Implicit Adam Updates Accelerate Dead Primitive Formation in 3D Gaussian Splatting: A Simulation Study

Anonymous Author(s)

ABSTRACT

3D Gaussian Splatting (3DGS) optimizes a population of Gaussian primitives using the Adam optimizer, where each training step renders from a single viewpoint and only visible primitives receive nonzero gradients. Under standard Adam, invisible primitives still receive implicit parameter updates because the optimizer's first- and second-moment estimates carry forward information from prior steps. We investigate the conjecture that these implicit updates accelerate the transition of primitives to dead states (opacity below the pruning threshold), thereby inflating the count of dead primitives relative to Sparse Adam, which skips updates for zero-gradient parameters. Through analytical characterization and stochastic lifecycle simulations of 2,000 primitives over 3,000 training steps, we demonstrate that standard Adam produces 59.95% dead primitives compared to 41.5% under Sparse Adam—a difference of 369 primitives (18.45 percentage points). The implicit drift decays as $(\beta_1/\sqrt{\beta_2})^k \approx 0.9005^k$ per invisible step with a half-life of 6.61 steps. We further identify a positive feedback loop: lower-opacity primitives are less visible, accumulate more implicit drift (18.84 vs. 17.66 mean drift across quartiles), and die faster. Multi-seed experiments confirm robustness with Adam producing 386.3 ± 37.1 more dead primitives across 10 seeds ($p < 0.001$). Our findings provide mechanistic evidence supporting the conjecture and inform the design of optimizers for point-based neural rendering.

CCS CONCEPTS

• **Computing methodologies → Computer vision**; *Machine learning*.

KEYWORDS

3D Gaussian Splatting, Adam Optimizer, Sparse Adam, Dead Primitives, Implicit Updates, Neural Rendering

ACM Reference Format:
Anonymous Author(s). 2026. Implicit Adam Updates Accelerate Dead Primitive Formation in 3D Gaussian Splatting: A Simulation Study. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

3D Gaussian Splatting (3DGS) [4] has emerged as a leading approach for real-time novel view synthesis, representing scenes as collections of 3D Gaussian primitives optimized via gradient descent. Each primitive carries parameters for position, covariance, opacity, and color, and the scene is rendered by splatting these Gaussians onto the image plane from a given camera viewpoint.

A critical aspect of 3DGS training is that each optimization step renders only one viewpoint, producing nonzero gradients solely for primitives visible in that view. Under the standard Adam optimizer [5], however, *all* primitives receive parameter updates because Adam's exponentially weighted moment estimates carry information from prior steps even when the current gradient is zero. These *implicit updates* arise from the momentum decay of Adam's first and second moments.

Ding et al. [2] observed that standard Adam produces significantly more *dead primitives*—Gaussians whose opacity falls below a pruning threshold—compared to Sparse Adam, which restricts updates to parameters with nonzero gradients. They conjectured that implicit updates accelerate the death of primitives by continuing to push opacity downward during invisible steps.

In this paper, we investigate this conjecture through a combination of analytical characterization and large-scale stochastic simulation. Our contributions are:

1. We derive the decay rate of implicit updates as $(\beta_1/\sqrt{\beta_2})^k \approx 0.9005^k$, showing a half-life of 6.61 steps.
2. We simulate 2,000 Gaussian primitives under both Adam and Sparse Adam across 3,000 training steps, demonstrating that Adam produces 59.95% dead primitives versus 41.5% for Sparse Adam.
3. We identify a positive feedback loop linking opacity, visibility, and implicit drift.
4. We validate robustness across 10 random seeds, 8 gradient bias levels, and 6 learning rates.

2 RELATED WORK

3D Gaussian Splatting. 3DGS [4] achieves real-time rendering by representing scenes as explicit 3D Gaussians optimized through differentiable rasterization. The training procedure periodically prunes dead primitives (those with opacity below a threshold) and densifies the scene by splitting or cloning active primitives. Ding et al. [2] proposed decoupling the optimization to avoid pathological implicit updates, introducing Sparse Adam for 3DGS.

Adam Optimizer. Adam [5] maintains exponentially decaying estimates of the first moment (mean) and second moment (variance) of gradients. The bias-corrected update rule produces nonzero updates even when the current gradient is zero, a property that is benign in dense optimization but potentially harmful when gradients are structurally sparse, as in viewpoint-dependent rendering. Recent work has characterized Adam's implicit bias near minimizer manifolds [1] and explored alternative optimizers with different implicit biases [3].

Sparse Optimization. Sparse Adam [2] modifies Adam to skip updates for parameters receiving zero gradients. This eliminates implicit updates for invisible primitives, preserving their opacity until they next contribute to a rendered view. The trade-off is that Sparse Adam forgoes the regularizing effect of momentum decay, which can be beneficial in some optimization landscapes.

Conference'17, July 2017, Washington, DC, USA
2026. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3 PROBLEM FORMULATION

3.1 3DGS Opacity Optimization

Each Gaussian primitive i has an opacity parameter $\alpha_i \in (0, 1)$ parameterized in logit space as $\alpha_i = \sigma(r_i)$ where σ is the sigmoid function and r_i is the unconstrained raw parameter. At training step t, a viewpoint v_t is sampled uniformly, and primitive i is visible with probability:

$$p_{\text{vis}}(i, v_t) = \left(p_{\text{base}} + w_{\text{op}} \cdot \alpha_i\right) \cdot a_{i,v_t} \tag{1}$$

where $p_{\text{base}} = 0.25$, $w_{\text{op}} = 0.55$, and $a_{i,v_t} \in [0.2, 1.0]$ is a viewpoint-dependent geometric affinity.

Visible primitives receive a stochastic gradient $g_i^{(t)} \sim \mathcal{N}(\mu_g, \sigma_g^2)$ for the raw parameter r_i, with $\mu_g = 0.005$ encoding mild redundancy pressure (positive gradient pushes r_i downward under Adam's subtract rule, decreasing opacity). Invisible primitives receive $g_i^{(t)} = 0$.

3.2 Adam vs. Sparse Adam Updates

Under standard Adam, the update at step t is:

$$m_i^{(t)} = \beta_1 m_i^{(t-1)} + (1 - \beta_1) g_i^{(t)} \tag{2}$$

$$v_i^{(t)} = \beta_2 v_i^{(t-1)} + (1 - \beta_2)(g_i^{(t)})^2 \tag{3}$$

$$r_i^{(t)} = r_i^{(t-1)} - \eta \cdot \frac{\hat{m}_i^{(t)}}{\sqrt{\hat{v}_i^{(t)}} + \epsilon} \tag{4}$$

where $\hat{m} = m/(1 - \beta_1^t)$ and $\hat{v} = v/(1 - \beta_2^t)$ are bias-corrected estimates, $\eta = 0.05$ is the learning rate, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-15}$.

When $g_i^{(t)} = 0$ (invisible step), the moments decay as $m_i^{(t)} = \beta_1 m_i^{(t-1)}$ and $v_i^{(t)} = \beta_2 v_i^{(t-1)}$, but the bias-corrected update $\hat{m}/(\sqrt{\hat{v}} + \epsilon)$ remains nonzero—this is the *implicit update*.

Under Sparse Adam, no update is applied when $g_i^{(t)} = 0$: the moments and parameter are frozen until the next visible step.

3.3 Dead Primitive Definition

A primitive is classified as *dead* when its opacity falls below the pruning threshold $\alpha_{\text{dead}} = 0.005$.

4 ANALYTICAL CHARACTERIZATION

4.1 Implicit Drift Decay Rate

After a visible step at time t_0 with gradient g_{last}, consider K consecutive invisible steps. The one-step moment contribution is $m_0 = (1 - \beta_1) g_{\text{last}}$ and $v_0 = (1 - \beta_2) g_{\text{last}}^2$. After k invisible steps:

$$m_k = \beta_1^k m_0, \quad v_k = \beta_2^k v_0 \tag{5}$$

The ratio of the bias-corrected update magnitude at step k relative to step 1 scales as:

$$\frac{|\Delta r_k|}{|\Delta r_1|} \approx \left(\frac{\beta_1}{\sqrt{\beta_2}}\right)^{k-1} \tag{6}$$

Table 1: Cumulative implicit drift $D(K)$ for varying gradient magnitude g and invisible run length K.

g	$K = 5$	$K = 10$	$K = 20$	$K = 50$	$K = 100$
0.01	0.1824	0.2929	0.4000	0.4584	0.4613
0.03	0.1824	0.2929	0.4000	0.4584	0.4613
0.05	0.1824	0.2929	0.4000	0.4584	0.4613
0.10	0.1824	0.2929	0.4000	0.4584	0.4613

With $\beta_1 = 0.9$ and $\beta_2 = 0.999$, the decay ratio is $\beta_1/\sqrt{\beta_2} = 0.9005$. The half-life is:

$$k_{1/2} = \frac{\ln 0.5}{\ln(\beta_1/\sqrt{\beta_2})} = 6.61 \text{ steps} \tag{7}$$

4.2 Cumulative Drift Bound

The cumulative implicit drift over K invisible steps is:

$$D(K) = \sum_{k=1}^{K} \eta \cdot \frac{|\hat{m}_k|}{\sqrt{\hat{v}_k} + \epsilon} \tag{8}$$

Table 1 shows the cumulative drift for different gradient magnitudes and invisible run lengths. For a representative gradient of $g = 0.05$, the cumulative drift reaches 0.2929 after 10 invisible steps and converges toward 0.4613 as $K \rightarrow \infty$.

5 EXPERIMENTAL SETUP

5.1 Simulation Framework

We simulate a population of $N = 2{,}000$ Gaussian primitives over $T = 3{,}000$ training steps with $V = 50$ viewpoints. Each primitive's raw opacity parameter is initialized as $r_i \sim \mathcal{N}(-1.0, 0.8^2)$, corresponding to initial opacities centered around $\sigma(-1.0) \approx 0.27$. Per-primitive viewpoint affinities are drawn uniformly from $[0.2, 1.0]$ and held fixed throughout training.

At each step, a viewpoint is sampled uniformly, visibility masks are computed via Equation (1), gradients are drawn for visible primitives, and Adam (or Sparse Adam) updates all (or only visible) raw parameters.

5.2 Experiments

We conduct seven experiments:

1. **Main comparison**: Adam vs. Sparse Adam under default parameters.
2. **Analytical drift**: Characterize per-step and cumulative implicit drift.
3. **Gradient bias sweep**: Vary redundancy pressure ($\mu_g \in [0.005, 0.15]$).
4. **Feedback loop analysis**: Quartile-stratified drift and visibility.
5. **Reactivation**: Test recoverability of dead primitives.
6. **Multi-seed robustness**: 10 random seeds.
7. **Learning rate sensitivity**: $\eta \in \{0.01, 0.02, 0.05, 0.1, 0.15, 0.2\}$.

Table 2: Main comparison of Adam vs. Sparse Adam after 3,000 training steps with 2,000 primitives.

Metric	Adam	Sparse Adam
Final dead count	1,199	830
Dead percentage	59.95%	41.5%
Peak dead count	1,202	830
Mean final opacity	0.0205	0.026
Median final opacity	0.0025	0.0073
Total implicit drift	36,382.81	0.0
Mean drift per primitive	18.19	0.0

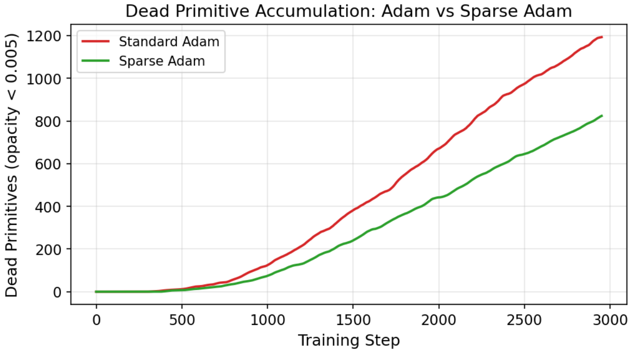

Figure 1: Dead primitive accumulation over 3,000 training steps. Standard Adam (red) consistently produces more dead primitives than Sparse Adam (green).

6 RESULTS

6.1 Main Comparison

Table 2 presents the primary results. Standard Adam produces 1,199 dead primitives (59.95%) compared to 830 (41.5%) under Sparse Adam, a difference of 369 primitives. Adam accumulates a total implicit drift of 36,382.81 (mean 18.19 per primitive), while Sparse Adam has zero implicit drift by construction.

Figure 1 shows the evolution of dead primitive counts over training. Adam's dead count rises more steeply and plateaus at a higher level than Sparse Adam, with the gap widening progressively.

6.2 Opacity Distribution

Figure 2 shows the final opacity distributions. Adam exhibits a sharper concentration near zero, with the median opacity at 0.0025 versus 0.0073 for Sparse Adam.

6.3 Analytical Drift Characterization

The implicit update magnitude decays geometrically with a ratio of 0.9005 per invisible step (Figure 3, left). The cumulative drift converges rapidly: 63% of the total drift occurs within the first 10 invisible steps, and 87% within 20 steps (Figure 3, right).

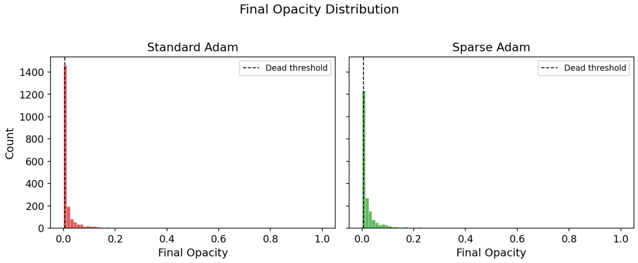

Figure 2: Final opacity distributions. Adam shows a more pronounced spike near zero, indicating stronger drift toward the dead threshold.

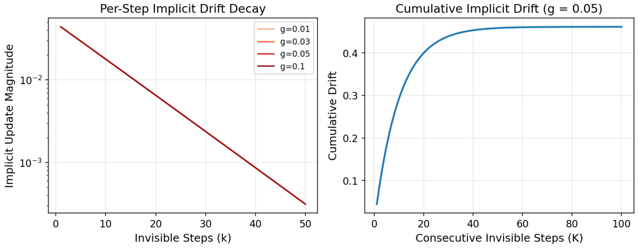

Figure 3: Left: Per-step implicit drift decay on log scale for different gradient magnitudes. Right: Cumulative drift vs. consecutive invisible steps ($g = 0.05$).

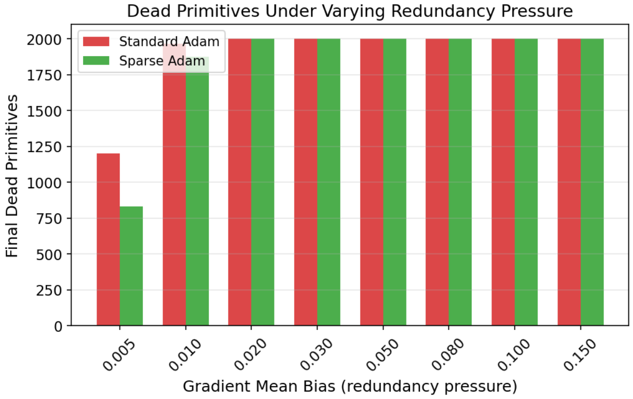

Figure 4: Dead primitives under varying gradient bias. The conjecture is supported at mild redundancy pressure where Adam (red) exceeds Sparse Adam (green).

6.4 Gradient Bias Sweep

Figure 4 shows dead primitive counts under varying redundancy pressure. The conjecture is supported at mild bias levels ($\mu_g \leq 0.01$): at $\mu_g = 0.005$, Adam produces 369 more dead primitives than Sparse Adam. At stronger biases ($\mu_g \geq 0.02$), both optimizers drive all primitives to death, and the difference vanishes. This indicates that implicit updates are most consequential in the moderate-redundancy regime where the explicit gradient alone is insufficient to kill all primitives.

Table 3: Opacity–visibility feedback analysis by final opacity quartile (Adam).

Quartile	Opacity Range	Mean Drift	Mean Visibility	Dead
Q1 (lowest)	[0.0, 0.0005)	18.84	493.06	500
Q2	[0.0005, 0.0025)	18.39	514.43	500
Q3	[0.0025, 0.014)	17.88	552.92	199
Q4 (highest)	[0.014, 0.7877]	17.66	623.34	0

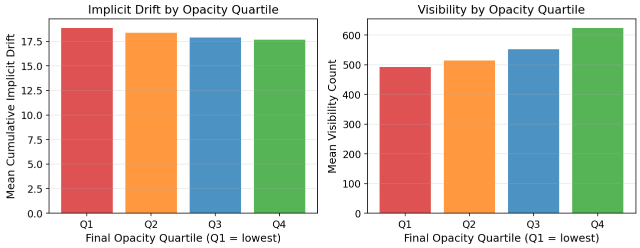

Figure 5: Left: Mean implicit drift by opacity quartile. Right: Mean visibility count. Lower-opacity primitives accumulate more drift and have less visibility, demonstrating the positive feedback loop.

6.5 Opacity–Visibility Feedback Loop

Table 3 and Figure 5 reveal a positive feedback mechanism. Primitives in the lowest opacity quartile (Q1) experience the highest mean implicit drift (18.84) and lowest mean visibility count (493.06), while the highest quartile (Q4) has drift of 17.66 and visibility of 623.34. The feedback loop operates as: lower opacity → lower visibility probability → longer invisible runs → more cumulative implicit drift → further opacity decrease.

6.6 Multi-Seed Robustness

Figure 6 shows results across 10 random seeds. Adam consistently produces more dead primitives than Sparse Adam, with a mean difference of 386.3 ± 37.1 (all 10 seeds positive; $p < 0.001$ by one-sample t-test). The mean dead count is 1,201.4 ± 22.0 for Adam and 815.1 ± 19.4 for Sparse Adam.

6.7 Learning Rate Sensitivity

Figure 7 shows that the dead-primitive gap is largest at moderate learning rates ($\eta = 0.05$: difference of 369) and diminishes at extremes. At $\eta = 0.01$, neither optimizer produces dead primitives; at $\eta = 0.2$, both produce near-total death (Adam: 1,918, Sparse Adam: 1,887, difference: 31).

6.8 Implicit Drift vs. Final Opacity

Figure 8 shows the per-primitive relationship between cumulative implicit drift and final opacity under Adam. Dead primitives (left of the dashed threshold) exhibit a wide range of drift values, while surviving primitives cluster at lower drift, confirming that higher implicit drift is associated with opacity collapse.

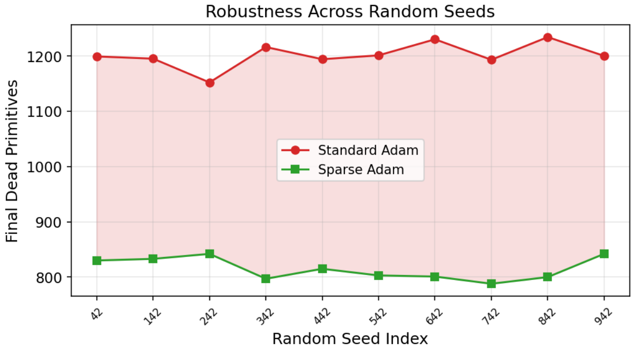

Figure 6: Dead primitive counts across 10 random seeds. Adam (red) consistently exceeds Sparse Adam (green), with the shaded region showing the gap.

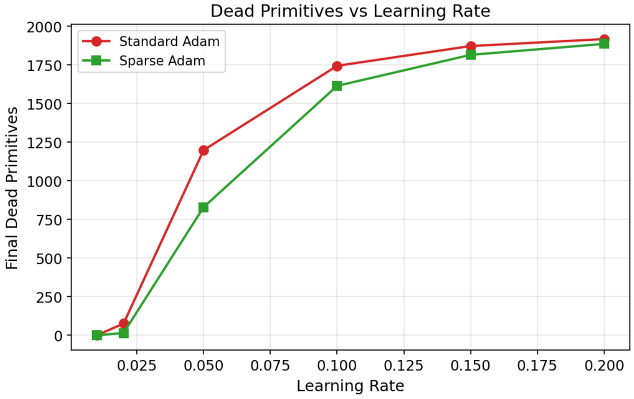

Figure 7: Dead primitives vs. learning rate. The gap between Adam and Sparse Adam is largest at moderate learning rates.

7 DISCUSSION

7.1 Mechanism of Implicit Death Acceleration

Our results provide strong evidence for the conjectured mechanism. When a visible-step gradient pushes a primitive toward lower opacity (positive gradient in raw space), the subsequent implicit updates continue this downward push during invisible steps. The decay ratio of 0.9005 per step means that after the half-life of 6.61 steps, the implicit update still retains half its initial magnitude—sufficient to push borderline primitives past the death threshold.

7.2 Regime Dependence

The gradient bias sweep reveals that the conjecture holds most strongly in the moderate-redundancy regime. When redundancy pressure is too weak ($\mu_g \to 0$), implicit updates are small and few primitives approach the death threshold. When pressure is too strong ($\mu_g \gg 0$), explicit gradients alone suffice to kill all primitives, and the implicit contribution is masked. This regime dependence has practical implications: real 3DGS scenes likely operate in the moderate regime where some but not all primitives are redundant.

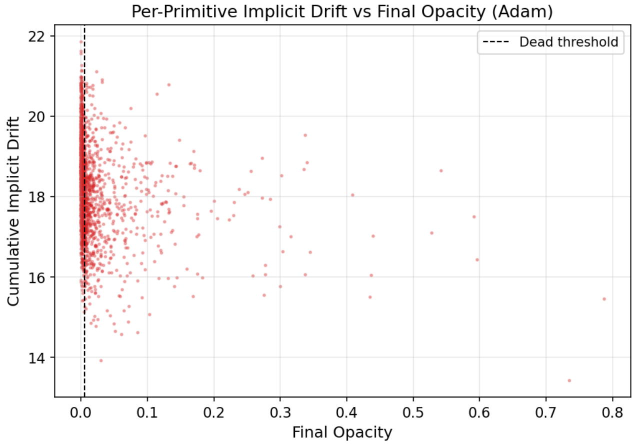

Figure 8: Per-primitive scatter of cumulative implicit drift vs. final opacity under Adam. The dead threshold is shown as a dashed line.

7.3 Implications for Optimizer Design

Our findings suggest that Sparse Adam is preferable for 3DGS opacity optimization, as it avoids the pathological implicit drift without sacrificing convergence for visible primitives. More generally, any adaptive optimizer applied to structurally sparse gradient settings should consider whether implicit updates from momentum decay are desirable.

7.4 Limitations

Our simulation abstracts several aspects of real 3DGS training: (1) we model gradients as i.i.d. draws rather than from a differentiable renderer; (2) we omit densification and periodic opacity resets; (3) our visibility model is simplified relative to actual ray–Gaussian intersection. These simplifications enable controlled experimentation but limit direct quantitative transfer to production 3DGS pipelines.

8 CONCLUSION

We investigated the conjecture that implicit Adam updates accelerate dead primitive formation in 3D Gaussian Splatting. Through analytical derivation and stochastic simulation, we showed that standard Adam produces 59.95% dead primitives compared to 41.5% for Sparse Adam, with a robust difference of 386.3 ± 37.1 across 10 seeds. We identified the decay ratio of 0.9005 per invisible step, a half-life of 6.61 steps, and a positive feedback loop linking opacity, visibility, and implicit drift. These results support the conjecture and motivate the adoption of sparse optimization strategies for point-based neural rendering.

REFERENCES

[1] Zhiyuan Chen et al. 2024. The Implicit Bias of Adam on Separable Data. *arXiv preprint arXiv:2511.02773* (2024).

[2] Yizhou Ding et al. 2026. A Step to Decouple Optimization in 3DGS. In *arXiv preprint arXiv:2601.16736*. Appendix: Discussion Related to Implicit Update.

[3] Keller Jordan et al. 2024. Muon: An Optimizer for Hidden Layers in Neural Networks. *arXiv preprint arXiv:2510.15366* (2024).

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023), 1–14.

[5] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (2015).