the same 800 Gaussian rows at 16k and 20k with the same six TRAIN-PROBE views. For the baseline paired TRAIN-OPT context (view 13, normalized time 0.0653), the Gaussian count remains 25,295 with no structural event between checkpoints; Top-25 coverage is 1.000/0.963 and Spearman is 0.001. We then repeat the same fixed-identity comparison for four additional TRAIN-OPT contexts chosen by a result-blind temporal-quantile rule. Across all five contexts, Spearman is 0.010 ± 0.020 (range -0.018 to 0.036), Kendall 0.012 ± 0.015, and Jaccard@25 0.241 ± 0.012. Appendix A.4 reports the full paired-context audit and tie handling.

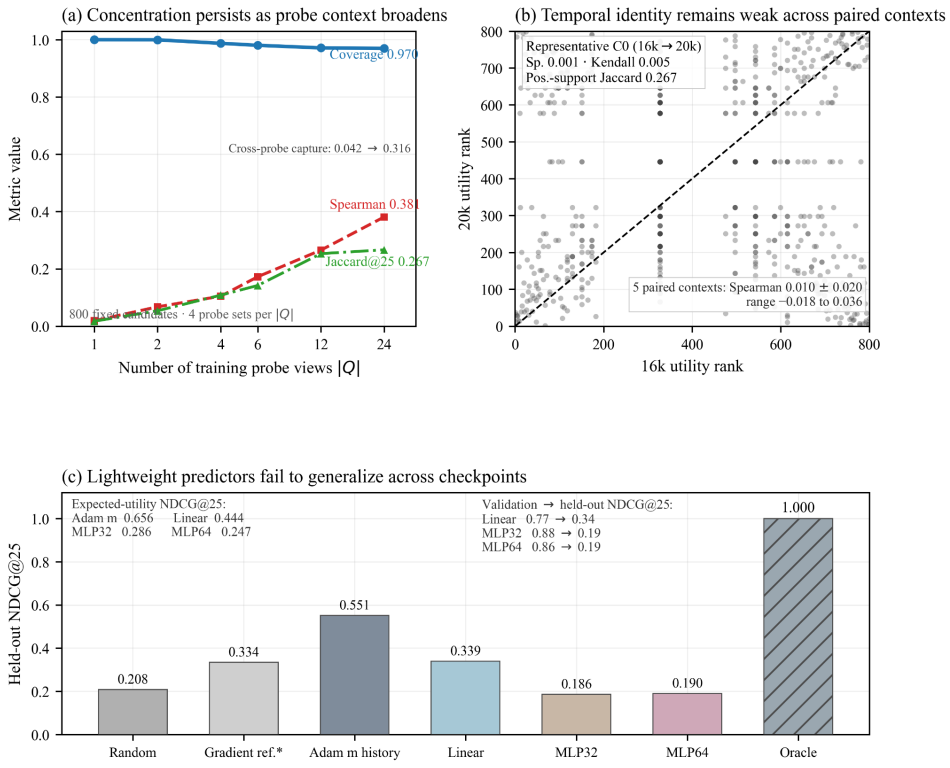


Figure 2: Concentrated one-step utility does not imply stable update identity. (a) As training-only probes increase from 1 to 24, Spearman rises from 0.019 to 0.381 while Top-25 positive-utility coverage stays above 0.969. (b) Representative C0 rank scatter at 16k/20k; across five result-blind paired Train-Opt contexts, temporal Spearman is 0.010 ± 0.020 (range -0.018 to 0.036). (c) Lightweight primitive-local predictors generalize poorly to held-out checkpoints. *The current gradient is a post-backward reference, not a deployable pre-backward signal.

Table 1: Concentration and predictability summary. Top-25 denotes positive-utility coverage. Gini is computed on absolute signed gain.

Setting	Distribution statistic	Identity / prediction statistic
4D-GS, mid/late	Top-25=0.972–0.998; Gini=0.698–0.852; raw positive=0.288–0.383	–
4D-GS, six-view probes	Top-25=0.953–0.986; Gini=0.765–0.822	Spearman=0.147; J25=0.266
Probe sweep, $\|Q\|$: 1→24	Top-25=1.000→0.970	Spearman=0.019→0.381; J25=0.017→0.267
C0 distribution; 5-context identity audit	C0 Top-25=1.000→0.963; Gini=0.781→0.795	Spearman=0.010 ± 0.020; J25=0.241 ± 0.012
Deformable-3D-GS	Top-25=0.964–0.968; Gini=0.712–0.773	Probe Spearman=0.098; temporal=0.246
Prev. Adam m, held-out	–	NDCG@25=0.551; capture=0.541
Ranking MLP, held-out	–	NDCG@25=0.14–0.19; capture=0.14–0.18

Cheap pre-backward signals do not resolve the gap. On held-out 4D-GS checkpoints, the strongest deployable single feature, the previous-step Adam first-moment norm, reaches NDCG@25=0.551 and top-25 capture=0.541. Small ranking MLPs trained on seven pre-backward features obtain within-checkpoint validation NDCG@25 near 0.93 but only 0.14–0.19 on held-out `jumpingjacks` and `hook` checkpoints, with capture 0.14–0.18. Averaging utility over four probes improves labels but not generalization: the best learned predictor reaches 0.444 NDCG@25 versus 0.656 for Adam history. Table 1 summarizes the central measurements; Appendix A.5 gives predictor details and Appendix A.3 gives the full probe-size sweep.

5 When Does Selection Matter?

We next sustain selective updating during training. Forward and backward remain dense, but explicit Gaussian optimizer steps are restricted to an exact-K active set. This isolates selection from changes to the shared deformation gradient. We vary the schedule-start fraction s and local active ratio β across Random, Round-Robin, Visibility, Motion, and a fixed history-based policy. At each scheduled iteration, $K = \mathrm{round}(\beta N)$ explicit rows are selected; exact policy definitions and fallback semantics are given in Appendix A.6. We distinguish the designed budget

$$B_{\mathrm{theory}} = 1 - f_{\mathrm{mask}}(1 - \beta) \tag{4}$$

from the trajectory-measured budget for a concrete run,

$$B_{\mathrm{meas}} = \frac{\sum_t |S_t|}{\sum_t N_t^{\mathrm{Full}}}, \tag{5}$$

where f_{mask} is the fraction of total training iterations in the scheduled selective-update window (Appendix A.7). Because policy-dependent densification can make designed and realized expenditure differ, for each scene–start–budget cell, with final PSNR Q, define $D_{\mathrm{best}} = Q_{\mathrm{best}} - Q_{\mathrm{Full}}$, $P = Q_{\mathrm{best}} - Q_{\mathrm{worst}}$, and $P_{\mathrm{norm}} = P/\sigma_{\mathrm{scene}}$. Here σ_{scene} is the sample standard deviation of final Full-run PSNR across three independent seeds for that scene; it is used only as a normalized scene-variation scale, not as an inferential standard error. The frozen selection-sensitive rule requires $P_{\mathrm{norm}} \geq 2$, $P \geq 0.10$ dB, and $R_{\mathrm{rand}} > 0.05$ dB or $R_{\mathrm{RR}} > 0.05$ dB, where $R_x = Q_{\mathrm{best}} - Q_x$; budget starvation is $D_{\mathrm{best}} < -0.15$ dB. A cell may satisfy neither criterion or both. Appendix A.6 reports the empirical scene scales, threshold sensitivity, and targeted multi-seed audit.

Across 48 4D-GS cells over five D-NeRF scenes, 33 satisfy the frozen selection-sensitive rule and 20 are budget-starved under empirical scene scales, with 15 cells satisfying both (Fig. 3). For the early-start schedule, `jumpingjacks` leaves starvation between $\beta = 25\%$ and 35%, while `hook` remains starved at 35%. At $\beta = 90\%$, `jumpingjacks` and `hook` remain selection-sensitive ($P_{\mathrm{norm}} = 11.1$ and 3.4), whereas `mutant` and `trex` no longer satisfy the normalized criterion ($P_{\mathrm{norm}} = 1.62$ and 0.79). We therefore treat the operating map as discrete tested states rather than a universal phase boundary. Ten threshold-adjacent cells were repeated with seeds 6666/7777/8888; the sensitive/non-sensitive classification is consistent across all three seeds in 9/10 cells, while starvation, risk-only, and RobustBenefit labels are less stable (3/3 agreement in 6/10, 5/10, and 5/10 cells).

Designed budget alone is insufficient, and measured budget must be tracked separately. For the exact-designed-budget pairs in Fig. 3(b), $B_{\mathrm{theory}} = 0.826$ for both paths; the winning-policy measured budgets are 0.762/0.762 on `lego`, 0.769/0.776 on `jumpingjacks`, and 0.740/0.723 on `trex`. The corresponding D_{best} differences are 0.009, 0.281, and 0.287 dB, respectively. On Deformable-3D-GS, a `jumpingjacks` pair at $\beta = 90\%$ with starts at 20% and 60% has absolute measured-budget differences 0.007, 0.017, and 0.0069 across three independent seeds; two seeds exceed the frozen outcome-divergence threshold and the third is directionally consistent but below it. We therefore treat path dependence as an empirical, scene-dependent effect rather than a universal law; Appendix A.7 reports both designed and measured budgets for the 4D-GS examples.

Policy spread is mainly downside risk. With

$$\mathrm{Benefit} = Q_{\mathrm{best}} - Q_{\mathrm{Random}}, \quad \mathrm{Risk} = Q_{\mathrm{Random}} - Q_{\mathrm{worst}}, \tag{6}$$

$$\mathrm{RobustBenefit} = Q_{\mathrm{best}} - \max(Q_{\mathrm{Random}}, Q_{\mathrm{RR}}), \tag{7}$$

Using the frozen $\delta = 0.10$ dB descriptive band, let $R_\delta = \mathbf{1}[\mathrm{Risk} > \delta]$ and $B_\delta = \mathbf{1}[\mathrm{RobustBenefit} > \delta]$; 37/48 cells are risk-sensitive (risk-only) ($R_\delta = 1, B_\delta = 0$), 6/48 are mixed, 2/48 are benefit-only, and 3/48 are neither. The unique non-starved benefit-only cell is `trex`, early start, $\beta = 90\%$, which no longer satisfies the empirical-σ selection-sensitive rule. Motion is worst in 42/48 cells. Removing Motion reduces selection-sensitive cells from 33 to 27, risk-only cells from 37 to 29, and median policy spread from 1.590 to 0.359 dB; budget-starved cells change from 20 to 21 and RobustBenefit remains 8/48 at $\delta = 0.10$. Motion therefore contributes to a subset of sensitive cells and strongly amplifies downside magnitude, but the risk-dominant pattern persists without it. The second backbone shows the same qualitative asymmetry with a different loser identity: Benefit exceeds 0.10 dB in 1/12 cells and Risk in 5/12. Figure 4 visualizes the decomposition; Appendix A.9 reports the full ablation.

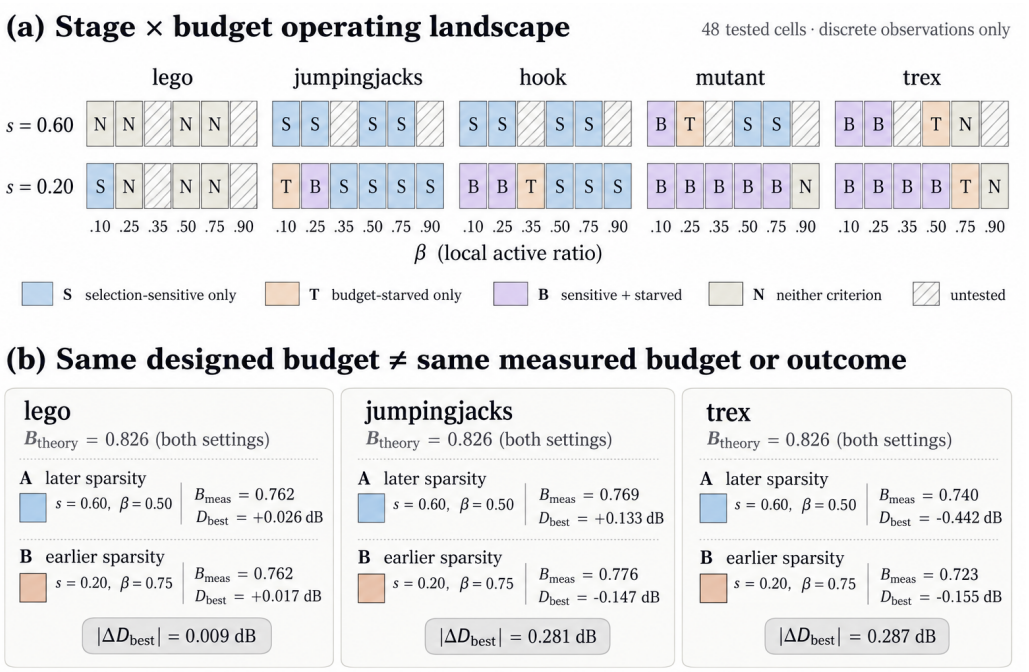


Figure 3: Selective-update behavior depends jointly on scene, training stage, local active ratio, and sparsity path. (a) Using empirical three-seed Full-run variation, S denotes selection-sensitive only, T budget-starved only, B both, and N neither criterion; hatched cells were not tested. The 48-cell primary grid contains 33 selection-sensitive and 20 budget-starved cells, with overlap allowed. (b) The showcased path pairs share $B_{\text{theory}} = 0.826$ but differ in trajectory-measured B_{meas} and final D_{best}, so equal designed budget need not imply equal realized expenditure or outcome.

Table 2: Cross-backbone scope of the main regime and semantic findings. Comparisons are structural; absolute PSNR is not compared across backbones.

Finding	4D-GS	Deformable-3D-GS
Scene-dependent regimes	Five scenes with distinct boundaries	`lego` redundancy; `jumpingjacks` 5/6 sensitive-or-starved
Stage / β dependence	Strong transitions across phase plane	Strong on `jumpingjacks`; absent on `lego`
No universal winner	Best policy varies by cell	Random 6 / Visibility 5 / RR 1
Matched-budget path effect	Close-budget cells can differ strongly	3/3 seeds close in B_{meas}; 2/3 exceed divergence threshold
Benefit vs. risk	37/48 risk-only; RobustBenefit> 0.10 in 8/48	Benefit> 0.10 in 1/12; Risk> 0.10 in 5/12
Shared-gradient bridge	$H \approx P \ll T$ on `trex`	Same ordering on `jumpingjacks`; $d(H,P)/d(T,P) = 0.0094$

6 Update Sparsity Is Not Compute Sparsity

The intervention above keeps rendering and shared-deformation backward dense while restricting explicit Gaussian optimizer steps. This differs from true sparse backward execution. Let $f_i(\phi)$ be the shared deformation output used by Gaussian i. The dense shared-model gradient is

$$\nabla_\phi \mathcal{L} = \sum_{i=1}^{N} J_\phi f_i(\phi)^\top \nabla_{f_i} \mathcal{L}. \tag{8}$$

Selective optimizer updating keeps this gradient dense and applies explicit optimizer steps only to θ_i for $i \in S$. Active-only backward instead detaches inactive deformation queries, yielding

$$\nabla_\phi^S \mathcal{L} = \sum_{i \in S} J_\phi f_i(\phi)^\top \nabla_{f_i} \mathcal{L}, \tag{9}$$

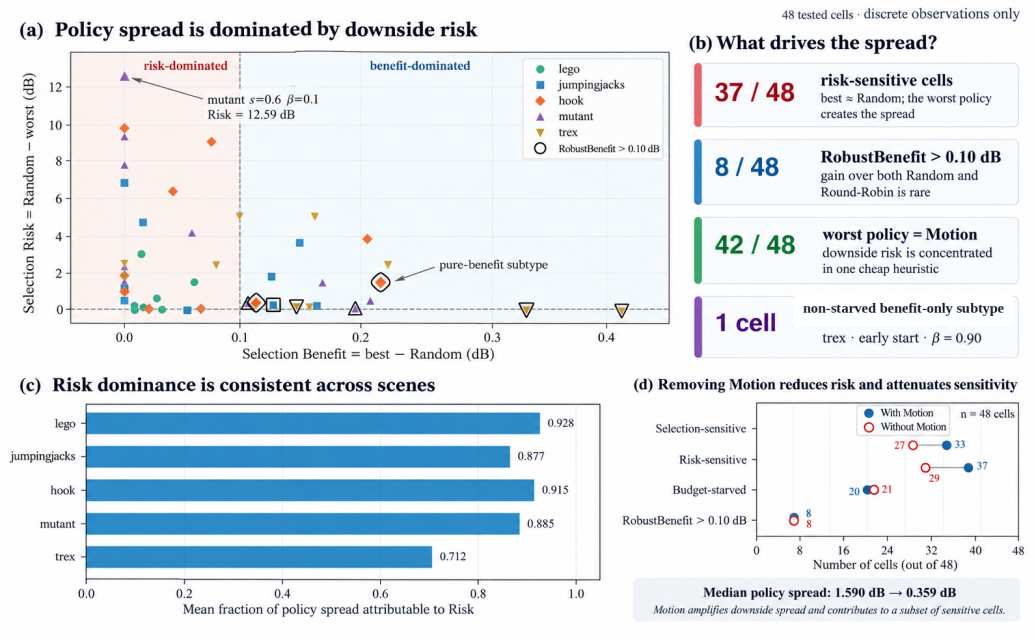


Figure 4: Policy spread is dominated by downside risk rather than consistent upside from a particular heuristic. (a–c) Under the frozen 0.10 dB descriptive band, poor choices account for most of the observed spread, RobustBenefit exceeds the band in only 8/48 cells, Motion is worst in 42/48, and the highlighted `trex` cell is the unique non-starved benefit-only subtype. (d) Under empirical scene calibration, removing Motion reduces selection-sensitive cells from 33/48 to 27/48 and risk-sensitive cells from 37/48 to 29/48, while RobustBenefit remains 8/48 and budget-starved cells change from 20/48 to 21/48; median policy spread falls from 1.590 to 0.359 dB.

which generally differs from the dense gradient. Dropping an explicit update and dropping that primitive's contribution to the shared backward pass are therefore different interventions.

We implement strict row-wise Adam so active rows update parameters, moments, and row-wise step counters while inactive rows remain bitwise unchanged. Six row-optimizer tests and a four-part sparse correctness battery pass; on a real `jumpingjacks` checkpoint, all-active and 25%-active sparse forward paths match dense rendering with maximum absolute error 0.0.

A targeted 300-iteration `trex` diagnostic at $\beta = 0.9$ isolates the semantic difference (Fig. 5). Prototype P uses dense forward/backward followed by selective explicit updates. Hybrid H retains the same dense shared-deformation gradient but uses strict row-wise Adam for selected rows. True-sparse T uses the same row-wise optimizer but propagates deformation gradients only through active queries. Using the 300-step training-loss sequences ℓ_t^A for arm $A \in \{H, T\}$ and prototype P, we measure the relative trajectory deviation

$$d(A, P) = \frac{\sum_{t=1}^{300} |\ell_t^P - \ell_t^A|}{\sum_{t=1}^{300} |\ell_t^P|}, \tag{10}$$

which gives

$$d(H, P) = 8.3 \times 10^{-6}, \qquad d(T, P) = 0.133. \tag{11}$$

Thus optimizer-row semantics alone have negligible effect, while removing inactive contributions to the shared deformation gradient changes the trajectory substantially. A controlled Deformable-3D-GS `jumpingjacks` replication at $\beta = 0.9$ shows the same qualitative ordering: H matches P's step-1 shared gradient to relative error 6.9×10^{-8}, whereas T differs by 0.0966; over 300 steps, $d(H, P) = 1.25 \times 10^{-4}$ and $d(T, P) = 1.33 \times 10^{-2}$. The T arm ends with slightly higher holdout PSNR, so this evidence concerns intervention semantics and trajectory change rather than quality ordering (Appendix A.10).

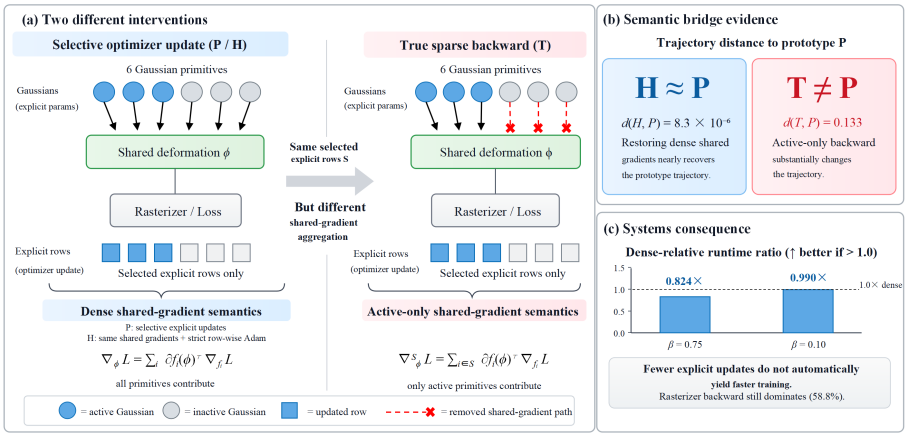


Figure 5: Update sparsity is not compute sparsity. Restoring dense shared gradients nearly recovers the selective-update prototype, whereas active-only backward changes the trajectory; fewer explicit updates do not automatically speed training.

Systems profiling supports this distinction: rasterizer backward remains 58.8% of iteration time because inactive Gaussians still render; scheduler and row-wise Adam fall to 0.06–0.07 ms and 1.5 ms, yet dense Adam is 0.87 ms. Even after optimization, real-model microbenchmarks reach only $0.824\times/0.990\times$ dense at $\beta = 0.75/0.10$. Thus update and compute redundancy are distinct; Appendix A.10 gives the full profile.

7 Discussion and Limitations

The counterfactual oracle is a measurement instrument, not a deployable scheduler: it requires a backward pass and many paired re-evaluations. The predictability gap is empirical rather than an impossibility result; richer global or context-aware predictors may generalize better than the lightweight signals tested here.

One-step gain is local, not long-horizon causal importance. In a pre-specified fixed-mask bridge, an Oracle top-25% mask capturing 99.77% of t_0 positive utility beats the tie-corrected history baseline but not five matched-budget Random masks on the independent 100-step training-only holdout; secondary test PSNR is similar. We therefore do not claim persistent scheduling advantage. The paired temporal audit spans five result-blind Train-Opt contexts but one scene/checkpoint pair.

Individual gains can approach floating-point resolution. Five repeats are bit-identical, float64 accumulation preserves within-scene ranking exactly, and all 26.5% sign flips in a 400-candidate audit lie inside $|\Delta L| \leq 1.699 \times 10^{-11}$; deadband-aware concentration remains high (Appendix A.2).

The 48-cell primary grid remains single-seed, so headline counts are tested-grid descriptions rather than seed-marginal probabilities. Three-seed Full-run scene scales span 0.0121–0.5584 dB, and ten threshold-adjacent repeats give 3/3 agreement in 9/10 cells for sensitive/non-sensitive classification but only 6/10, 5/10, and 5/10 for starvation/risk-only/RobustBenefit. Scope remains two dynamic Gaussian backbones, mainly synthetic D-NeRF; static 3DGS, real-world dynamic data, and broader scene/trajectory replication remain future work.

8 Conclusion

Counterfactual interventions reveal highly concentrated one-step utility but context- and time-dependent identity, while sustained selection varies with scene, stage, budget, and path. Near-complete one-step capture need not yield a fixed-mask advantage, and shared deformation gradients make update sparsity distinct from backward sparsity on both tested backbones.

AI USE STATEMENT

A generative AI assistant was used to assist with language editing and polishing of author-written text, including improvements to grammar, wording, clarity, concision, and readability. It was not used as a source of experimental measurements and did not autonomously conduct experiments or generate reported results. All technical content, experimental results, interpretations, and final claims were produced and verified by the authors. The authors take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

Section 3 specifies the counterfactual intervention and evaluation metrics; Section 5 defines the selective-update intervention and frozen criteria; and Section 6 defines the semantic-bridge comparison. Appendix A reports the probe-size sweep, five-context post-densification identity audit, deadband-aware numerical controls, the short-horizon fixed-mask bridge, empirical three-seed Full-run scene scales, the ten-cell threshold-adjacent multi-seed audit, designed-versus-measured budget semantics, Motion-policy ablation, predictor implementation details, the second-backbone semantic bridge, optimizer-correctness checks, and systems profiling details used by the main paper. The anonymous supplementary artifact identifies canonical frozen CSV/JSON sources, hashes, seed/config manifests, figure/table provenance, and reproduction commands; it additionally includes the five-context temporal audit, short-horizon fixed-mask bridge, and second-backbone semantic bridge, while superseded exploratory outputs are explicitly marked non-canonical.

REFERENCES

R. Ding, Y. Wang, M. Liu, J. Zhu, J. Wang, J. Zhao, W. Shen, F. He, and X. Chen. A Step to Decouple Optimization in 3DGS. In *ICLR*, 2026.

F. Hahlbohm, L. Franke, M. Eisemann, and M. Magnor. Faster-GS: Analyzing and Improving Gaussian Splatting Optimization. In *CVPR*, pp. 18946–18957, 2026.

A. Hanson, A. Tu, G. Lin, V. Singla, M. Zwicker, and T. Goldstein. Speedy-Splat: Fast 3D Gaussian Splatting with Sparse Pixels and Sparse Primitives. In *CVPR*, pp. 21537–21546, 2025.

Y.-H. Huang, Y.-T. Sun, Z. Yang, X. Lyu, Y.-P. Cao, and X. Qi. SC-GS: Sparse-Controlled Gaussian Splatting for Editable Dynamic Scenes. In *CVPR*, pp. 4220–4230, 2024.

B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), Article 139, 2023. doi:10.1145/3592433.

S. Kheradmand, D. Rebain, G. Sharma, W. Sun, Y.-C. Tseng, H. Isack, A. Kar, A. Tagliasacchi, and K. M. Yi. 3D Gaussian Splatting as Markov Chain Monte Carlo. In *NeurIPS*, 2024.

Z. Li, Z. Chen, Z. Li, and Y. Xu. Spacetime Gaussian Feature Splatting for Real-Time Dynamic View Synthesis. In *CVPR*, pp. 8508–8520, 2024.

T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai. Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering. In *CVPR*, pp. 20654–20664, 2024.

A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, pp. 10318–10327, 2021.

Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *ICLR*, 2018.

Y. Nesterov. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. doi:10.1137/100802001.

S. Ren, T. Wen, Y. Fang, and B. Lu. FastGS: Training 3D Gaussian Splatting in 100 Seconds. In *CVPR*, pp. 26094–26103, 2026.

X. Sun, X. Ren, S. Ma, and H. Wang. meProp: Sparsified Back Propagation for Accelerated Deep Learning with Reduced Overfitting. In *ICML*, pp. 3299–3308, 2017.

G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. In *CVPR*, pp. 20310–20320, 2024.

Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. In *CVPR*, pp. 20331–20341, 2024.

W. Yang, F. Kong, Z. Kang, and X. Huang. SparseOIT: Improving Order-Independent Transparency 3DGS via Active Set Method. In *CVPR*, pp. 41012–41021, 2026.

A Detailed Experimental Evidence

This appendix collects detailed measurements from the original study together with targeted reviewer-closure controls added before submission. All added controls use training-only data and author-run code; they are reported here to make the claim boundaries and robustness checks explicit.

A.1 Counterfactual protocol and metrics

Update unit and rollback. The candidate proposal is the optimizer step computed from one fixed Train-Opt view. Unless an attribute group is explicitly selected, a candidate is a full-primitive update over all explicit parameter groups of Gaussian i while shared parameters ϕ remain fixed during the candidate intervention. Train-Opt and Train-Probe are sampled independently from the training split. In the primary experiments, Train-Opt contains one view and Q contains six fixed training views. For each candidate, the affected explicit parameters and optimizer state are snapshotted, the candidate is applied, the identical Q is re-evaluated without gradient tracking, and the state is exactly restored. Training iteration, camera sampling, learning-rate schedule, and densification do not advance inside this loop. Rollback is checked through equality of parameters, optimizer state, and probe loss after restoration.

Concentration versus predictability. The non-negative relevance $r_b = \max(\Delta L_b, 0)$ is used for positive-utility coverage and NDCG/capture metrics. Signed counterfactual gain is retained for rank-correlation and sign analyses. The reported Gini statistic is $\mathrm{Gini}(|\Delta L|)$, matching the implementation; positive-utility prevalence is reported separately. Identity stability is summarized by Spearman rank correlation and top-q Jaccard overlap. The oracle intentionally uses a backward pass to obtain candidate proposals and therefore is not a deployable pre-backward scheduler.

A.2 Numerical precision control

A targeted precision audit evaluates 400 candidates across `jumpingjacks` and `hook`. Five exact repeats are bit-identical after rollback. Replacing the default probe-loss accumulation with float64 accumulation leaves the within-scene ranking unchanged (Spearman 1.0 and top-25 Jaccard 1.0 in both scenes), with maximum absolute gain discrepancy on the order of 10^{-11}. Of the 400 candidates, 106 (26.5%) change raw gain sign; every sign flip lies inside the empirical deadband

$$\tau = 1.699 \times 10^{-11}.$$

For diagnostic prevalence statistics we therefore distinguish resolvably positive ($\Delta L > \tau$), near-zero ($|\Delta L| \le \tau$), and resolvably negative ($\Delta L < -\tau$) gains. To test whether near-zero signs drive the concentration result, we additionally define $r_b^\tau = \max(\Delta L_b - \tau, 0)$ and recompute coverage without changing the primary oracle target.

Table 3: Deadband-aware concentration robustness. CI is a 95% nonparametric candidate-bootstrap interval (2,000 resamples) for the deadband-aware Top-25 coverage.

Setting	Raw cov.	r^τ cov.	Resolvable positive	95% CI
`jumpingjacks` 7k	0.998	0.999	0.288	[0.996,1.000]
`jumpingjacks` 14k	0.986	0.996	0.313	[0.991,0.999]
`jumpingjacks` 20k	0.972	0.987	0.358	[0.975,0.993]
`hook` 14k	0.980	0.989	0.305	[0.982,0.996]

At the 14k four-probe audit, zeroing $|\Delta L| \le \tau$ changes mean pairwise Spearman only from 0.147 to 0.146 and leaves mean Top-25 Jaccard at 0.266. Thus the near-zero band affects gain sign prevalence but not the concentration or identity-stability conclusions.

A.3 Probe-size sensitivity

To test whether cross-probe instability is merely a small-Q artifact, we fix the same 800 visible `jumpingjacks` 14k candidates and the same Train-Opt view, then evaluate four independent

Table 4: Probe-size robustness. Broader averaging improves identity agreement while positive-utility concentration remains high.

$\|Q\|$	Top-25 coverage	Spearman	Jaccard@25	Cross-probe capture
1	1.0000	0.019	0.017	0.042
2	0.9996	0.067	0.054	0.070
4	0.9874	0.105	0.108	0.156
6	0.9805	0.172	0.142	0.189
12	0.9714	0.265	0.253	0.237
24	0.9698	0.381	0.267	0.316

training-only probe sets at each $|Q| \in \{1, 2, 4, 6, 12, 24\}$. The corrected top-25 statistic always uses $K = \lfloor 0.25N \rfloor = 200$ candidates.

The trend is monotone for Spearman: larger probe sets average away part of the context variance. The original four-probe robustness experiment used probe seeds 42–45, whereas this sweep uses a different set of four probe draws at each $|Q|$; accordingly, its $|Q| = 6$ cross-probe capture need not numerically match the earlier approximately 0.50 value. However, even at 24 views the mean rank correlation is only 0.381, Jaccard@25 is 0.267, and cross-probe capture is 0.316. The concentration statistic remains above 0.969 throughout. Thus the evidence supports a narrower claim than context-independent "identity instability": probe averaging improves agreement, but a stable top-update identity is not recovered over the tested range.

A.4 Post-densification fixed-identity temporal audit

The original 14k–20k temporal diagnostic begins before the final densification phase and is therefore not used as fixed-identity evidence in the revised paper. We instead continue one `jumpingjacks` seed-6666 trajectory from the 14k checkpoint, save new checkpoints at 16k and 20k, and audit structural events. The final densification occurs at iteration 14,900. Both post-freeze checkpoints contain exactly 25,295 Gaussians, and no split, clone, prune, reorder, or compaction event occurs between them. The same 800 visible row IDs, the same six training probe views (indices 66, 101, 111, 121, 187, 193), and exactly the same Train-Opt camera/time (index 13, normalized time 0.0653266; seed 123) are evaluated at both checkpoints.

Table 5: Strict post-freeze temporal comparison.

Metric	16k	20k
Gaussian rows	25,295	25,295
Top-25 positive coverage	1.000	0.963
Gini($\|\Delta L\|$)	0.781	0.795
Resolvable-positive rate	0.225	0.391
Near-zero rate	0.375	0.349

Cross-time metric	Value
Spearman	0.001
Kendall	0.005
Deterministic Jaccard@25	0.246
Positive-support Jaccard	0.267
Capture 16k→20k	0.379
Capture 20k→16k	0.584

At 16k only 184 candidates have raw positive gain while top-25 contains 200 candidates, so the Jaccard@25 cutoff includes zero-gain ties. We resolve ties deterministically by raw gain descending and Gaussian ID ascending; the positive-support Jaccard of 0.267 provides a complementary tie-robust set statistic. The primary temporal evidence is the signed-rank correlation, which is unaffected by the top-K cutoff.

The continuation restores model parameters and optimizer state from 14k, but the source checkpoint does not store the learning-rate scheduler state, Python/NumPy/PyTorch/CUDA RNG states, or camera-sampling state. The new 16k and 20k checkpoints are generated in one continuous continuation and therefore define a valid within-trajectory post-freeze comparison, but we do not claim that the new 20k checkpoint bit-reproduces the historical 20k trajectory.

Paired Train-Opt context robustness. To test whether the near-zero temporal correlation is peculiar to view 13, we pre-specify four additional Train-Opt contexts by normalized-time quantiles

0.2/0.4/0.6/0.8, excluding the fixed probe views, before evaluating any new oracle result. The resulting views are 39/79/119/159 (times 0.196/0.397/0.598/0.799). Every context uses the same 16k/20k checkpoint pair, the same 800 Gaussian rows, and the same six Train-Probe views. Across the five paired contexts, Spearman is 0.010 ± 0.020 (range -0.018 to 0.036), Kendall 0.012 ± 0.015, Jaccard@25 0.241 ± 0.012, and positive-support Jaccard 0.275 ± 0.014. Thus the magnitude of set overlap remains nonzero, but near-zero temporal rank agreement is consistent across the tested proposal contexts.

Table 6: Five-context paired Train-Opt temporal audit. Contexts C1–C4 were selected by a result-blind temporal-quantile rule; C0 is the original controlled context.

Ctx.	View/time	Spearman	Kendall	J25	J^+	Cap. 16→20 / 20→16
C0	13 / .065	.001	.005	.246	.267	.379 / .584
C1	39 / .196	.036	.028	.231	.272	.475 / .534
C2	79 / .397	-.018	-.010	.227	.277	.395 / .495
C3	119 / .598	.022	.021	.258	.297	.467 / .606
C4	159 / .799	.010	.014	.242	.262	.405 / .609

A.5 Concentration, identity stability, and predictor details

At the 4D-GS `jumpingjacks` 14k checkpoint, cross-probe top-25 utility capture averages approximately 0.50. The observed top-25 Jaccard of 0.266 is above the random-set expectation of approximately 0.143 but far below complete identity agreement. On Deformable-3D-GS, available cheap proxies provide incomplete explanations of oracle rank (best NDCG@25 ≤ 0.624; $|\mathrm{Spearman}| \leq 0.146$). Previous-step Adam history could not be tested for that backbone because the restored checkpoints did not contain optimizer moments.

The ranking MLP uses seven pre-backward features. It reaches validation NDCG@25 around 0.93 when validation candidates come from checkpoints represented during training but only 0.14–0.19 on held-out `jumpingjacks` and `hook` checkpoints, with top-25 capture 0.14–0.18. A random ranking gives NDCG@25=0.208 and capture=0.204 in the reported comparison. Averaging utility over four independent probes improves label stability; the best learned predictor then reaches 0.444 NDCG@25, compared with 0.656 for previous-step Adam history. The seven inputs are opacity, log scale, log projected radius, log motion magnitude, log previous-step Adam first- and second-moment norms, and normalized training progress. Features are standardized using training checkpoints only. The MLPs use hidden widths 32 or 64 and are trained with within-checkpoint pairwise logistic ranking loss plus a 0.1-weight gain-sign auxiliary loss, Adam at 10^{-3} with weight decay 10^{-4}, and early stopping; the held-out checkpoints are `jumpingjacks` 14k and `hook` 14k.

Short-horizon fixed-mask oracle bridge. We additionally test whether high one-step utility at a real optimizer state translates into a short-window fixed-mask advantage. From the full 16k `jumpingjacks` model+Adam state, we set $\beta = 0.25$ ($K = 6324$) for 300 steps and pre-specify the 100-step endpoint before evaluating outcomes. A full-population oracle ranks all 25,295 rows at t_0; its fixed top-K mask captures 0.9977 of positive utility and all selected rows are resolvably positive. We compare this mask with a tie-corrected history baseline (all ages are equal at the Full branch point, so ranking uses the restored Adam first-moment norms) and five fixed Random masks under identical explicit-row budgets, camera sequence, dense shared backward, and a six-view training-only holdout not used by the oracle. Full training is a dense reference, not a budget-matched arm.

At the pre-specified 100-step holdout endpoint, define $\Delta_{100} = L_0 - L_{100}$ (higher is better) and AUC from the corresponding improvement curve. Oracle obtains $\Delta_{100} = -2.23 \times 10^{-5}$ and $\mathrm{AUC}_{100} = -9.72 \times 10^{-4}$; the history baseline gives -3.47×10^{-5} and -1.59×10^{-3}, while Random averages $(-1.48 \pm 0.17) \times 10^{-5}$ and $(-6.90 \pm 0.79) \times 10^{-4}$ over five masks. Oracle therefore improves over the history baseline but lies below every Random replica on both primary holdout metrics. The same qualitative ordering persists at 300 steps. Secondary standard-evaluation PSNR is close across selective arms (Oracle/Random mean: 35.151/35.145 at 100 steps and 35.124/35.120 at 300), so this diagnostic supports a narrow conclusion: near-complete one-step utility capture does not guarantee a persistent fixed-mask advantage under a different training-only evaluation context.

Table 7: Short-horizon fixed-mask bridge from the full 16k optimizer state. Selective arms share the same 25% explicit-row budget; Full is a dense reference. Random reports mean±sample std over five fixed masks.

Arm	Δ_{100}	AUC_{100}	PSNR@100	PSNR@300
Full	-5.66×10^{-5}	-2.99×10^{-3}	35.127	35.087
Oracle Top-K	-2.23×10^{-5}	-9.72×10^{-4}	35.151	35.124
History (tie-corrected)	-3.47×10^{-5}	-1.59×10^{-3}	35.140	35.121
Random	$(-1.48\pm0.17)\times10^{-5}$	$(-6.90\pm0.79)\times10^{-4}$	35.145 ± 0.005	35.120 ± 0.002

A.6 Selective-update policies and regime definitions

At scheduled iterations the local active set uses $K = \mathrm{round}(\beta N)$ rows. Random samples rows uniformly without replacement. Round-Robin is the staleness-only baseline, prioritizing the least-recently updated rows. Visibility uses the renderer's per-view `visibility_filter`, with the frozen fallback used by the phase-study implementation to maintain a valid mask after densification. Motion ranks the deformation-displacement score evaluated at the fixed reference time used by the scheduler. The representative history-aware policy H1 is frozen before the phase grid and scores each row as

$$s_i^{\mathrm{H1}} = 0.75\,\mathrm{rank}(\|m_i\|_2) + 0.25\,\mathrm{rank}(\log(1+\mathrm{age}_i)),$$

where m_i is the previous-step Adam first moment and age_i is iterations since the row's last explicit update; H1 uses no exploration in the phase study. For each scene we estimate σ_{scene} as the sample standard deviation ($n = 3$, $N - 1$ denominator) of final Full-run PSNR for seeds 6666, 7777, and 8888. We use $P_{\mathrm{norm}} = P/\sigma_{\mathrm{scene}}$ only as a normalized scene-variation index, not as an inferential z-score. The frozen selection-sensitive rule requires $P_{\mathrm{norm}} \geq 2$, $P \geq 0.10$ dB, and either $R_{\mathrm{rand}} > 0.05$ dB or $R_{\mathrm{RR}} > 0.05$ dB; budget starvation is $D_{\mathrm{best}} < -0.15$ dB. The thresholds are unchanged from the primary phase analysis. The threshold-adjacent audit uses the deterministic selector $|P - 0.10| \leq 0.05$ dB, or $|P_{\mathrm{norm}} - 2| \leq 0.5$, or $|D_{\mathrm{best}} + 0.15| \leq 0.05$ dB; this selects exactly the ten replicated cells. The 0.05 dB baseline-deviation gate is audited separately because it is a conjunctive classification gate rather than part of this selector: changing that gate to 0.04/0.05/0.06 dB yields 34/33/32 selection-sensitive cells. The supplementary distance audit reports every cell's distance to all frozen thresholds.

Table 8: Empirical scene variation from three independent Full-training seeds. σ_{scene} is the sample standard deviation of final PSNR.

Scene	Seed 6666	Seed 7777	Seed 8888	σ_{scene}
`lego`	25.0306	25.0539	25.0484	0.0121
`jumpingjacks`	35.2876	35.3101	35.2271	0.0429
`hook`	32.7187	32.6940	32.5973	0.0642
`mutant`	38.0037	37.7925	37.5490	0.2276
`trex`	34.2262	33.1665	34.0012	0.5584

Under these empirical scales the primary-grid selection-sensitive count is 33/48. Varying the normalized threshold from 1.5/2.0/2.5 gives 34/33/31 cells; varying the absolute spread threshold from 0.08/0.10/0.12 dB gives 33/33/32 cells. The starvation count remains more threshold-sensitive (23/20/17 cells for $D_{\mathrm{best}} < -0.10/-0.15/-0.20$ dB). These are descriptive robustness checks on the tested grid, not calibrated hypothesis tests.

A.7 Designed versus measured update budget

The phase grid is controlled by (s, β). With $T_{\mathrm{coarse}} = 3000$ and $T_{\mathrm{fine}} = 20000$, the implementation sets $T_{\mathrm{start}} = \mathrm{round}(sT_{\mathrm{fine}})$ and defines $f_{\mathrm{mask}} = (T_{\mathrm{fine}} - T_{\mathrm{start}})/(T_{\mathrm{coarse}} + T_{\mathrm{fine}}) \in [0, 1]$, the fraction of total training iterations in the scheduled selective-update window. Thus $B_{\mathrm{theory}} = 1 - f_{\mathrm{mask}}(1 - \beta)$. Actual update expenditure is trajectory dependent because policies can change densification and Gaussian counts; each run therefore also has $B_{\mathrm{meas}} = \sum_t |S_t| / \sum_t N_t^{\mathrm{Full}}$. For a cell-level best-outcome comparison we report $B_{\mathrm{meas}}^{\mathrm{best}}$, the measured budget of the policy attaining Q_{best}.

Table 9: Exact-designed-budget path pairs used in Fig. 3(b). Both settings have $B_{\text{theory}} = 0.826$; measured budgets follow the best-policy trajectories.

Scene	$B_{m,A}$	$B_{m,B}$	D_A	D_B	P_A	P_B
lego	0.762	0.762	+0.026	+0.017	0.309	0.079
jumpingjacks	0.769	0.776	+0.133	-0.147	1.258	0.111
trex	0.740	0.723	-0.442	-0.155	0.392	0.375

Here A is later sparsity ($s = 0.60, \beta = 0.50$) and B is earlier sparsity ($s = 0.20, \beta = 0.75$). Thus Fig. 3 no longer conflates the designed and measured budgets.

A.8 SELECTIVE-UPDATE REGIME STUDY

The primary 4D-GS phase study contains 48 single-seed cells across five scenes, two schedule starts, and multiple local active ratios. With the empirical scene scales in Table 8, 33/48 cells are selection-sensitive, 20/48 are budget-starved, 15 satisfy both criteria, and 10 satisfy neither. Per-scene selection-sensitive counts are 1/8 on lego, 9/10 on jumpingjacks, 9/10 on hook, 8/10 on mutant, and 6/10 on trex; the corresponding starvation counts are 0, 2, 3, 7, and 8.

To target reviewer-relevant classification uncertainty rather than rerun the entire grid, we repeat all ten cells selected by the deterministic adjacency rule in Appendix A.6 using seeds 6666/7777/8888. The eight newly replicated cells require 80 selective-training trajectories (two additional seeds $\times$ five policies per cell); all 80 complete. Table 10 reports mean$\pm$sample-standard-deviation across the three seed-level metrics. The sensitive/non-sensitive classification agrees across all three seeds in 9/10 cells; 3/3 agreement is 6/10 for starvation and 5/10 for both risk-sensitive and RobustBenefit labels, so the latter subtype boundaries are explicitly treated as seed-sensitive.

Table 10: Three-seed robustness for the ten threshold-adjacent cells. Metrics are computed per seed and then summarized as mean$\pm$sample std; "Sens." is the number of seeds satisfying the frozen selection-sensitive rule.

Cell (s/β)	P	D_{best}	Benefit	Risk	RobustBenefit	Sens.
hook, 0.2/0.75	0.290 ± 0.157	0.040 ± 0.107	0.018 ± 0.017	0.272 ± 0.149	0.018 ± 0.017	3/3
jumpingjacks, 0.2/0.75	0.153 ± 0.042	-0.075 ± 0.082	0.018 ± 0.032	0.134 ± 0.071	0.018 ± 0.032	3/3
mutant, 0.2/0.75	0.455 ± 0.121	-0.074 ± 0.092	0.123 ± 0.057	0.332 ± 0.097	0.043 ± 0.057	2/3
lego, 0.6/0.75	0.084 ± 0.024	0.017 ± 0.007	0.018 ± 0.015	0.066 ± 0.018	0.003 ± 0.005	0/3
trex, 0.2/0.75	0.707 ± 0.348	0.290 ± 0.528	0.128 ± 0.115	0.580 ± 0.329	0.074 ± 0.128	0/3
mutant, 0.2/0.90	0.249 ± 0.132	-0.001 ± 0.087	0.159 ± 0.088	0.090 ± 0.064	0.111 ± 0.073	0/3
trex, 0.6/0.25	2.588 ± 0.122	0.086 ± 0.369	0.278 ± 0.126	2.311 ± 0.052	0.141 ± 0.043	3/3
trex, 0.6/0.75	0.452 ± 0.115	0.343 ± 0.436	0.139 ± 0.054	0.314 ± 0.073	0.050 ± 0.087	0/3
jumpingjacks, 0.2/0.35	1.260 ± 0.043	-0.150 ± 0.043	0.000 ± 0.000	1.260 ± 0.043	0.000 ± 0.000	3/3
lego, 0.2/0.75	0.063 ± 0.014	0.009 ± 0.017	0.004 ± 0.004	0.059 ± 0.012	0.003 ± 0.005	0/3

Matched-budget path evidence. Table 9 gives the primary 4D-GS exact-B_{theory} comparisons together with their trajectory-measured budgets. On Deformable-3D-GS, the jumpingjacks $\beta = 90\%$ starts at 20% and 60% have absolute B_{meas} differences 0.007, 0.017, and 0.0069 across three independent initializations. For seed 6666, policy spread changes from 0.369 to 0.097 dB, the worst-policy deficit from -0.211 to +0.070 dB, and the best policy from Random to Visibility. Two of the three seeds exceed the frozen outcome-divergence threshold; the third remains close-budget and directionally consistent but below threshold.

Risk decomposition. For a descriptive band δ, define $R_\delta = \mathbf{1}[\text{Risk} > \delta]$ and $B_\delta = \mathbf{1}[\text{RobustBenefit} > \delta]$. We use four mutually exclusive benefit–risk subtypes: risk-sensitive (risk-only) $(R_\delta, B_\delta) = (1, 0)$, benefit-only $(0, 1)$, mixed $(1, 1)$, and neither $(0, 0)$. Budget starvation is an orthogonal flag, not a subtype. This decomposition is descriptive and was not part of the frozen primary regime criterion. At the frozen $\delta = 0.10$ dB band, the counts are 37 risk-only, 2 benefit-only, 6 mixed, and 3 neither; RobustBenefit exceeds the band in 8/48 cells and Motion is the worst policy in 42/48. The unique non-starved benefit-only cell is trex, early start, $\beta = 0.90$; the second benefit-only cell is budget-starved. Across scenes, the mean fraction of policy spread attributable to Risk is 71–93%. Varying δ over 0.08/0.10/0.12 dB gives risk-only counts 35/37/39

and RobustBenefit-above-band counts 10/8/6; after removing Motion the corresponding counts are 29/29/31 and 10/8/6. Thus the risk-dominant interpretation persists over the tested band, while the exact RobustBenefit count is threshold-dependent. On Deformable-3D-GS, only one of 12 regime cells has Benefit above 0.10 dB while five have Risk above 0.10 dB. Under the final reported recount, Round-Robin is worst in 9/12 cells, Visibility in 2/12, and Random in 1/12.

A.9 MOTION-POLICY ABLATION

The original 48-cell analysis uses Random, Round-Robin, Visibility, Motion, and the fixed history-based policy. Because Motion is the worst policy in 42/48 cells, we rerun the analysis on the identical 48 cells after removing Motion while retaining the same scene statistics, thresholds, baselines, and cell definitions.

Table 11: 48-cell empirical-σ robustness to excluding Motion and to the descriptive benefit–risk band δ. Threshold-sensitivity rows report risk-only / RobustBenefit-above-band counts.

Statistic	Original policy set	Without Motion
Selection-sensitive cells	33/48	27/48
Budget-starved cells	20/48	21/48
Risk-sensitive (risk-only), $\delta = 0.10$	37/48	29/48
RobustBenefit > 0.10 dB	8/48	8/48
Mean policy spread (dB)	2.644	1.085
Median policy spread (dB)	1.590	0.359
Risk-only / RobustBenefit, $\delta = 0.08$	35 / 10	29 / 10
Risk-only / RobustBenefit, $\delta = 0.10$	37 / 8	29 / 8
Risk-only / RobustBenefit, $\delta = 0.12$	39 / 6	31 / 6

Removing Motion changes six primary-grid cells from selection-sensitive to non-sensitive and none in the opposite direction, reducing the count from 33 to 27. The median spread falls by more than a factor of four and risk-only cells decrease from 37 to 29 at $\delta = 0.10$, while RobustBenefit remains 8/48 and budget-starved cells change from 20 to 21. Across $\delta = 0.08$–0.12, risk-only counts remain 35–39 with Motion and 29–31 without it, whereas RobustBenefit-above-band counts vary from 10 to 6. Thus Motion contributes to some classification sensitivity and strongly amplifies downside magnitude, but the risk-dominant interpretation is not an artifact of the single 0.10 dB band.

A.10 SPARSE OPTIMIZER SEMANTICS AND SYSTEMS DETAILS

Correctness checks. Strict row-wise Adam updates parameters, moments, and row-wise step counters for active rows while inactive rows remain bitwise unchanged. Six row-optimizer tests and a four-part sparse correctness battery pass. On a real `jumpingjacks` checkpoint, all-active and 25%-active sparse forward paths match the dense rendering output with maximum absolute error 0.0.

Semantic bridge. The 300-iteration `trex` diagnostic uses $\beta = 0.9$ and a fixed motion-based active mask. Prototype P uses dense forward/backward through the shared deformation model followed by selective explicit updates. Hybrid H retains the dense shared-deformation gradient but uses strict row-wise Adam for selected rows. True-sparse T uses the same row-wise optimizer and propagates deformation gradients only through active queries. For the 300-step training-loss sequences, the reported relative trajectory distance is $d(A, P) = \sum_t |\ell_t^P - \ell_t^A| / \sum_t |\ell_t^P|$, yielding $d(H, P) = 8.3 \times 10^{-6}$ and $d(T, P) = 0.133$.

Second-backbone semantic bridge. We repeat the same intervention logic on Deformable-3D-GS using a fresh controlled `jumpingjacks` Full trajectory, a post-structure 20k state with 14,769 Gaussians, a fixed result-blind 90% active mask ($K = 13{,}292$), and a common 300-step camera sequence. Step-0 forward outputs match exactly across $P/H/T$. At step 1, H reproduces P's shared gradient to relative error 6.87×10^{-8}, whereas active-only T differs by 0.0966; selected-row P/H updates agree to maximum absolute parameter-delta error 4.77×10^{-7} with no step mismatch. From the final raw trajectory CSVs, $d(H, P) = 1.25 \times 10^{-4}$ and $d(T, P) = 1.33 \times 10^{-2}$, with holdout distances 6.04×10^{-5} and 7.31×10^{-3}. The ordering $H \approx P \ll T$ therefore replicates qualitatively

on the second backbone. Because per-row relative update tensors were not retained, we treat this as a partial replication rather than an exact equivalence proof. T finishes with slightly higher holdout PSNR (37.871 versus 37.729/37.732 for P/H), reinforcing that the diagnostic tests intervention semantics rather than a quality ordering.

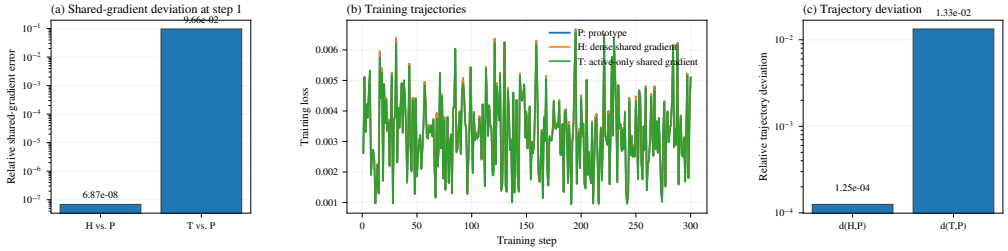


Figure 6: Second-backbone semantic bridge on Deformable-3D-GS. Forward values are matched; H nearly restores the prototype shared gradient and trajectory, whereas active-only T produces a larger shared-gradient and trajectory deviation. Absolute distance magnitudes are not compared across backbones.

Runtime profile. Rasterizer backward accounts for 58.8% of iteration time in the real-model sparse profile because inactive Gaussians remain rendered. Scheduler overhead is reduced to 0.06–0.07 ms and vectorized row-wise Adam from roughly 7.3 ms to 1.5 ms, but sparse row-wise optimization remains slower than dense Adam at 0.87 ms. A quality-safe `lego` $\beta = 0.75$ configuration preserves PSNR within 0.02 dB but runs at 0.61–0.67$\times$ the dense training rate; an aggressive `jumpingjacks` $\beta = 0.10$ configuration runs at 0.47–0.48$\times$ and loses about 1.02 dB. After the optimizer and scheduler optimizations, microbenchmarks improve to 0.824$\times$ at $\beta = 0.75$ and 0.990$\times$ at $\beta = 0.10$, but still do not exceed dense execution.