BLUEPRINT-GS: PERSISTENT REPRESENTATION DEMAND FOR ADAPTIVE 3D GAUSSIAN GROWTH

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) reconstructs photorealistic scenes that can be rendered from arbitrary viewpoints. It attains this fidelity by progressively densifying Gaussian primitives wherever reconstruction remains imperfect. However, such densification commonly produces millions of primitives for complex scenes, straining memory, storage, and rendering. This redundancy persists because the signals that trigger growth are ambiguous: a large error may indicate missing representation or merely unconverged Gaussians, and a genuinely unrepresented region may elicit no response at all. Densification therefore adds Gaussians where none are needed and compounds the redundancy by splitting them further. To suppress the redundancy, we present **BluePrint-GS**, a demand-driven framework that decides where to place new Gaussian primitives from the state of the scene rather than the state of individual primitives. BluePrint-GS maintains *representation demand*, a persistent world-space state that records where representation remains insufficient. Particularly, BluePrint-GS determines where additional representation is needed, constructs new Gaussian geometry from the spatial structure of local demand, and quantifies how much to introduce through conserved representation intensity, so that growth follows demand rather than a prescribed target population, and redundancy is suppressed. Experimental results demonstrate that, across 13 scenes from Mip-NeRF 360, Deep Blending, and Tanks and Temples, BluePrint-GS reduces the Gaussian population by 86.0%, 90.0%, and 87.7%, respectively, relative to vanilla 3DGS, with average PSNR losses of at most 0.45 dB. These results demonstrate a competitive quality-compactness trade-off for demand-guided Gaussian growth.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) Kerbl et al. (2023) has emerged as an effective representation for photorealistic novel-view synthesis, combining high rendering quality with real-time rasterization. Compared with NeRF-style methods Mildenhall et al. (2021), which typically require repeated neural-field evaluations along camera rays, 3DGS represents a scene explicitly using anisotropic Gaussian primitives and supports efficient differentiable rendering.

High-quality reconstruction often relies on progressive densification and may produce millions of Gaussians. Compact representation methods reduce the resulting costs through pruning Liu et al. (2024); Lee et al. (2024); Hanson et al. (2025); Lin et al. (2025), quantization Papantonakis et al. (2024), attribute or context encoding Chen et al. (2024; 2025); Girish et al. (2024); Shin et al. (2025), and hierarchical organization Kerbl et al. (2024). These mechanisms may operate during reconstruction or on an existing representation. Density-control methods directly regulate growth by selecting Gaussians to split, remove, or relocate Kheradmand et al. (2024); Ye et al. (2024); Mallick et al. (2024); Wang et al. (2025). Both lines of work seek favorable trade-offs between representation cost and rendering quality.

One difficulty in controlling growth is *signal ambiguity*. A large reconstruction residual may indicate insufficient representation, but it may also decrease through further optimization of existing Gaussians. Conversely, poorly covered regions may produce weak signals on individual primitives. As Gaussians move, split, or are pruned, maintaining the history of a spatial region through primitive-level statistics also becomes less direct. These observations motivate a persistent spatial assessment

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.

of where added capacity is worth testing, without assuming that residual magnitude alone establishes a need for new geometry.

We introduce **BluePrint-GS**, which maintains this assessment as *representation demand* in fixed world-space cells. Covered residuals are weighted by their lack of improvement and the measured response of their contributing Gaussians to optimization. A separate Null channel attributes potentially uncovered evidence using depth support from other views. This channel can propose additions without a designated parent primitive, but depends on cross-view support from the current representation. The resulting demand state prioritizes local capacity trials.

BluePrint-GS couples this state to candidate construction and growth accounting. Candidate fitting and an incremental training-view gain test then assess the proposed addition. Particularly, BluePrint-GS resolves the ambiguity by attributing rendering evidence to world-space cells rather than to primitives, so that regions covered by no Gaussian remain observable, and by admitting that evidence as demand only once it persists and the primitives responsible for it cease to respond to optimization. Specifically, the spatial distribution of local demand determines the center, scale, and orientation of each new Gaussian through its first two moments, while accumulated demand is conserved under spatial aggregation and converted into discrete growth one primitive at a time. The demand field is discarded after training, leaving a standard 3DGS representation.

We evaluate BluePrint-GS on 13 scenes from Mip-NeRF 360 Barron et al. (2022), Deep Blending Hedman et al. (2018), and Tanks and Temples Knapitsch et al. (2017), against vanilla 3DGS and seven compact or density-control methods. Average Gaussian counts decrease from 3.87M to 0.54M, from 2.31M to 0.23M, and from 2.04M to 0.25M, respectively, with dataset-average PSNR losses of at most 0.45 dB. On Mip-NeRF 360, BluePrint-GS uses 19.4% and 34.1% fewer Gaussians than OMG Lee et al. (2026) and Taming-3DGS Mallick et al. (2024) at higher PSNR. On Deep Blending, it has the smallest reported population and the highest SSIM among the compared methods.

Our contributions are as follows:

- **New Problem: Signal ambiguity in density control.** We identify *signal ambiguity* as the root of the redundancy–quality trade-off: a large error indicates either unconverged Gaussians or missing ones, so growth cannot be contained without sacrificing quality.
- **New Method: BluePrint-GS.** We design BluePrint-GS, which replaces primitive selection with *placement*: it maintains *representation demand* as a persistent world-space state, derives new Gaussian geometry from local demand, and grows without a prescribed target population.
- **Comprehensive Evaluation.** We evaluate BluePrint-GS against vanilla 3DGS and seven compact and density-control methods on 13 scenes from three benchmarks. Under a single configuration, it reduces the population by 86.0%, 90.0%, and 87.7% at a quality loss of at most 0.45 dB PSNR.

2 Related Work

3D Gaussian Splatting. NeRF Mildenhall et al. (2021) achieves photorealistic novel-view synthesis through differentiable volume rendering, but its dense ray sampling limits real-time performance; subsequent works improve efficiency through faster neural encodings Müller et al. (2022) and explicit scene representations Fridovich-Keil et al. (2022). 3DGS Kerbl et al. (2023) instead represents a scene with explicit anisotropic Gaussian primitives and renders them by differentiable rasterization, attaining high quality at real-time frame rates. Following 3DGS, extensive studies refine its parameterization, optimization, and rasterization Radl et al. (2024); Wang et al. (2024); Ren et al. (2026); Goli et al. (2024), and extend it to anchor-based Lu et al. (2024), hierarchical Kerbl et al. (2024), and resource-constrained Du et al. (2026) settings. More recently, Gaussian modeling has begun to move beyond information attached to individual primitives: Eulerian Splatting Polansky et al. (2026) optimizes a persistent volumetric probability distribution from which Gaussian locations are sampled, demonstrating that spatial information need not remain tied to primitive identities. BluePrint-GS uses a spatial field for a different role: prioritizing candidate additions from reconstruction evidence, with separate geometry construction and growth accounting.

Compression of Gaussian Representations. As Gaussian populations grow, a large body of work compresses the resulting representation. Pruning approaches discard primitives with limited ren-

Table 1: Comparison of representative methods for reducing 3D Gaussian representations.

Method	Scope		Decision Basis				Realization			Output
	Stage	Reduces	Signal	Domain	Memory	Null	Action	Geometry	w/o Budget	Std. 3DGS
Compression-oriented										
LightGaussian	Post-hoc	Count+Size	Significance	Prim.	–	–	Prune/Quant.	–	×	Partial
OMG	Training	Count+Size	Redundancy	Prim.	–	–	Prune/Encode	–	✓	Partial
HAC++	Training	Count+Size	Context	Anchor	–	–	Mask/Entropy	–	✓	×
LocoGS	Training	Count+Size	Locality	Prim.	–	–	Mask/Encode	–	✓	×
Selection-based density control										
3DGS	Growth	Count	Gradient	Prim.	×	×	Clone/Split	Parent	✓	✓
AbsGS	Growth	Count	Abs. gradient	Prim.	×	×	Split	Parent	✓	✓
3DGS-MCMC	Growth	Count	Opacity	Prim.	×	×	Relocate	Parent	×	✓
Taming-3DGS	Growth	Count	Score	Prim.	×	×	Top-k	Parent	×	✓
MaskGaussian	Growth	Count	Mask	Prim.	×	×	Prune	–	✓	✓
SteepGS	Growth	Count	Curvature	Prim.	×	×	Split	Parent	✓	✓
Placement-based density control (ours)										
BluePrint-GS	Growth	Count	Rendering	**Scene**	✓	✓	**Place**	**Demand**	✓	✓

[1] Prim.: decisions are made over individual primitives. – indicates not applicable.
[2] *Memory*: persistent scene-space evidence used for growth, beyond ordinary primitive-level gradient accumulation. *Null*: depth-supported attribution without a contributing primitive on the sampled ray; unsupported surfaces may remain unobserved.

dering contribution, including learned compact primitive sets Liu et al. (2024); Lee et al. (2024), sensitivity-based pruning Hanson et al. (2025), and pruning under real-time budgets Lin et al. (2025). Attribute-oriented approaches instead reduce the cost of each primitive through quantization Papantonakis et al. (2024), lightweight encodings Girish et al. (2024), joint pruning and distillation Fan et al. (2024), and joint sparsity–compressibility optimization Ali et al. (2025). Context-based coders push storage further: hash-grid assisted context modeling Chen et al. (2024; 2025) and locality-aware attribute coding Shin et al. (2025) substantially reduce model size, while OMG Lee et al. (2026) jointly targets a compact primitive set and attribute representation. These methods include both post-hoc compression and training-time optimization: HAC++ incorporates rate-aware masking, while LocoGS learns compact attributes with masking during reconstruction. BluePrint-GS focuses on candidate placement and growth control while retaining standard Gaussian attributes.

Density Control for Gaussian Growth. Density control instead regulates how many primitives are created during reconstruction. The adaptive density control of 3DGS Kerbl et al. (2023) clones and splits primitives whose accumulated view-space gradient exceeds a threshold, a criterion refined by AbsGS Ye et al. (2024) through absolute gradients and by SteepGS Wang et al. (2025) through the local curvature of the loss. Others reformulate the update itself: 3DGS-MCMC Kheradmand et al. (2024) relocates low-opacity Gaussians as samples of a distribution, MaskGaussian Liu et al. (2025) models primitives as probabilistic masks, and DropGaussian Park et al. (2025) regularizes growth by random dropout. Taming-3DGS Mallick et al. (2024) instead ranks candidates under a prescribed budget, and further works revisit refinement criteria and training efficiency Rota Bulò et al. (2024); Deng et al. (2024); Kim et al. (2026). SteepGS explicitly studies when splitting can improve optimization beyond a stationary primitive configuration. Table 1 summarizes the decision mechanisms considered here. BluePrint-GS complements primitive-level control with persistent spatial evidence used to construct and validate new candidates.

3 Problem Formulation

Problem Setting. Selection-based density control decides growth over the current Gaussian population, so its signal is ambiguous: a large residual may indicate that a region lacks representation, or merely that its Gaussians have not yet converged. Moreover, the primitive carrying that signal may itself be split or pruned, so no evidence accumulates for the region it covers. We therefore formulate density control around an explicit *representation demand* maintained in the scene rather than on primitives, and decompose it into three questions.

1. **RQ1: Demand Identification.** How can the signals be converted into a persistent world-space demand that identifies both where representation is needed and whether it is genuinely required?
2. **RQ2: Demand Realization.** How should the demand be instantiated as new Gaussians, from the spatial structure of the under-represented region rather than from a parent primitive?
3. **RQ3: Demand Quantification.** How should the magnitude of accumulated local demand determine how much representation is added?

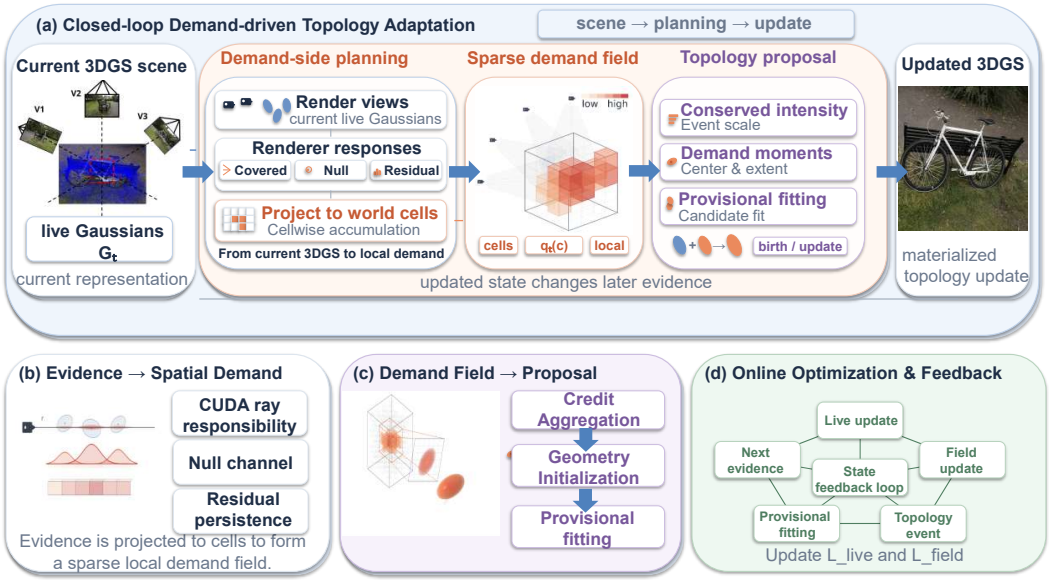


Figure 1: Overview of **BluePrint-GS**. (a) The overall pipeline links spatial demand with Gaussian growth. (b) The evidence-to-demand component in (a) localizes and maintains demand. (c) The demand-to-proposal component determines growth requests and geometry. (d) Sequential optimization supplies feedback across the pipeline.

In general, these questions define a placement process that first identifies unresolved representation, then determines how it should be realized, and finally controls how much capacity is added.

4 METHOD

4.1 OVERVIEW

BluePrint-GS maintains spatial demand before deciding how and how much to grow. In Fig. 1(a), renderer and optimizer responses identify unresolved representation, whose spatial structure and accumulated magnitude guide new Gaussians. Accepted proposals change the live representation and the evidence used to estimate subsequent demand, closing the loop. Following Sec. 3, Fig. 1(b) addresses **RQ1** (Sec. 4.2); Fig. 1(c) couples **RQ2** construction (Sec. 4.3) with **RQ3** quantification (Sec. 4.4). Their online interaction is shown in Fig. 1(d) and described in Sec. 4.5.

4.2 SPATIAL DEMAND ESTIMATION

For **RQ1**, image error neither uniquely locates its source nor establishes a need for more Gaussians. We separate spatial attribution from demand assessment, then retain that assessment across observations. Let $\mathcal{G}$ be the live Gaussian set, $\mathcal{V}$ the calibrated training views, and t the current evidence exposure.

Localizing reconstruction evidence. Primitive-level signals can lose their history as Gaussians move. We instead associate evidence with fixed world-space cells of width h. For sampled ray r and depth-ordered Gaussian i, the compositing weight $w_{ri} = T_{ri}\widetilde{\alpha}_{ri}$ combines incoming transmittance T_{ri} and pixel-dependent opacity $\widetilde{\alpha}_{ri}$. It supplies *Covered* evidence; terminal transmittance T_r^{end} supplies potential *Null* evidence. For the ray-distance interval $[a_{rc}, b_{rc}]$ intersecting cell c, assign

$$
\begin{aligned}
m_{ric}^{\text{cov}} &= w_{ri} \int_{a_{rc}}^{b_{rc}} k_{ri}(s)\,\mathrm{d}s, \\
m_{rc}^{\text{null}} &= T_r^{\text{end}} V_r^{\text{null}} \int_{a_{rc}}^{b_{rc}} q_r(s)\,\mathrm{d}s.
\end{aligned}
\tag{1}
$$

Here s is distance along the unit ray and k_{ri} is its restricted Gaussian density with finite query support. Null has no contributing primitive to locate it, so its density q_r uses depth hypotheses rendered by the current Gaussians in other views. The binary gate V_r^{null} requires multi-view depth support; unsupported locations receive no Null evidence. Appendix A.1 defines the kernels and finite query intervals.

Interpreting localized error. Depth-supported Null evidence suggests insufficient coverage, whereas Covered error may still decrease through optimization. We therefore retain their residuals separately. Let $\mathcal{R}_t$ be the current view's sampled rays, $\nu_r \geq 0$ their weights with $\sum_{r\in\mathcal{R}_t} \nu_r = 1$, and $m_{rc}^{\text{cov}} = \sum_i m_{ric}^{\text{cov}}$. For rendered and target RGB values $\widehat{\mathbf{I}}(r)$ and $\mathbf{I}^{\text{gt}}(r)$, use $e_r = \min\{1, \|\widehat{\mathbf{I}}(r) - \mathbf{I}^{\text{gt}}(r)\|_1/3\}$ and

$$M_t^b(c) = \sum_{r\in\mathcal{R}_t} \nu_r m_{rc}^b, \quad R_t^b(c) = \sum_{r\in\mathcal{R}_t} \nu_r e_r m_{rc}^b, \quad M_t(c) = \sum_b M_t^b(c). \tag{2}$$

The branch index is $b \in \{\text{cov}, \text{null}\}$; M_t^b measures support and R_t^b its residual-weighted mass. Covered error receives greater priority when it fails to improve and its contributing Gaussians respond weakly to optimization. A same-view non-improvement score $P_t(c)$, an optimizer-response score $S_t(c)$, and binary history validity $V_t^{\text{cov}}(c)$ yield

$$D_t(c) = \frac{R_t^{\text{null}}(c) + V_t^{\text{cov}}(c) P_t(c) S_t(c) R_t^{\text{cov}}(c)}{M_t(c) + \epsilon}. \tag{3}$$

The numerical safeguard $\epsilon > 0$ and gates in $[0, 1]$ give $D_t(c) \in [0, 1]$. This score prioritizes capacity trials; it does not certify that further optimization is ineffective. Gate definitions appear in Appendix A.2.

Maintaining persistent demand. View-dependent evidence should revise a region's assessment rather than replace its history. Each actionable cell stores a logit $z_t(c) \in \mathbb{R}$ and demand $d_t(c) = \sigma(z_t(c))$, where $\sigma(z) = (1 + e^{-z})^{-1}$. The field labeled $q_t(c)$ in Fig. 1 denotes $d_t(c)$. For supported cells $\mathcal{C}_t$, normalize exposure weights as $\omega_t(c) = M_t(c)/\sum_{c'\in\mathcal{C}_t} M_t(c')$ and take finite updates on

$$\mathcal{L}_{\text{demand}} = \sum_{c\in\mathcal{C}_t} \omega_t(c) \left[\ell_{\text{BCE}}(d_t(c), D_t(c)) + \lambda_{\text{temp}}(d_t(c) - d_{t^-}(c))^2\right]. \tag{4}$$

Here $\ell_{\text{BCE}}(d, D) = -D \log d - (1 - D) \log(1 - d)$ is binary cross-entropy with a soft target. It fits the demand state to detached evidence, while $\lambda_{\text{temp}} \geq 0$ anchors it to its pre-exposure value d_{t^-}. Denote the weighted BCE and squared-change sums by $\mathcal{L}_{\text{BCE}}$ and $\mathcal{L}_{\text{temp}}$, respectively, excluding the latter's coefficient. Targets, weights, and d_{t^-} remain fixed during the inner steps. Memory depends on both the temporal penalty and the finite updates; demand is a growth priority, not a calibrated probability of missing geometry (Appendix A.2).

4.3 Demand-Conditioned Gaussian Construction

Demand identifies where to consider expansion, but nearby primitives need not match its spatial structure. For **RQ2**, we derive candidate geometry from demand, then test its incremental rendering benefit.

Demand-derived geometry. Combining unrelated modes could place a candidate between surfaces. We partition eligible cells into bounded, connected, depth-supported regions $\mathcal{A}_j$, indexed by j. Their unspent credits $a_t(c)$, introduced in Sec. 4.4, define $\eta_c = a_t(c)/\sum_{c'\in\mathcal{A}_j} a_t(c')$. For cell centers $\mathbf{x}_c$ and within-cell covariance $\Sigma_c = (h^2/12)\mathbf{I}_3$, initialize

$$\begin{aligned} \boldsymbol{\mu}_j^D &= \sum_{c\in\mathcal{A}_j} \eta_c \mathbf{x}_c, \\ \boldsymbol{\Sigma}_j^D &= \sum_{c\in\mathcal{A}_j} \eta_c \left[(\mathbf{x}_c - \boldsymbol{\mu}_j^D)(\mathbf{x}_c - \boldsymbol{\mu}_j^D)^\top + \Sigma_c\right]. \end{aligned} \tag{5}$$

Here $\mathbf{I}_3$ is the three-dimensional identity and superscript D marks demand-derived initialization. These moments transfer regional position, extent, and orientation to the candidate; bounded scales and a spatial trust region preserve this association during fitting. The depth support required for both branches is specified in Appendix A.3.

From spatial alignment to useful capacity. An aligned candidate must still explain observations alongside the live Gaussians. Freeze $\mathcal{G}$ and fit candidate g on its supporting views $\mathcal{V}_j \subseteq \mathcal{V}$:

$$\mathcal{L}_{\text{prov}}(g) = \mathcal{L}_{\mathcal{V}_j}^{\text{photo}}(\mathcal{G} \cup \{g\}) + \lambda_{\text{op}} \log(1 + 255\alpha_g). \tag{6}$$

The physical opacity is $\alpha_g \in (0, 1)$ and $\lambda_{\text{op}} \geq 0$ penalizes opacity. For any nonempty view set $\mathcal{S}$, $\mathcal{L}_{\mathcal{S}}^{\text{photo}}$ averages the RGB L_1 and structural dissimilarity loss over its views (Appendix A.3). Local improvement may conceal damage elsewhere, so acceptance requires a global training-view gain

$$\Delta_j = \mathcal{L}_{\mathcal{V}}^{\text{photo}}(\mathcal{G}) - \mathcal{L}_{\mathcal{V}}^{\text{photo}}(\mathcal{G} \cup \{g\}) > \tau_{\text{gain}} > 0. \tag{7}$$

The threshold τ_{gain} sets the minimum accepted gain against the same frozen live set. Multi-view contribution and spatial constraints must also hold. A useful candidate can be committed only if its region has enough accumulated credit, as defined next.

4.4 Conserved Representation Intensity

Construction determines how to realize demand, but a bounded score does not specify a growth allowance. For **RQ3**, supported demand accumulates as spendable intensity, and each accepted addition consumes one unit.

From persistent demand to intensity. Clip demand to $d_t^{\text{clip}}(c) = \min\{d_t(c), 1 - \epsilon_d\}$ and update the cell's credit for each new actionable exposure:

$$\rho_t(c) = \left[\log \frac{1 - \tau_d}{1 - d_t^{\text{clip}}(c)} \right]_+, \tag{8}$$
$$a_t^+(c) = e^{-(\tau_t - \tau_c^-)/\tau_a} a_{t^-}(c) + \kappa \Delta\tau_t M_t(c) \rho_t(c).$$

Here $[x]_+ = \max(x, 0)$, τ_d is a demand dead zone, $\epsilon_d > 0$ prevents a logarithmic singularity, $\kappa > 0$ sets the credit rate, and $\tau_a > 0$ is its decay lifetime. Time τ_t is measured in camera sweeps; τ_c^- records the cell's last ledger update and $\Delta\tau_t = 1/|\mathcal{V}|$ is one new exposure. The nonnegative balance $a_{t^-}(c)$ becomes $a_t^+(c)$ after decay and injection, before any debit. Unobserved cells receive decay only (Appendix A.4).

From accumulated intensity to additions. To let neighboring fractional credits jointly authorize a trial, pool current balances over each eligible region:

$$A_t(\mathcal{A}_j) = \sum_{c \in \mathcal{A}_j} a_t(c), \qquad K_t(\mathcal{A}_j) = \lfloor A_t(\mathcal{A}_j) \rfloor. \tag{9}$$

Here $a_t(c)$ is the balance after all preceding events, including any debit. K_t is available whole-unit credit: $K_t \geq 1$ authorizes at most one candidate per region per round, not K_t immediate births. Acceptance consumes one unit; rejection consumes none. Further trials require fresh evidence, and stale credit decays or expires. A unique base-cell ledger prevents regrouping from duplicating credit. Conservation applies to this ledger, including its sources and sinks (Appendix A.4).

4.5 Online Optimization and Feedback

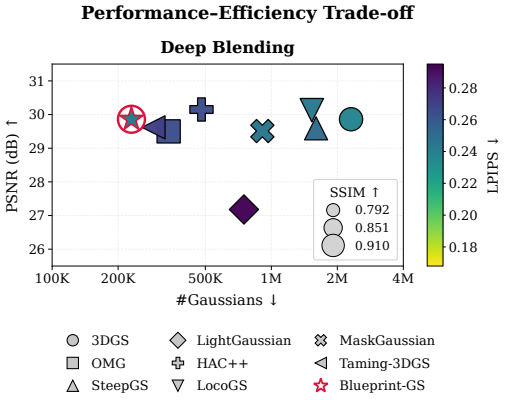
Growth changes the evidence that motivates it. We therefore alternate reconstruction, demand estimation, and candidate fitting through three objectives, each updating only its first argument:

$$\begin{aligned} \mathcal{L}_{\text{rec}}(\mathcal{G}) &= \mathcal{L}_{\{v\}}^{\text{photo}}(\mathcal{G}), \\ \mathcal{L}_{\text{demand}}(z; \bar{\mathcal{E}}_t) &= \mathcal{L}_{\text{BCE}}(z; \bar{\mathcal{E}}_t) + \lambda_{\text{temp}} \mathcal{L}_{\text{temp}}(z; \bar{\mathcal{E}}_t), \\ \mathcal{L}_{\text{prov}}(g; \bar{\mathcal{G}}) &= \mathcal{L}_{\mathcal{V}_j}^{\text{photo}}(\bar{\mathcal{G}} \cup \{g\}) + \lambda_{\text{op}} \log(1 + 255\alpha_g). \end{aligned} \tag{10}$$

Here v is the current view and z collects active demand logits. Bars in this equation denote detached, fixed inputs: $\mathcal{E}_t = (\mathcal{C}_t, \omega_t, D_t, d_{t^-})$ and the live set $\mathcal{G}$. The figure labels $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{demand}}$ as $\mathcal{L}_{\text{live}}$ and $\mathcal{L}_{\text{field}}$, respectively. Optimizing a Gaussian set means updating its continuous parameters with topology fixed. Each live step refreshes demand; at sweep boundaries, credit activates sequentially verified trials. Acceptance consumes credit and updates the live set; only real exposures inject credit (Algorithm 1). Inference retains standard Gaussians.

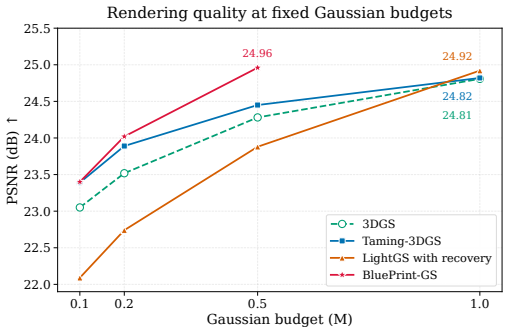


(a) Comparison of images rendered using different algorithms.

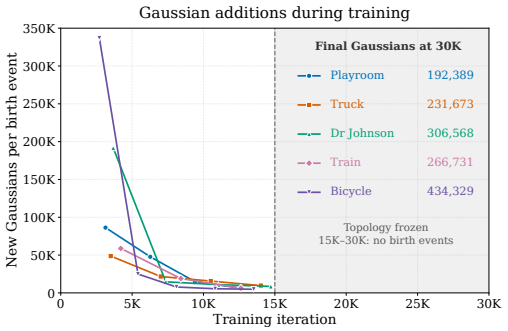


(b) Novel-view PSNR versus Gaussian population on the Deep Blending dataset.

Figure 2: Qualitative and quantitative comparisons. (a) Example renderings with the reported PSNR, model size, and Gaussian count. (b) Dataset-average PSNR versus Gaussian count on Deep Blending; marker size and color indicate SSIM and LPIPS.



(a) Rendering quality at the evaluated Gaussian budgets after 30K training iterations.



(b) Gaussian additions across five scenes; topology is frozen from 15K to 30K iterations.

Figure 3: Quality–compactness and growth analysis. (a) PSNR at the evaluated Gaussian budgets. (b) Additions during adaptive growth and the subsequent scheduled topology freeze.

5 Experiments

5.1 Experimental Setup

Datasets and metrics. We evaluate on 13 scenes from Mip-NeRF 360 Barron et al. (2022), Deep Blending Hedman et al. (2018), and Tanks and Temples Knapitsch et al. (2017). We report PSNR, SSIM, and LPIPS for rendering quality, and Gaussian count (#GS), model size, training time, and peak training memory for efficiency.

Baselines. We compare with 3DGS Kerbl et al. (2023) and seven compact or density-control methods: OMG Lee et al. (2026), SteepGS Wang et al. (2025), LightGaussian Fan et al. (2024), HAC++ Chen et al. (2025), LocoGS Shin et al. (2025), MaskGaussian Liu et al. (2025), and Taming-3DGS Mallick et al. (2024).

Implementation details. BluePrint-GS trains for 30K iterations on a single NVIDIA V100 GPU, replacing native 3DGS densification with the controller in Sec. 4. Initialization uses each scene's COLMAP reconstruction, consistent with the baselines. Unless otherwise specified, one configuration is used across scenes. The demand field is discarded after training, retaining compatibility

Table 2: Quantitative results on three benchmarks. #GS is the average Gaussian count in millions. Best and second-best results are **bold** and underlined, respectively.

Method	Mip-NeRF 360				Deep Blending				Tanks & Temples			
	PSNR↑	SSIM↑	LPIPS↓	#GS(M)↓	PSNR↑	SSIM↑	LPIPS↓	#GS(M)↓	PSNR↑	SSIM↑	LPIPS↓	#GS(M)↓
3DGS	**27.54**	0.813	**0.217**	3.87	29.86	0.906	**0.2378**	2.31	23.77	**0.852**	**0.168**	2.04
OMG	26.79	0.806	0.247	0.67	29.50	0.906	0.265	0.34	23.24	0.833	0.205	**0.24**
SteepGS	27.07	0.798	0.248	2.09	29.59	0.903	0.251	1.60	23.37	0.837	0.193	1.30
LightGaussian	27.36	**0.816**	0.224	1.07	27.18	0.870	0.295	0.75	23.07	0.820	0.223	0.62
HAC++	27.52	0.802	0.252	1.16	**30.15**	0.906	0.265	0.48	**24.13**	0.848	0.190	0.51
LocoGS	27.34	0.807	0.235	1.43	30.13	0.909	0.244	1.53	23.88	0.850	0.178	1.03
MaskGaussian	27.21	0.811	0.229	1.57	29.51	0.906	0.245	0.91	23.71	0.846	0.180	0.74
Taming-3DGS	27.16	0.792	0.263	0.82	29.62	0.904	0.271	0.29	23.53	0.835	0.210	0.31
BluePrint-GS	27.24	0.813	0.226	**0.54**	29.86	**0.910**	0.241	**0.24**	23.32	0.839	0.191	0.25

Table 3: Representation and computational efficiency on Mip-NeRF 360.

Method	Size(MB)↓	Time(min)↓	Mem.(GB)↓
3DGS	625.93	31.1	8.54
SteepGS	494.67	24.283	8.9
MaskGaussian	356.83	28.303	8.5
Taming-3DGS	164.54	12.745	8.7
BluePrint-GS	101.3	28.59	10.6

Table 4: Ablations of BluePrint-GS on the Bicycle scene.

Variant	PSNR↑	SSIM↑	LPIPS↓	#GS↓
Residual-only	23.81	0.632	0.422	312553
Primitive state	24.29	0.657	0.384	427829
Parent geometry	24.53	0.648	0.332	431487
Non-conserved	24.77	0.687	0.283	843756
Full model	24.96	0.709	0.271	434329

with standard 3DGS rendering. Appendix A specifies the state updates and event ordering. Table 5 distinguishes fixed constants from run-configuration inputs.

5.2 Comparison with State-of-the-Art Methods

Novel-view quality and Gaussian population. Our central question is whether growth can be restricted while retaining the capacity needed for reconstruction. Table 2 shows that BluePrint-GS reduces Gaussian populations by 86.0%, 90.0%, and 87.7% relative to 3DGS on Mip-NeRF 360, Deep Blending, and Tanks and Temples, respectively, with dataset-average PSNR losses of at most 0.45 dB. The corresponding SSIM and LPIPS values show the accompanying quality trade-offs. These results support the complete pipeline as a compact representation strategy on the evaluated benchmarks.

Comparison with compact density-control methods. On Mip-NeRF 360, BluePrint-GS uses 19.4% and 34.1% fewer Gaussians than OMG and Taming-3DGS while achieving higher PSNR. On Deep Blending, it combines the smallest population with the highest SSIM. On Tanks and Temples, the trade-off is mixed: compared with Taming-3DGS, BluePrint-GS uses fewer Gaussians and improves SSIM and LPIPS, while PSNR is 0.21 dB lower. These results support the complete pipeline in the low-population regime; Sec. 5.4 examines selected components on Bicycle.

Training and representation efficiency. The compact final model must also be assessed against the cost of maintaining demand and evaluating growth proposals. Table 3 shows that, on Mip-NeRF 360, BluePrint-GS reduces model size from 625.93 to 101.3 MB and construction time from 31.1 to 28.59 minutes relative to 3DGS, while increasing peak training memory from 8.54 to 10.6 GB. These values correspond to an 83.8% reduction in model size and an 8.1% reduction in training time, with a 24.1% increase in peak training memory. The demand field is discarded after reconstruction, so its storage and computation are confined to training, and the resulting model retains standard 3DGS rendering compatibility.

5.3 Analysis of Representation Demand

Quality–compactness trade-off. Figure 3(a) compares rendering quality across the evaluated Gaussian budgets. BluePrint-GS closely matches Taming-3DGS at 100K, with PSNR advantages of 0.13 and 0.51 dB at 200K and 500K, respectively. Its 500K-budget result is 24.96 dB, com-

pared with 24.82 dB for Taming-3DGS and 24.92 dB for LightGS with recovery both evaluated at a 1M budget. These operating points support a favorable quality–compactness trade-off within the evaluated range.

Growth dynamics. Figure 3(b) tracks newly added Gaussians per birth event across five scenes. Additions decrease before topology is frozen at 15K iterations. The existing Gaussians are then optimized until 30K without further additions. This curve characterizes growth under the training schedule. The declining trend before topology freezing suggests that representation demand is gradually subsiding, but does not establish that the underlying need for additional capacity has been fully met.

5.4 Ablation Study

The ablations in Table 4 follow the three research questions introduced in Sec. 3. For **RQ1**, *Residual-only* removes temporal persistence and optimizer-response evidence, while *Primitive state* replaces the persistent world-space demand state with decisions attached to the current Gaussians. For **RQ2**, *Parent geometry* replaces demand-conditioned construction with geometry inherited from an existing Gaussian. For **RQ3**, *Non-conserved* removes the aggregation of representation intensity before discretization. These variants examine demand identification, demand-conditioned realization, and demand quantification, respectively.

Residual-only reduces PSNR by 1.15 dB with 28% fewer Gaussians, suggesting that instantaneous residuals alone provide insufficient guidance for allocating capacity to unresolved representation needs. Primitive state reduces PSNR by 0.67 dB despite using 1.5% fewer Gaussians, supporting the separation of demand state from the evolving primitive configuration so that demand remains anchored to the spatial regions requiring representation. Parent geometry reduces PSNR by 0.43 dB with only a 0.7% decrease in Gaussian count, indicating that demand provides useful information about not only where capacity is needed but also what geometry new Gaussians should adopt. Non-conserved increases the Gaussian count to 1.94 times that of the full model while degrading all three quality metrics, supporting the accumulation of fractional demand before discretization to limit redundant growth. Together, these results support persistent, representation-independent demand identification, demand-conditioned geometry, and conservation of demand during quantification.

5.5 Limitations

Our current implementation builds on standard 3DGS, leaving room to explore more expressive representations for improved reconstruction quality. Storage savings primarily come from reducing the Gaussian population; attribute quantization and entropy coding have not yet been explored. Combining demand-driven growth with these techniques may further reduce storage costs, although their integration and the resulting quality–storage trade-off remain to be evaluated.

6 Conclusion

We introduced BluePrint-GS, a demand-driven framework that mitigates ambiguous growth decisions through persistent world-space representation demand. Local demand guides Gaussian placement and geometry, while conserved intensity regulates additions without a prescribed population target. Across 13 scenes from three benchmarks, BluePrint-GS reduces Gaussian populations by 86.0%–90.0% relative to 3DGS, with dataset-average PSNR losses of at most 0.45 dB. These results support controlling redundancy during representation formation while preserving compatibility with standard 3DGS rendering.

Ethics Statement

This work studies compact 3D scene reconstruction and novel-view synthesis using the publicly available Mip-NeRF 360, Deep Blending, and Tanks and Temples benchmarks. Our experiments use existing scene imagery and do not involve human-subject studies or animal experimentation. These datasets remain subject to the usage terms specified by their original providers.

Reproducibility Statement

The main paper describes the formulation and optimization procedure of BluePrint-GS. Appendix A provides additional details on demand estimation, candidate initialization and fitting, acceptance criteria, and credit accounting, together with pseudocode for the optimization procedure. Section 5 reports the datasets, evaluation metrics, training duration, and hardware. We use a common configuration across scenes, with initialization obtained from each scene's COLMAP reconstruction. The three evaluation benchmarks are publicly available from their original providers.

AI Use Statement

In this work, we used generative AI tools to assist in developing key steps in mathematical proofs and drafting the proofs, and to provide feedback on experimental design and refinements. We also used these tools to support translation, manuscript drafting and language editing, the design of code-level implementation details, and the summarization and analysis of related literature. The authors take full responsibility for the final content of this work, including all mathematical claims and proofs, methods, experimental results, code, and references.

References

Muhammad Salman Ali, Sung-Ho Bae, and Enzo Tartaglione. Elmgs: Enhancing memory and computation scalability through compression for 3d gaussian splatting. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2591–2600. IEEE, 2025.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469. IEEE, 2022.

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, pp. 422–438. Springer, 2024.

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac++: Towards 100x compression of 3d gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Xiaobin Deng, Changyu Diao, Min Li, Ruohan Yu, and Duanqing Xu. Efficient density control for 3d gaussian splatting. *arXiv preprint arXiv:2411.10133*, 2024.

Xiaobiao Du, Yida Wang, Kun Zhan, and Xin Yu. Mobile-gs: Real-time gaussian splatting for mobile devices. *arXiv preprint arXiv:2603.11531*, 2026.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37:140138–140158, 2024.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5501–5510, 2022.

Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, pp. 54–71. Springer, 2024.

Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' rays: Uncertainty quantification for neural radiance fields. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20061–20070. IEEE, 2024.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5949–5958, 2025.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions On Graphics (TOG)*, 43(4):1–15, 2024.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37:80965–80986, 2024.

Hyunjin Kim, Haebeom Jung, and Jaesik Park. Metropolis-hastings sampling for 3d gaussian reconstruction. *Advances in Neural Information Processing Systems*, 38:142862–142887, 2026.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21719–21728, 2024.

Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Optimized minimal 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 38:135864–135888, 2026.

Weikai Lin, Yu Feng, and Yuhao Zhu. Metasapiens: Real-time neural rendering with efficiency-aware pruning and accelerated foveated rendering. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pp. 669–682, 2025.

Xiangrui Liu, Xinju Wu, Pingping Zhang, Shiqi Wang, Zhu Li, and Sam Kwong. Compgs: Efficient 3d scene representation via compressed gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 2936–2944, 2024.

Yifei Liu, Zhihang Zhong, Yifan Zhan, Sheng Xu, and Xiao Sun. Maskgaussian: Adaptive 3d gaussian representation from probabilistic masks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 681–690, 2025.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20654–20664, 2024.

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Panagiotis Papantonakis, Georgios Kopanas, Bernhard Kerbl, Alexandre Lanvin, and George Drettakis. Reducing the memory footprint of 3d gaussian splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1):1–17, 2024.

Hyunwoo Park, Gun Ryu, and Wonjun Kim. Dropgaussian: Structural regularization for sparse-view gaussian splatting. In *Proceedings of the computer vision and pattern recognition conference*, pp. 21600–21609, 2025.

Mia Gaia Polansky, George Kopanas, Stephan Garbin, Todd Zickler, and Dor Verbin. Eulerian gaussian splatting using hashed probability pyramids. *arXiv preprint arXiv:2605.29136*, 2026.

Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering. *ACM Transactions on Graphics (TOG)*, 43(4):1–17, 2024.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. Fastgs: Training 3d gaussian splatting in 100 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26094–26103, 2026.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in gaussian splatting. In *European Conference on Computer Vision*, pp. 347–362. Springer, 2024.

Seungjoo Shin, Jaesik Park, and Sunghyun Cho. Locality-aware gaussian compression for fast and high-quality rendering. In *International Conference on Learning Representations*, volume 2025, pp. 63–90, 2025.

Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, et al. Steepest descent density control for compact 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26663–26672, 2025.

Xinzhe Wang, Ran Yi, and Lizhuang Ma. Adr-gaussian: Accelerating gaussian splatting with adaptive radius. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–10, 2024.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 1053–1061, 2024.

A Appendix

A Additional Method Details

This appendix defines the geometry, state variables, objectives, and event ordering used in Sec. 4. Controls without a fixed value below are run-configuration inputs (Table 5). Rendered depth, attribution, gates, and credit are detached from live optimization.

A.1 Coordinates, Observations, and Spatial Attribution

Geometry and camera notation. A live Gaussian indexed by i has center $\boldsymbol{\mu}_i \in \mathbb{R}^3$, positive scales $\mathbf{s}_i \in \mathbb{R}^3_{>0}$, unit quaternion $\mathbf{q}_i \in \mathbb{R}^4$, physical opacity $\alpha_i \in (0,1)$, and spherical-harmonic (SH) color coefficients. Its covariance is $\Sigma_i = \mathbf{R}(\mathbf{q}_i)\,\mathrm{diag}(\mathbf{s}_i^2)\mathbf{R}(\mathbf{q}_i)^\top \succ 0$, where $\mathbf{R}(\mathbf{q}_i)$ is its rotation matrix. A candidate g has the same parameterization. The integer SH degree and basis convention are renderer inputs.

Camera $v \in \mathcal{V}$ has center $\mathbf{o}_v$, intrinsics $\mathbf{K}_v$, and world-to-camera rotation $\mathbf{R}_v$. Write $\mathbf{y}_v(\mathbf{x}) = \mathbf{R}_v(\mathbf{x} - \mathbf{o}_v)$, $z_v(\mathbf{x}) = [\mathbf{y}_v(\mathbf{x})]_3$, and $\pi_v(\mathbf{x}) = \mathrm{inhom}(\mathbf{K}_v\mathbf{y}_v(\mathbf{x}))$; inhom divides the first two coordinates by the third. Thus z_v denotes axial depth, whereas $z_t(c)$ is a demand logit. For pixel center p, its unit world-space ray direction is the normalized $\mathbf{R}_v^\top\mathbf{K}_v^{-1}(p^\top, 1)^\top$. Ray $r = (v, p)$ has origin $\mathbf{o}_r = \mathbf{o}_v$ and direction $\mathbf{d}_r$, so $\mathbf{x}_r(s) = \mathbf{o}_r + s\mathbf{d}_r$ with ray distance $s > 0$. Projection, image lookup, and rendering use the same pixel-coordinate convention.

Let $\mathbf{o}_0$ be the mean training-camera center and $L > 0$ their root-mean-square distance from $\mathbf{o}_0$. Apply $\mathbf{x}' = (\mathbf{x} - \mathbf{o}_0)/L$ and $\Sigma' = \Sigma/L^2$ to cameras and Gaussians, then omit primes. A coincident-camera configuration requires a supplied positive scale L. Cells are indexed by $c \in \mathbb{Z}^3$, with center $\mathbf{x}_c = h(c + \frac{1}{2}\mathbf{1})$; here $\mathbf{1} = (1,1,1)^\top$. No scene contraction is used. The interval $[a_{rc}, b_{rc}]$ is the intersection of the positive ray with cell c; an empty interval has zero mass.

Exposure, images, and time. Each sweep visits all $N_{\text{cam}} = |\mathcal{V}| > 0$ cameras once in a shuffled order. Each visit performs one live update followed by one evidence exposure. The exposure index is t, the sweep-time clock is $\tau_t = t/N_{\text{cam}}$, and $\Delta\tau_t = 1/N_{\text{cam}}$. Topology checks advance neither clock nor exposure count. The time superscript t^- denotes the state before processing the current exposure. Images have at least 16 pixels in each dimension. In each stratum of a 16×16 partition, choose the pixel nearest the stratum center, breaking ties lexicographically. Use uniform weights $\nu_r = 1/|\mathcal{R}_t|$. These pixels are reused for same-view comparisons. This fixed sampling does not guarantee observation of every residual region. Target images $\mathbf{I}_v^{\text{gt}}(p)$ use RGB values in $[0,1]$; $\widehat{\mathbf{I}}_v(p;\mathcal{G})$ is the corresponding rendering.

Compositing and Covered allocation. For each ray, $k \prec_r i$ denotes front-to-back order, with globally fixed Gaussian IDs. The effective pixel opacity $\widetilde{\alpha}_{ri} \in [0,1)$ includes the projected footprint and renderer clamp. Define

$$T_{ri} = \prod_{k \prec_r i} (1 - \widetilde{\alpha}_{rk}), \qquad w_{ri} = T_{ri}\widetilde{\alpha}_{ri}, \qquad T_r^{\text{end}} = \prod_i (1 - \widetilde{\alpha}_{ri}). \tag{11}$$

Products and depth statistics use all renderer-retained contributors, including its standard culling rules. The additional attribution cache retains at most 128 contributors with largest w_{ri}, breaking ties by depth and Gaussian ID; omitted cache entries receive no Covered mass. Full T_r^{end} is never replaced by one minus the cached weight sum.

Restriction of a Gaussian to the unit ray gives

$$\beta_{ri}^2 = (\mathbf{d}_r^\top \Sigma_i^{-1} \mathbf{d}_r)^{-1}, \qquad \gamma_{ri} = \beta_{ri}^2 \mathbf{d}_r^\top \Sigma_i^{-1}(\boldsymbol{\mu}_i - \mathbf{o}_r). \tag{12}$$

With $\mathcal{N}(s;\gamma,\beta^2)$ the scalar normal density, the actual finite-query kernel in Eq. 1 is $k_{ri}(s) = \mathcal{N}(s;\gamma_{ri},\beta_{ri}^2)\mathbf{1}\{s > 0,\ |s - \gamma_{ri}| \le 3\beta_{ri}\}$. Here $\mathbf{1}\{\cdot\}$ denotes an indicator, distinct from the vector $\mathbf{1}$ above. Only intersected cells are queried. Tails remain unassigned without renormalization. This allocates an optical weight; it does not replace the rasterizer with volumetric rendering.

Rendered depth and valid hypotheses. At each sweep's start, render a detached depth bank from the live snapshot. For $z_{vi} = z_v(\boldsymbol{\mu}_i) > 0$ and pixel weights w_{pi} in view v,

$$C_v(p) = \sum_i w_{pi}, \quad Z_v(p) = \frac{\sum_i w_{pi} z_{vi}}{C_v(p)}, \quad U_v(p) = \frac{\sum_i w_{pi}(z_{vi} - Z_v(p))^2}{C_v(p)}. \tag{13}$$

C_v is accumulated opacity, Z_v is mean center depth, and U_v is its between-contributor variance. A pixel is depth-supported when $C_v(p) \ge \tau_C > 0$, $Z_v(p) > 0$, and $\sqrt{U_v(p)}/Z_v(p) \le \tau_Z$. Undefined divisions are not evaluated. Depth lookups use the nearest pixel center, with lexicographic tie-breaking; out-of-image and behind-camera projections are invalid. U_v excludes within-Gaussian extent and is not a calibrated uncertainty. The bank is fixed within a sweep and refreshed for proposal checks.

For target ray $r = (v, p_r)$, backproject depth-supported pixels of other cameras using Z_u and the calibrated camera transform. Retain a point $\mathbf{x}$ only if $\|\pi_v(\mathbf{x}) - p_r\|_2 \le \tau_{\text{px}}$ and its projected ray distance $s = \mathbf{d}_r^\top(\mathbf{x} - \mathbf{o}_r) > 0$. For a camera u with a valid projection, define the relative depth discrepancy

$$\delta_u(\mathbf{x}) = \left| Z_u(\pi_u(\mathbf{x}))/z_u(\mathbf{x}) - 1 \right|. \tag{14}$$

Let $\mathcal{U}(\mathbf{x};r)$ contain non-target cameras $u \ne v$ with valid, depth-supported projections and $\delta_u(\mathbf{x}) \le \tau_{\text{inv}}$. Require at least two distinct non-target cameras; the generating camera may count as one. Project the point onto $\mathbf{x}_r(s)$ and apply the same support checks again. Among surviving projected hypotheses, select the one with smallest mean δ_u^2 over its supporting cameras, breaking ties by s, source camera ID, and source pixel index. Its distance is $s_r^\star$. The threshold τ_{inv} is a relative depth-ratio tolerance; its legacy subscript does not denote an unnormalized inverse-depth difference.

Null allocation and its scope. For a surviving hypothesis set $V_r^{\text{null}} = 1$ and use

$$q_r(s) = \frac{\mathcal{N}(s; s_r^\star, h^2)\mathbf{1}\{s > 0,\ |s - s_r^\star| \le 3h\}}{\Phi(s_r^\star/h)}, \tag{15}$$

where Φ is the standard normal cumulative distribution function. The denominator normalizes positive-ray truncation; the finite $3h$ query tails remain unassigned, so this allocation kernel integrates to at most one. If no hypothesis survives, set $V_r^{\text{null}} = 0$ and $q_r \equiv 0$. Consequently $\sum_c (m_{rc}^{\text{cov}} + m_{rc}^{\text{null}}) \le 1$. The kernel is a localization convention, not a posterior. Depth is predicted by the current Gaussians, with no external depth supervision. Surfaces without cross-view depth support, or hidden behind opaque erroneous geometry in all relevant views, can remain undiscovered.

A.2 Temporal Evidence and Demand Updates

Observation records and gates. Maintain branch masses, residuals, weighted target-RGB sums, camera IDs, and sweep timestamps for each observed cell, even before creating a trainable demand state. RGB sums use the same ray/attribution weights as the masses. Records older than $H = 2$ sweeps are invalid. A branch observation has sufficient support when its mass is at least $m_{\min} > 0$. For the current camera v, compare its previous and current supported Covered observations before overwriting the previous record. Write E_{vc}^{prev} and E_{vc}^{now} for their respective $R^{\text{cov}}/M^{\text{cov}}$ ratios. The non-improvement score is

$$P_t(c) = \exp\left(-\frac{[E_{vc}^{\text{prev}} - E_{vc}^{\text{now}}]_+}{\tau_P(E_{vc}^{\text{prev}} + \epsilon)}\right). \tag{16}$$

$\tau_P > 0$ controls sensitivity to relative improvement. This score is also high for newly increased error; persistence is supplied by repeated observations and the demand state, rather than this score alone. $V_t^{\text{cov}}(c) = 1$ requires the valid same-view pair and sufficient Covered observations from at least two distinct cameras within H sweeps, including the current observation. Otherwise set $V_t^{\text{cov}}(c) = 0$ and $P_t(c) = 0$. The actionable set is

$$\mathcal{C}_t = \{c : V_t^{\text{cov}}(c) = 1 \text{ or } M_t^{\text{null}}(c) \ge m_{\min}\}. \tag{17}$$

Each nonzero Null ray already has the required cross-view depth support. An empty $\mathcal{C}_t$ causes no demand update or credit injection.

Optimizer-response score. For a Gaussian with nonzero post-update compositing weight in the current view, $-$ and $+$ denote pre- and post-update parameters. Define

$$\begin{aligned} b_i^\mu &= \frac{\|\Delta\boldsymbol{\mu}_i\|_2}{\max(h, \|\mathbf{s}_i^-\|_2)}, & b_i^s &= \frac{\|\Delta \log \mathbf{s}_i\|_2}{\sqrt{3}}, \\ b_i^\alpha &= |\Delta\alpha_i|, & b_i^q &= 2\arccos\big(\min\{1, |\langle \mathbf{q}_i^+, \mathbf{q}_i^-\rangle|\}\big), \\ b_i^c &= \|\Delta\mathbf{c}_i\|_2/\sqrt{3}. \end{aligned} \tag{18}$$

Δ means post-update minus pre-update, and logs of scales are componentwise. $\mathbf{c}_i$ is the Gaussian's RGB color evaluated at the same viewing direction for both states, using the pre-update center. Quaternion distance is sign invariant. The index k ranges over active optimizer parameter blocks $\mathcal{B}_{i,t}$, with actual scalar learning rates $\ell_{k,t} > 0$. Set $u_{i,t} = |\mathcal{B}_{i,t}|^{-1} \sum_{k \in \mathcal{B}_{i,t}} b_i^k/\ell_{k,t}$. If SH groups have different learning rates, use one color block per group, measuring its isolated color change while other groups and viewing direction are fixed. Disabled blocks are excluded; with none active, the current response is unavailable.

Track the exponential moving average $\bar{u}_{i,t} = (1 - \beta_S)\bar{u}_{i,t^-} + \beta_S u_{i,t}$ with $\beta_S \in (0, 1]$, updated only for available visible responses and initialized at the first such update. With a current response and at least two such updates, set $s_{i,t} = \exp(-\bar{u}_{i,t}/\tau_S)$, where $\tau_S > 0$ is a response scale; otherwise set $s_{i,t} = 0$. This scalar score is distinct from physical scales $\mathbf{s}_i$. The cell score is

$$S_t(c) = \frac{\sum_{r \in \mathcal{R}_{t,i}} \nu_r m_{ric}^{\text{cov}} s_{i,t}}{M_t^{\text{cov}}(c) + \epsilon}. \tag{19}$$

Small realized responses increase this settlement score. Learning-rate normalization does not remove Adam history, parameterization, or saturation, and does not certify convergence.

Loss components and stable evaluation. For the frozen evidence tuple $\mathcal{E}_t = (\mathcal{C}_t, \omega_t, D_t, d_{t^-})$, the two terms used in Sec. 4.5 are exactly

$$\begin{aligned}\mathcal{L}_{\mathrm{BCE}}(z; \mathcal{E}_t) &= \sum_{c \in \mathcal{C}_t} \omega_t(c) \ell_{\mathrm{BCE}}(\sigma(z(c)), D_t(c)), \\ \mathcal{L}_{\mathrm{temp}}(z; \mathcal{E}_t) &= \sum_{c \in \mathcal{C}_t} \omega_t(c) (\sigma(z(c)) - d_{t^-}(c))^2.\end{aligned} \tag{20}$$

Thus λ_{temp} is applied once, outside the second sum. Stable evaluation uses the logit form

$$\ell_{\mathrm{BCE}}(\sigma(z), D) = \mathrm{softplus}(z) - Dz, \qquad \mathrm{softplus}(z) = \max(z, 0) + \log(1 + e^{-|z|}). \tag{21}$$

Only z is differentiated, with derivative $\sigma(z) - D$; the soft target need not be binary. The loss is not bounded merely because its state is bounded, and it does not calibrate a missing-geometry probability.

State initialization and update clock. At first actionability, set $d_{t^-}(c) = \tau_d/2$, $z_0(c) = \mathrm{logit}(d_{t^-}(c))$, and zero credit, where $\mathrm{logit}(x) = \log(x/(1-x))$. Later exposures use the last stored demand as d_{t^-}. Use four masked Adam steps per actionable exposure with learning rate $\eta_z = 0.1$, $\lambda_{\mathrm{temp}} = 0.1$, $(\beta_1, \beta_2) = (0.9, 0.999)$, and $\epsilon_A = 10^{-8}$. To specify finite-update memory unambiguously, each cell has its own step counter k_c and moments m_c^z, n_c^z, all initialized to zero. For each active inner step, increment k_c, set $g_c^z = \partial \mathcal{L}_{\mathrm{demand}}/\partial z(c)$, and apply

$$\begin{aligned}m_c^z &\leftarrow \beta_1 m_c^z + (1-\beta_1) g_c^z, \qquad n_c^z \leftarrow \beta_2 n_c^z + (1-\beta_2)(g_c^z)^2, \\ z(c) &\leftarrow z(c) - \eta_z \frac{m_c^z/(1-\beta_1^{k_c})}{\sqrt{n_c^z/(1-\beta_2^{k_c})} + \epsilon_A}.\end{aligned} \tag{22}$$

Inactive cells receive neither zero targets nor optimizer steps; their moments and counters remain unchanged. All four steps share the same detached targets, weights, and pre-exposure state. Store d_t after the fourth step. This cell-local clock is an explicit masked-Adam convention; a shared tensor clock in an off-the-shelf SparseAdam implementation is not equivalent. Expiry resets the logit, moments, counter, and credit before any later recreation.

A.3 Compatible Regions and Provisional Gaussians

Fresh eligibility. Let τ_c^{obs} be the last actionable exposure time and τ_c^{try} the last fitting-attempt time, initialized to $-\infty$. A cell must have $d_t(c) > \tau_d$, positive current credit, valid support within H sweeps, and $\tau_c^{\mathrm{obs}} > \tau_c^{\mathrm{try}}$. At a topology round, settle credit to its current time, rerender the last actionable ray set, and refresh relevant depth-support maps using the current live set. Recompute masses, residuals, and support validity while holding the cached scalar $P_t(c)$ and $S_t(c)$ fixed; call the resulting target $D^{\mathrm{check}}(c)$. Require $\min\{d_t(c), D^{\mathrm{check}}(c)\} > \tau_d$. This read-only check creates neither history nor credit.

Deterministic compatible partition. Process eligible cells by decreasing credit, then lexicographic cell index. Each unassigned cell seeds a region, grown breadth first through face-adjacent unassigned cells within distance Rh of the seed, with $R = 8$; visit neighbors lexicographically. Let $\boldsymbol{\chi}_c$ be the mean target RGB over the cell's support-valid records in the history window, using weights $\nu_r(m_{rc}^{\mathrm{cov}} + m_{rc}^{\mathrm{null}})$ and dividing by their sum. Require mean channel difference $\|\boldsymbol{\chi}_c - \boldsymbol{\chi}_{\mathrm{seed}}\|_1/3 \le \tau_{\mathrm{RGB}}$ with $\tau_{\mathrm{RGB}} = 0.25$.

For every admitted cell, require two common depth-supported cameras whose current maps satisfy Eq. 14 with $\delta_v(\mathbf{x}_c) \le \tau_{\mathrm{inv}}$. The same pair must also satisfy

$$\frac{\max_{c \in \mathcal{A}_j} z_v(\mathbf{x}_c) - \min_{c \in \mathcal{A}_j} z_v(\mathbf{x}_c)}{\min_{c \in \mathcal{A}_j} z_v(\mathbf{x}_c)} \le \tau_{\mathrm{layer}}. \tag{23}$$

At every growth step test the entire tentative region, accepting a new cell only if some camera pair satisfies all conditions; choose the lexicographically first valid pair. A seed without such support forms no proposal. $\tau_{\mathrm{layer}} > 0$ controls the permitted relative depth span. These tests apply to both Covered and Null cells. Thus even Covered demand can be withheld when the current representation lacks consistent center depths. Ray-restricted modes and rendered center depths are different geometric quantities; this compatibility test is an additional heuristic, not a theorem that all attributed demand is realizable.

Initialization and constrained fitting. Regions contain positive credit, so moment weights η_c are well defined. Clip eigenvalues of $\mathbf{\Sigma}_j^D$ to $[h^2/12, (Rh)^2]$ and call the result Σ_j^0. Its eigenvalue square roots initialize scales; choose eigenvector signs so the rotation lies in $\mathrm{SO}(3)$, the group of proper three-dimensional rotations, and convert it to a unit quaternion. Choose the DC (constant) SH coefficient to render the weighted color $\sum_c \eta_c \boldsymbol{\chi}_c$ under the renderer's basis and color-offset convention, set higher-order coefficients to zero, and initialize opacity at $\alpha_{\mathrm{init}} \in (0,1)$. Candidate parameters satisfy

$$\begin{gathered}(\boldsymbol{\mu}-\boldsymbol{\mu}_j^D)^\top(\Sigma_j^0)^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_j^D)\le 1, \qquad \|\boldsymbol{\mu}-\mathbf{x}_{\mathrm{seed}}\|_2 \le Rh, \\ h/\sqrt{12} \le s_k \le Rh \quad (k=1,2,3).\end{gathered} \tag{24}$$

Here $\boldsymbol{\mu}$ and s_k are the candidate center and principal scales. The initial mean is feasible by convexity of the seed ball. After each candidate Adam step, project position onto the intersection of the two convex position sets, clamp log scales, and normalize the quaternion (retain its preceding value if its norm becomes zero). Opacity remains sigmoid parameterized. $\mathcal{V}_j$ is the union of the region's valid observation cameras and selected depth-support cameras. Fit for the configured integer $J_{\mathrm{prov}} \ge 16$ steps on the full $\mathcal{V}_j$ objective, with declared parameter-group learning rates and the live set frozen.

Photometric loss and SSIM convention. For the pixel set Ω_v of view v, define $\mathcal{L}_1^v(\mathcal{G}) = (3|\Omega_v|)^{-1}\sum_{p\in\Omega_v}\|\widehat{\mathbf{I}}_v(p;\mathcal{G}) - \mathbf{I}_v^{\mathrm{gt}}(p)\|_1$. For any nonempty view set $\mathcal{S}$,

$$\mathcal{L}_{\mathcal{S}}^{\mathrm{photo}}(\mathcal{G}) = \frac{1}{|\mathcal{S}|}\sum_{v\in\mathcal{S}}\ell_v(\mathcal{G}), \qquad \ell_v = (1-\lambda_{\mathrm{DSSIM}})\mathcal{L}_1^v + \lambda_{\mathrm{DSSIM}}(1-\mathrm{SSIM}^v). \tag{25}$$

Use $\lambda_{\mathrm{DSSIM}} = 0.1$ for live, candidate, and gain evaluation. SSIM denotes structural similarity; dissimilarity here is $1-\mathrm{SSIM}$, without an additional factor of $1/2$. Define SSIM^v as the average across RGB channels and a nonempty set of image windows, using

$$\frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}. \tag{26}$$

Here x, y are the rendered and target channel values in a window; μ_x, μ_y are its weighted means, σ_x^2, σ_y^2 its weighted variances, and σ_{xy} its weighted covariance, without sample-variance correction. The window shape, nonnegative weights summing to one, evaluated centers, boundary handling, and positive constants C_1, C_2 are declared SSIM configuration inputs. No unreported numerical setting is inferred. Preprocessing, backgrounds, and all SSIM settings are identical for both sides of every comparison.

Contribution and acceptance. Besides Eq. 7, require the trust constraints, $\alpha_g \ge 1/255$, and at least two views in $\mathcal{V}_j$ each containing a pixel with actual candidate compositing weight $w_{pg} \ge \tau_w > 0$. This weight includes occlusion by the live set and is not the bare opacity. Let $\mathcal{W}_j \subseteq \mathcal{V}$ be the training views whose rasterized image changes after inserting g. Then the exact gain is

$$\Delta_j = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{W}_j}[\ell_v(\mathcal{G}) - \ell_v(\mathcal{G}\cup\{g\})]. \tag{27}$$

Evaluate every changed view, or conservatively all training views, with full image losses including SSIM. Hold the live parameters and renderer settings fixed. Acceptance certifies improvement of this training objective at that instant, not geometric accuracy or future monotonic improvement.

A.4 Intensity Accounting and Freshness

Ledger state and event order. Every base cell owns one nonnegative balance $a(c)$, last ledger-update time τ_c^-, and the observation/attempt timestamps defined above. On state creation, set $a(c) = 0$, $\tau_c^- = \tau_c^{\mathrm{obs}} = \tau_t$, and $\tau_c^{\mathrm{try}} = -\infty$. Before reading or updating a balance at current sweep time τ, multiply it by $e^{-(\tau-\tau_c^-)/\tau_a}$ and set $\tau_c^- = \tau$. For a real actionable exposure only, inject $\kappa\Delta\tau_t M_t(c)\rho_t(c)$ once, using demand after the four inner updates. Equation 8 combines these two operations; it must not apply decay twice. Require $0 < \tau_d < 1-\epsilon_d < 1$, $\kappa > 0$, and $\tau_a > 0$. Changing ray-batch partition does not create extra exposure duration. Each actionable real exposure

also sets $\tau_c^{\mathrm{obs}} = \tau_t$. In Eq. 9, all $a_t(c)$ have been settled to the same current time and already include preceding accepted debits.

For current regional balance $A = \sum_{c \in \mathcal{A}_j} a(c) \geq 1$, an acceptance atomically debits

$$b_j(c) = a(c)/A, \qquad a(c) \leftarrow a(c) - b_j(c), \qquad \sum_{c \in \mathcal{A}_j} b_j(c) = 1. \tag{28}$$

The debit shares $b_j(c)$ use the same pre-debit balances. They are unrelated to the branch index b or update magnitudes b_i^k. All balances remain nonnegative. Regions cannot spend another region's credit; any hierarchy stores cached sums, not additional balances.

Expiry and retry. If $\tau - \tau_c^{\mathrm{obs}} > H$, expire the remaining credit and invalidate that cell's demand and comparison history before processing new evidence. Remove old observation-only records by the same age rule; new records may accumulate before a demand state becomes actionable again. After any actual fitting attempt, accepted or rejected, set $\tau_c^{\mathrm{try}} = \tau$ for its participating cells. Rejection consumes no credit. Neither outcome is retried in the same round, and later attempts require a later actionable real exposure and a new round. Before a queued region is fitted, refresh it against the current live set; if it splits into compatible components, try only the component with greatest current credit, breaking ties by seed index, and defer the rest.

Accounting and population bound. Let A_0 be initial total credit, I_t cumulative injected credit, B_t the number of accepted births, E_t^{credit} credit removed by decay and expiry, and A_t^{all} all remaining credit, settled to a common time. Then, up to floating-point error,

$$A_0 + I_t = B_t + E_t^{\mathrm{credit}} + A_t^{\mathrm{all}}. \tag{29}$$

Injection adds equally to both sides; a birth transfers exactly one unit to B_t; decay and expiry transfer removed credit to E_t^{credit}. Rejection and regrouping do not change totals. This proves the identity by induction. Since $\sum_c M_t(c) \leq 1$ and $\rho_t(c) \leq \rho_{\max} = \log((1-\tau_d)/\epsilon_d)$, zero initial credit gives $B_T \leq \lfloor \kappa T_{\mathrm{sweep}} \rho_{\max} \rfloor$ after T_{sweep} sweeps. Here T denotes the final exposure index. This bounds additions, not the minimum representation required by a scene.

Persistent demand need not reach a birth threshold. For a continuously valid fixed region with constant exposure interval $\Delta\tau$ and injection factor $J = \sum_{c \in \mathcal{A}_j} M(c)\rho(c)$, no debit or expiry, its balance approaches $\kappa \Delta\tau J/(1 - e^{-\Delta\tau/\tau_a})$. If this limit is at most one and initial credit is zero, no finite-time birth is authorized. Rates and lifetimes must therefore be specified with the intended support scale. Accounting does not establish convergence, grid invariance, or minimality.

A.5 Online Execution

Each topology round begins with a finite queue of disjoint regions, ordered by current balance and then seed index. It allows at most one attempt per queued region; checks and commits are sequential. Initial live Gaussians, live/candidate learning-rate schedules, SH degree, and renderer settings are configuration inputs. Live and candidate optimization use Adam; the candidate starts with zero moments and step count. On acceptance its live optimizer state is reset, including its own bias-correction counter, and its response history starts anew. These counters are attached to each Gaussian's parameter blocks and advance once per live Adam step after that Gaussian's insertion. Native clone/split and periodic opacity resets are disabled; this controller does not prune live Gaussians. Depth supports attribution, not a live depth loss.

A.6 Control Parameters and Figure Notation

Table 5 separates prescribed constants from required configuration inputs. All learning rates refer to their optimized coordinates: position, log scale, opacity logit, quaternion, and SH coefficients; the response statistic instead measures the physical changes explicitly defined above. Live/candidate Adam coefficients are $(0.9, 0.999)$ with $\epsilon_A = 10^{-8}$. Their numerical learning-rate schedules and actual J_{prov} must accompany a reproducible run.

In Fig. 1, the displayed field $q_t(c)$ denotes $d_t(c)$, not the Null ray kernel $q_r(s)$ or quaternion $\mathbf{q}_i$. The displayed live set $\mathcal{G}_t$ is $\mathcal{G}$ at that stage; $\mathcal{L}_{\mathrm{live}} = \mathcal{L}_{\mathrm{rec}}$ and $\mathcal{L}_{\mathrm{field}} = \mathcal{L}_{\mathrm{demand}}$. Residual persistence in

Algorithm 1 Online optimization with explicit state and credit updates

Require: Initial $\mathcal{G}$, calibrated $\mathcal{V}$, images and run controls
Initialize observation records, demand states, and credits as empty; $t \leftarrow 0$
for $n = 1, \ldots, T_{\text{sweep}}$ **do**
  Render the detached depth bank; shuffle all cameras
  for each camera v **do**
    $t \leftarrow t + 1$; set $\tau_t = t/N_{\text{cam}}$ and expire stale states
    Save live parameters; take one live RGB Adam step
    Rerender sampled rays and measure the realized parameter changes
    Form Covered and validated Null masses and residuals
    Compare previous records before storing current records; form $\mathcal{C}_t$
    Initialize newly actionable states and compute D_t
    Take four demand steps with fixed evidence and cell-local clocks
    Settle balances and inject once for this real exposure
  end for
  Refresh eligibility, settle credit, and partition a finite region queue
  for each queued region **do**
    Refresh against current $\mathcal{G}$; select its largest valid component
    if fresh eligibility holds and current balance $A \geq 1$ **then**
      Initialize by demand moments; fit for J_{prov} constrained steps
      Evaluate gain, actual contribution, and spatial constraints
      if all acceptance tests pass **then**
        Commit the candidate, reset its live optimizer state, and debit one unit
      end if
      Stamp attempted cells and defer further trials to a later round/exposure
    end if
  end for
end for
return $\mathcal{G}$

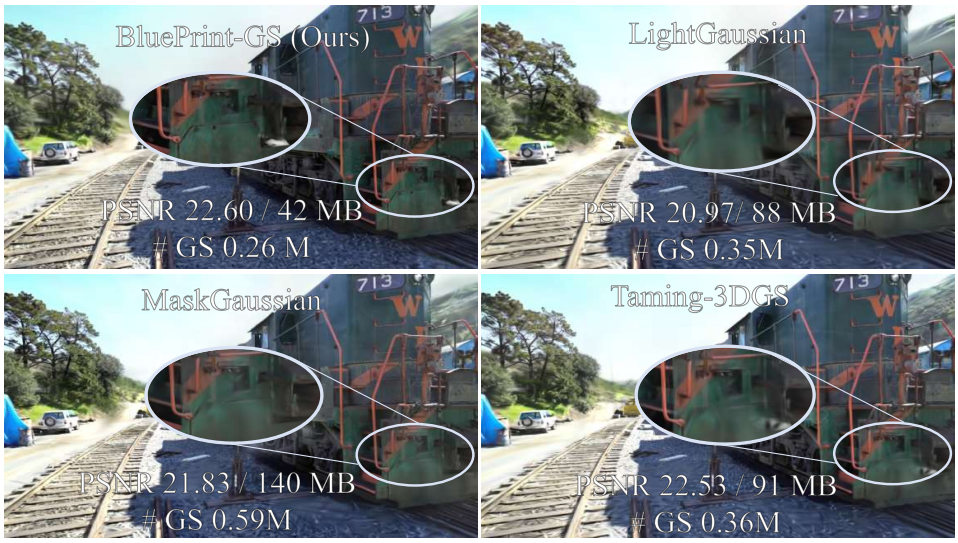


Figure 4: Rendering result of Train Dataset

panel (b) refers collectively to repeated evidence and demand memory; event scale in panels (a,c) is credit allowance, not an instruction to insert all K_t candidates immediately.

Table 5: Controls and admissible inputs. A range defines an algorithm input, not a measured or tuned default.

Control	Meaning and specification
h, R, H	Cell width 0.001, seed-radius multiplier 8, history lifetime 2 sweeps.
Sampling/cache	Fixed 16×16 rays per camera; at most 128 attribution contributors.
ϵ, ϵ_d	Positive mass safeguard and demand clipping margin; require $0 < \tau_d < 1 - \epsilon_d < 1$.
τ_C, τ_Z	Minimum accumulated opacity in $(0, 1]$ and maximum relative center-depth spread > 0.
$\tau_{\mathrm{px}}, \tau_{\mathrm{inv}}$	Pixel reprojection radius and relative depth-ratio tolerance, both > 0.
$m_{\min}$	Minimum branch support, in $(0, 1]$.
τ_P, β_S, τ_S	Non-improvement scale > 0, response EMA coefficient in $(0, 1]$, response scale > 0.
$\eta_z, \lambda_{\mathrm{temp}}$	Demand learning rate 0.1, temporal coefficient 0.1; four inner steps.
τ_d, κ, τ_a	Demand dead zone, positive credit rate per sweep, and positive lifetime in sweeps.
$\tau_{\mathrm{RGB}}, \tau_{\mathrm{layer}}$	RGB tolerance 0.25 and configured relative depth-span tolerance > 0.
$\alpha_{\mathrm{init}}, J_{\mathrm{prov}}$	Candidate opacity in $(0, 1)$ and configured integer fitting steps ≥ 16.
$\lambda_{\mathrm{op}}, \tau_{\mathrm{gain}}, \tau_w$	Nonnegative opacity penalty, positive minimum global gain, and contribution threshold in $(0, 1]$.
λ_{DSSIM}	Photometric structural-loss weight 0.1.
T_{sweep}	Configured positive integer training duration in camera sweeps.
Optimizer/rendering	Declared live/candidate learning rates, initialization, SH degree/basis, image preprocessing, background, and renderer culling/clamping conventions.
SSIM settings	Declared window, normalized weights, evaluated centers, boundary rule, and positive stabilizers C_1, C_2.