

BOGausS: Better Optimized Gaussian Splatting

Stéphane Pateux*, Matthieu Gendrin*[†], Luce Morin[†], Théo Ladune*, Xiaoran Jiang[†]

*Orange Innovation, Cesson Sévigné, France

[†]Univ Rennes, INSA Rennes, CNRS, IETR-UMR 6164, F-35000 Rennes, France
name.surname@orange.com, name.surname@insa-rennes.fr

Abstract—3D Gaussian Splatting (3DGS) proposes an efficient solution for novel view synthesis. Its framework provides fast and high-fidelity rendering. Although less complex than other solutions such as Neural Radiance Fields (NeRF), there are still some challenges building smaller models without sacrificing quality. In this study, we perform a careful analysis of 3DGS training process and propose a new optimization methodology. Our Better Optimized Gaussian Splatting (BOGausS) solution is able to generate models up to ten times lighter than the original 3DGS with no quality degradation, thus significantly boosting the performance of Gaussian Splatting compared to the state of the art.

Index Terms—Gaussian splatting, SGD optimizer.

I. INTRODUCTION

3D scene reconstruction has had significant improvements since the rise of Neural Radiance Fields (NeRF) [1] and 3D Gaussian splatting (3DGS) [2], opening the opportunity to many free view-point applications. 3DGS has gained high popularity thanks to its high quality image rendering, reasonable training time and real time rendering. Among the various challenges associated to 3DGS is the optimization step of parameters estimation of the 3D Gaussians composing a scene. Several studies [3]–[6] propose to either improve 3DGS rendering quality or reduce the number of Gaussians.

In this study, we propose to revisit the optimization process of 3DGS in order to jointly improve on model quality and compaction. First we propose a model to derive attainable precision reconstruction of a 3D scene. This precision model is then exploited to provide precision-aware updates during Stochastic Gradient Descent (SGD). We also propose a new unbiased optimizer based on Sparse Adam to improve SGD when dealing with sparsely-viewed and split Gaussians. Targeting compactness of the model, we propose an improved densification mechanism with an analogy to Rate Distortion Optimization [7]. Our main contributions are as follows:

- *Confidence estimation of 3D reconstruction.* Through a parallel with Variational Bayesian Inference, we propose to define confidence metrics on geometry reconstruction. From these findings a minimal size for each Gaussian is defined depending on its localization.
- *Unbiased Sparse Adam Optimizer.* We propose a revised Optimizer that ensures unbiased smooth Stochastic Gradient Descent. This optimizer also exploits reconstruction’s confidence intervals to automatically adapt to scene content and camera setup.
- *Density preserving Gaussian splitting.* We propose a new splitting mechanism that allows smooth loss transition.

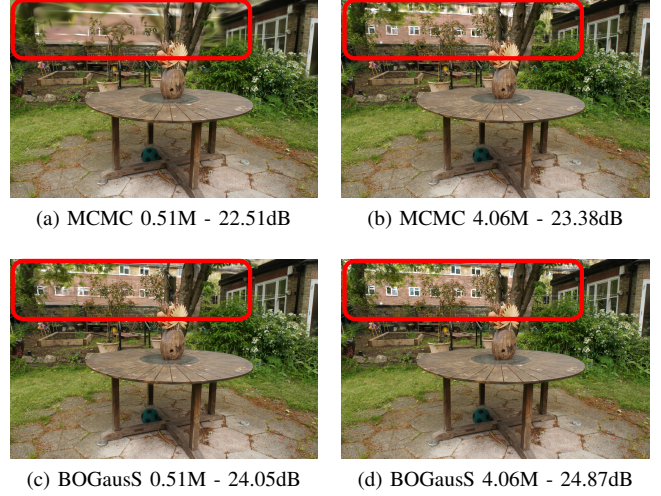


Fig. 1. Qualitative visual comparison of our proposed method with MCMC approach [5] on Garden scene with varying numbers of Gaussians. Our proposed method outperforms both visually and objectively MCMC thanks to a better reconstruction on background.

- *Distortion-based densification of Gaussians.* We propose densification and pruning strategy inspired from rate-distortion optimization with new prioritization metrics.

Our proposed solution achieves 0.5-1 dB PSNR increase over existing methods. Extensive experimental results and ablation study demonstrate the benefits of our proposed BOGausS method on several benchmarks and datasets.

II. RELATED WORK

A. 3D Gaussian Splatting

3D Gaussian Splatting [2] models the scene geometry as a set $\mathbb{G}$ of 3D Gaussians. Each Gaussian is characterized by its opacity $o_i \in [0, 1]$, center’s localization $P_i \in \mathbb{R}^3$ and a covariance matrix in world space Σ_i defining its spatial density $\mathcal{G}_i(X) = e^{-\frac{1}{2}(X-P_i)^T \Sigma_i^{-1}(X-P_i)}$. The covariance matrix Σ_i is defined via sizes $\sigma_{i,a}, a \in [1..3]$ and a rotation. To render an image $R^{(c)}$ associated to c -th camera, 3DGS sorts the Gaussians in an approximate depth order based on the distance from their center P_i to the image plane, and further applies alpha blending as initially proposed by Elliptical Weighted Averaging (EWA) [8] for each pixel p :

$$\begin{aligned}
R^{(c)}(p) &= \sum_{i \in \mathbb{G}} w_i^{(c)} c_i \\
\text{with } w_i &= T_i^{(c)} o_i \mathcal{G}_i^{2D,c}(p) \\
\text{and } T_i^{(c)} &= \Pi_{j=1}^{i-1} (1 - o_j \mathcal{G}_j^{2D,c}(p)) \quad (1)
\end{aligned}$$

where c_i represents the view-dependent color modeled by spherical harmonics coefficients, and $\mathcal{G}_j^{2D,c}$ is the projected 2D Gaussian distribution of $\mathcal{G}_j$ through a local affine transformation.

From (1) a loss function is defined based on a combination of distortion metrics such as L_1 or L_2 error, SSIM or other regularization metrics (e.g. depth regularization as in revised version of [2], opacity and/or sizes decay [5]). This loss function is then minimized using Stochastic Gradient Descent techniques.

B. Densification Optimization

Finding the right number of Gaussians is important in order to get the best compromise between model complexity and quality rendering. Gaussians are generally initialized from sparse points (either issued from Structure-from-Motion pre-calibration or from random samples) and their density is iteratively refined by using pruning and splitting operations interleaved with loss minimization. Several approaches have been proposed to optimize the selection of Gaussians to be pruned or split.

Pruning. In original 3DGS [2], several criteria are used to prune Gaussians. Gaussians with low opacity or with too large size are pruned. [3], [6] propose a prioritization criterion based on the effective contribution of a Gaussian. Effective contribution to camera c rendering is measured as $\mathcal{P}_i^{(c)} = \sum_p w_i^{(c)}(p)$.

Splitting. Original 3DGS [2] uses a prioritization mechanism based on visual positional gradient. Further studies have improved 3DGS performance by proposing various prioritization criteria based on the marginal loss contribution [9], on opacity [5], or on a mixture of metrics [4].

Although providing significant improvements, these criteria remain empirically defined and tuned (e.g. in [4] a dozen of hyperparameters have to be tuned).

III. PRECISION MATRIX DERIVATION

One of the well-known artifacts of 3DGS is the 'needle' effect. Needle effect typically occurs when zooming in; thin Gaussians show up. [10], [11] propose solutions to limit these artifacts by considering either correction of low pass filtering as proposed in original EWA [8], or minimal size constraints for Gaussians. This highlights the need to control the minimal size of Gaussians. [11] suggests that this minimal size should be defined so that its projected size is not smaller than pixel size. We propose here to rather use the precision accuracy on size estimation to define this minimal size.

Inspired from [12] using Variational Bayesian estimation in the context of 3D reconstruction from multiple cameras,

we model the reconstruction confidence on a Gaussian localization P_i . 3D Gaussians' localizations are obtained from their projected 2D localization. So if 2D projected localizations are subject to Gaussian noise with variance $\sigma^{(uv)^2}$, then the 3D localization of the i -th Gaussian will also be subject to Gaussian noise with covariance $\Sigma_i^{(XYZ)}$ given by:

$$\Sigma_i^{(XYZ)^{-1}} = \sum_c \left(z_i^{(c)^2} \sigma^{(uv)^2} W_c^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} W_c \right)^{-1} \quad (2)$$

where W_c is the world transformation matrix associated to camera c and $z_i^{(c)}$ is the distance of the Gaussian along the z axis of the c -th camera. The same reasoning can be applied to derive the precision matrix for size, which results in the same formula, $\Sigma_i^{(\sigma)} = \Sigma_i^{(XYZ)}$.

In this study, for simplicity, a scalar isotropic precision approximation is used to define confidence intervals for localization and sizes of the Gaussian as $\Delta_i = \alpha \text{tr}(\Sigma_i^{(XYZ)^{-1}})^{-0.5}$ where α is a scaling factor set to 2. This scaling factor helps to accommodate with modeling approximations and also with the fact that in (2) a Gaussian may not be seen by all cameras due to occlusions.

IV. IMPROVED GRADIENT DESCENT

A. Proposed optimizer

First we propose to modify the traditional Adam optimizer [13] used as follows:

$$\begin{cases} \alpha_{s,t} &= \max(1/t, 1 - \beta_s^*) \\ \hat{g}_0 &= 0 \\ \hat{g}_t &= \widehat{g_{t-1}} + \alpha_{1,t} (\nabla_{\theta} \mathcal{L}_t - \widehat{g_{t-1}}) \\ \hat{g}_t^2 &= \widehat{g_{t-1}^2} + \alpha_{2,t} ((\nabla_{\theta} \mathcal{L}_t)^2 - \widehat{g_{t-1}^2}) \\ \theta_{t+1} &= \theta_t + \eta_{\theta} \Delta_{\theta} \frac{\hat{g}_t}{\widehat{g}_t^2 + \epsilon} \end{cases} \quad (3)$$

where η_{θ} is the learning rate associated with θ parameter to optimize. (β_s^*) are set as in traditional Adam optimizer. Modifications are related to the introduction of the Δ_{θ} scaling factor in the update and the computation of filtered gradient states. These modifications allow us to introduce positional-aware parameters' updates, a better handling of Gaussians not seen in some cameras and Gaussian states after splitting operations.

B. Precision-aware parameters updates

The confidence intervals derived in section III can also link the 3D variations of localization and sizes to their impact on associated 2D projections and consequently to rendering variations. Thus, we propose to use the confidence interval Δ_i as Δ_{θ} parameter in (3) for localization and sizes parameters. For all other parameters we use $\Delta_{\theta} = 1$.

Consequently, sizes parameters (σ_a) are linearly parameterized, unlike in 3DGS implementations where logarithmic parameterization is used (i.e. $\theta_{\sigma} = \log \sigma$). Logarithmic parameterization allows helping Gaussians sizes to evolve fast which is beneficial for far away Gaussians. Logarithmic

parameterization of localization has also been proposed for 3D coordinates contraction (e.g. [14]) for the same purpose. Unfortunately these parameterizations are also more sensitive: any strong update may throw away Gaussians or explode their sizes which often occurs in outdoor scenes. Our proposed precision-aware update is less prone to such artifacts and adapts to localization of the Gaussians and the camera setup. It is thus more general than other positional-aware mechanism such as proposed in [15] which implicitly assumes a specific camera setup.

C. Sparse and unbiased states updates

As in [4] we propose to use Sparse Adam to perform parameters update only on Gaussians that contribute to the rendered image being optimized. This allows to speed up code with a light performance decrease. However, unlike biased estimator used in [4], the BoGausS proposed modified Adam’s optimizer defined by (3) allows ensuring unbiased states ($\hat{g}_t, \hat{g}_t^2$) estimation which prevents performance decrease and rather helps improve convergence (see ablation study in section VI-C). Indeed instant t in (3) is Gaussian dependent and only incremented when a Gaussian is updated. t then naturally refers to Gaussian’s lifespan. This additional state in optimizer is needed to ensure unbiased states.

D. States inheritance after Gaussian splitting

When performing a splitting operation, the states of the Adam’s optimizer are generally reset to 0. In our approach, gradient states of Adam optimizer are partially propagated to Gaussian children via the following inheritance:

$$\begin{cases} \overline{t_{child}} &= \alpha_t \times \overline{t_{mother}} \\ \overline{(g_{child})_t} &= \alpha_g \times \overline{(g_{mother})_t} \\ \overline{(g_{child}^2)_t} &= (\alpha_g)^2 \times \overline{(g_{mother}^2)_t} \end{cases} \quad (4)$$

where α_t, α_g are fading factors for inheritance of Gaussian lifespan (t value used in (3)) and for Gaussian gradients states respectively. The lifespan fading factor α_t allows to control the lifespan inheritance and to control confidence in the reuse of the inherited filtered gradient. α_g allows to scale the gradient inheritance since the gradient amplitude naturally changes when considering a Gaussian child vs a Gaussian mother (e.g. since size of the splatted Gaussians will change, gradient should be scaled). These fading parameters have been set empirically to ($\alpha_t = 1, \alpha_g = 0.2$) for all datasets.

V. IMPROVED DENSIFICATION

A. Effective opacity-based Gaussian pruning

For pruning operations, we propose to use the prioritization metric based on $\mathcal{P}_i^{(c)} = \frac{\sum_p w_i^{(c)}(p)}{S_i^{(c)}}$, where $S_i^{(c)}$ is the surface covered by the i -th Gaussian in c -th camera’s image. $\mathcal{P}_i^{(c)}$ can be interpreted as the effective opacity of the i -th Gaussian. The resulting prioritization metric for pruning is then defined as:

$$\mathcal{P}_{prune}(i) = \max_c \mathcal{P}_i^c \quad (5)$$

Gaussians to be pruned are the ones having $\mathcal{P}_{prune}$ value below a threshold $\tau_{prune} = 0.02$. To prevent excessive pruning of Gaussians, we restrict the number of pruned Gaussians to a fraction of the current total number of Gaussians (1% in our experiments). This pruning strategy naturally prunes Gaussians with too weak opacity and also those that may get out of view of training cameras. In BOGausS, it replaces opacity reset used in 3DGS. Note that [4] studied the use of a similar criterion but did not keep it because of inconsistent results for indoor and outdoor scenes.

B. Density-preserving Gaussian splitting

As mentioned in [5], [9] the original splitting operation causes perturbations in the loss function. Correction mechanisms on opacity and size parameters of split Gaussian children are then proposed to limit those perturbations. As perturbations also occur during cloning, we choose to discard the cloning operations and rely only on splitting operations.

Inspired from [16] we propose to modify the splitting operation so that the resulting density after splitting is similar to the density of the Gaussian being split. Splitting is performed along the longest principal axis, by shifting means by $\pm\alpha\sigma$, where α is a constant (set to 0.3 in our case) and σ is the size along the splitting axis. To ensure that the resulting density of the sum of the two children is close to the density of the mother Gaussian, we adjust the sizes and opacity of the split Gaussians. This adaptation can be learned offline so that $(\sigma_{child}, o_{child}) = \text{SplitAdaptation}(\sigma_{mother}, o_{mother})$. [17] uses a similar splitting mechanism, but does not take into account opacity variations. This proposed splitting also prevents clusters of Gaussians as observed by [3].

C. Distortion-based prioritization of Gaussians splitting

We propose a prioritization criterion inspired from rate-distortion optimization [7]. Gaussians to be split are the ones that will induce the highest gain in the PSNR quality metric.

First we associate a reconstruction Squared Error $SE_{c,i}$ to each Gaussian that is defined as:

$$SE_{c,i} = \sum_{p \in I_c} w_i^{(c)} (R^{(c)}(p) - I^{(c)}(p))^2 \quad (6)$$

where $I^{(c)}$ corresponds to the ground-truth image acquired by the c -th camera. As in [9], this formulation allows spreading error metric across Gaussians.

When performing a splitting operation, the intent is to increase densification in the associated area in order to reduce reconstruction error. When splitting a Gaussian, the local reconstruction error will decrease from $SE_{c,i}$ to $\alpha_{split} SE_{c,i}$ with $0 < \alpha_{split} < 1$. The resulting variation in PSNR for that camera will then be:

$$\begin{aligned} \Delta PSNR_{c,i} &= -10(\log_{10} SE_c^{(new)} - \log_{10} SE_c^{(prev)}) \\ &= -10 \log_{10} \frac{SE_c - (1 - \alpha_{split}) SE_{c,i}}{SE_c} \\ &\simeq \frac{10}{\ln(10)} \frac{(1 - \alpha_{split}) SE_{c,i}}{SE_c} \end{aligned} \quad (7)$$

Considering that the Squared Error for camera c can be expressed as $SE_c = \sum_j SE_{c,j}$ (assuming that the background

contribution is negligible) and that α_{split} can be considered constant, then we can define the following prioritization densification criterion for Gaussian i :

$$\mathcal{P}_{SNR}(i) = \sum_c \frac{SE_{c,i}}{\sum_j SE_{c,j}} \quad (8)$$

Note that unlike the prioritization criterion proposed in [9] which is based only on the error $\mathcal{P}_{SE}(i) = \sum_c SE_{c,i}$, our proposed criterion is based on relative contributed error.

VI. EXPERIMENTS

A. Experimental setup

We run benchmarks on three established datasets: MipNeRF360 [1], Tanks&Temples [18], and Deep Blending [19], which contain 9, 2, and 2 scenes respectively. We use the same train/test split as the original 3DGS publication and follow-up works. SSIM, PSNR and LPIPS metrics are evaluated. Training hyperparameters have same value for all scenes. We increase the interval between two densification operations to 500 iterations, allowing more time for Gaussians to stabilize and thereby get more reliable prioritization metrics, as proposed in [4]. Evaluations with and without exposure compensation are also proposed (based on technique proposed in [20] which uses an affine color correction per viewpoint).

B. Results

Comparative results are provided in Table I and Fig. 2. Starting from the initial number of Gaussians obtained by vanilla 3DGS [2] for each scene, we run our BOGausS solution for various targeted budgets that are fractions of initial number of Gaussians. Results are compared with Taming [4] and Mini-Splatting [3] at various rates. Available code for MCMC [5] has been run with aforementioned budget targets. Looking at results in Table I and Fig. 2, our method outperforms original 3DGS on all metrics. Even when using 5 times less Gaussians, our method exhibits higher quality. Compared to [3]–[5], BOGausS consistently provides better PSNR, and similar SSIM and LPIPS values. Compared to MCMC, BOGausS may provide only slightly higher PSNR, but it provides significantly higher subjective quality as shown in Fig. 1. Typically, background content is richer with finer details thanks to denser Gaussians in these areas without sacrificing reconstruction of foreground content. The discrepancy between subjective quality gains and objective metrics may be due to camera calibrations issues. For instance when applying exposure correction (such as proposed in [20]), large gains in objective metrics can be observed (see bottom part of Table I). Pose estimation is also a problem: some misalignment can be observed between rendered images and ground-truth images.

C. Ablation study

Table II examines the effect of individually removing several of our contributions. The analysis is performed on Garden sequence with an intermediate number of Gaussians. Removing any of these contributions degrades PSNR performance. Removing inherited optimizer states during splitting operations

does not bring much degradation, but we have observed that it introduces more variations during loss minimization. It also comes with more Gaussians being pruned due to more instabilities. Replacing SNR prioritization with MCMC prioritization mechanism for densification provides slightly better SSIM and LPIPS values, but lower PSNR. Moreover, as can be observed in Fig. 1, it does not have a good behavior in background areas with significant visual artifacts.

VII. CONCLUSION

We have presented a Better Optimized Gaussian Splatting (BOGausS) algorithm for generating high-quality novel view synthesis. Our approach provides significant improvements over previous work allowing to generate Gaussian splatting content with fewer primitives and higher quality. Compared to original 3DGS, BOGausS provides similar quality with 5 times less Gaussians, even 10 times less Gaussians if using exposure correction. Allowing graceful degradation with reduced number of Gaussians is especially beneficial for low-resources systems.

First observations show that some additional gain could be obtained by tuning the new introduced hyperparameters. We left these optimizations for further studies.

REFERENCES

- [1] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields,” *ICCV*, 2021.
- [2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3D gaussian splatting for real-time radiance field rendering,” *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- [3] G. Fang and B. Wang, “Mini-splatting: Representing scenes with a constrained number of gaussians,” in *European Conference on Computer Vision*, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:268554047>
- [4] S. S. Mallick, R. Goel, B. Kerbl, M. Steinberger, F. V. Carrasco, and F. De La Torre, “Taming 3DGS: High-quality radiance fields with limited resources,” in *SIGGRAPH Asia 2024 Conference Papers*, ser. SA ’24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: <https://doi.org/10.1145/3680528.3687694>
- [5] S. Kheradmand, D. Rebain, G. Sharma, W. Sun, Y.-C. Tseng, H. Isack, A. Kar, A. Tagliasacchi, and K. M. Yi, “3D gaussian splatting as markov chain monte carlo,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024, spotlight Presentation.
- [6] S. Girish, K. Gupta, and A. Shrivastava, “EAGLES: Efficient accelerated 3D gaussians with lightweight encodings,” in *Computer Vision – ECCV 2024*, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:266052463>
- [7] G. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.
- [8] M. Zwicker, H. Pfister, J. van Baar, and M. Gross, “EWA volume splatting,” in *Proceedings Visualization, 2001. VIS '01.*, 2001, pp. 29–538.
- [9] S. Rota Bulò, L. Porzi, and P. Kotschieder, “Revising densification in gaussian splatting,” in *Computer Vision – ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds. Cham: Springer Nature Switzerland, 2025, pp. 347–362.
- [10] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, “Mip-splatting: Alias-free 3D gaussian splatting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 19 447–19 456.
- [11] Y. Seo, Y. S. Choi, H. S. Son, and Y. Uh, “FLoD: Integrating flexible level of detail into 3D gaussian splatting for customizable rendering,” 2024. [Online]. Available: <https://arxiv.org/abs/2408.12894>

TABLE I

QUANTITATIVE COMPARISON OF OTHER METHODS WITH OUR TECHNIQUE. UPPER SET OF METHODS IN EACH HALF TARGETS SAME NUMBER OF GAUSSIANS THAN VANILLA 3DGS, LOWER SET TARGETS LOWER NUMBER OF GAUSSIANS. **BEST** AND **SECOND BEST** RESULTS ARE HIGHLIGHTED FOR EACH DATASET AND CATEGORY

	Mip-Nerf 360				Tanks&Temples				Deep Blending			
	SSIM ↑	PSNR ↑	LPIPS ↓	#G.	SSIM ↑	PSNR ↑	LPIPS ↓	#G.	SSIM ↑	PSNR ↑	LPIPS ↓	#G.
Without Exposure Correction												
3DGS [2]	0.816	27.57	0.215	3.31	0.853	23.79	0.169	1.84	0.906	29.69	0.238	2.81
Mini-SplattingD [3]	0.831	27.51	0.176	4.69	0.853	23.23	0.140	4.28	0.906	29.88	0.211	4.63
Taming (big) [4]	0.822	27.79	0.205	3.31	0.851	24.04	0.170	1.84	0.907	30.14	0.235	2.81
MCMC [5]	0.846	28.30	0.174	3.21	0.871	24.58	0.147	1.83	0.905	29.34	0.238	2.80
Ours (Big)	0.844	28.40	0.172	3.21	0.866	24.78	0.176	1.83	0.915	30.41	0.230	2.80
Taming (small) [4]	0.799	27.29	0.253	0.63	0.835	23.89	0.207	0.29	0.902	29.89	0.263	0.27
Mini-Splatting [3]	0.822	27.34	0.217	0.49	0.835	23.18	0.202	0.20	0.908	29.98	0.253	0.35
Ours (Light)	0.805	27.21	0.251	0.32	0.830	23.72	0.250	0.18	0.908	30.18	0.259	0.28
Ours (Medium)	0.836	28.21	0.192	1.28	0.857	24.62	0.199	0.73	0.913	30.37	0.237	1.12
With Exposure Correction												
3DGS [2]	0.818	27.74	0.214	3.31	0.862	24.94	0.164	1.84	0.912	30.59	0.236	2.81
MCMC [5]	0.840	28.36	0.182	3.21	0.877	26.00	0.170	1.83	0.908	30.60	0.237	2.80
Ours (Big)	0.844	28.82	0.171	3.21	0.872	25.99	0.170	1.83	0.915	31.25	0.229	2.80
Ours (Light)	0.804	27.54	0.251	0.32	0.837	24.85	0.244	0.18	0.908	30.96	0.259	0.28
Ours (Medium)	0.836	28.59	0.191	1.28	0.862	25.79	0.193	0.73	0.914	31.24	0.237	1.12

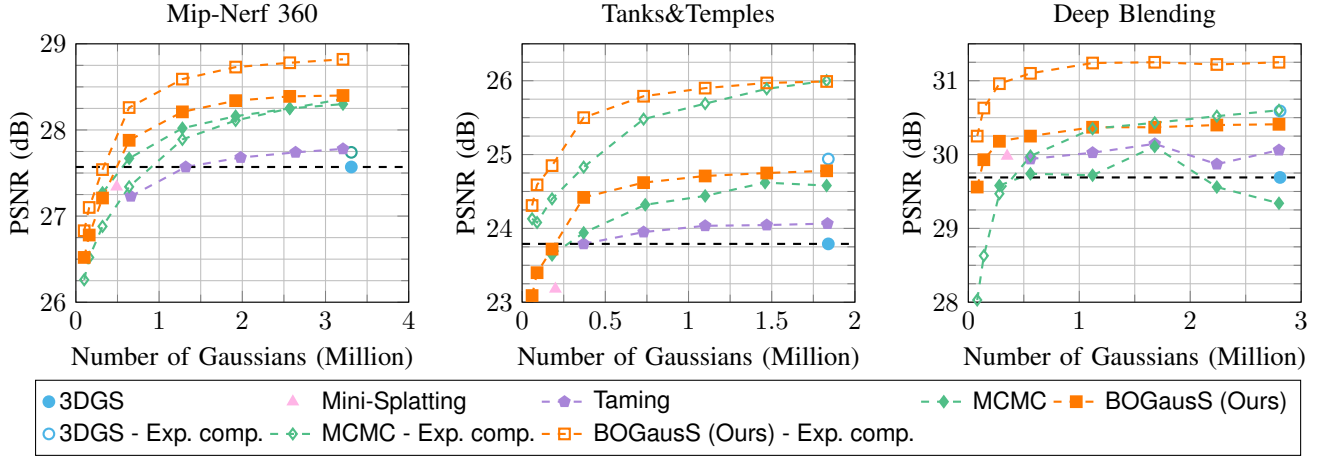


Fig. 2. Comparative PSNR performances vs number of Gaussians. Unfilled-marks plots use exposure compensation while filled-marks plots don't.

TABLE II

ABLATION OF BOGAUSS'S MAIN COMPONENTS. TESTED ON GARDEN SCENE WITH 1.01M GAUSSIANS.

	SSIM ↑	PSNR ↑	LPIPS ↓
BOGausS	0.861	27.81	0.132
no Sparse Adam	0.854	27.49	0.140
no states inheritance at split	0.859	27.77	0.134
no scaled updates	0.854	27.51	0.140
no effective opacity pruning	0.855	27.77	0.1411
no SNR split prioritization	0.863	27.66	0.126

- Organizing-Gaussians/
- [15] Z. Zhang, W. Hu, Y. Lao, T. He, and H. Zhao, "Pixel-GS: Density control with pixel-aware gradient for 3D gaussian splatting," in *ECCV*, 2024.
- [16] U. Hanebeck, K. Briechle, and A. Rauh, "Progressive bayes: A new framework for nonlinear state estimation," *Proceedings of SPIE - The International Society for Optical Engineering*, 04 2003.
- [17] X. Deng, C. Diao, M. Li, R. Yu, and D. Xu, "Efficient density control for 3D gaussian splatting," 11 2024. [Online]. Available: 10.48550/arXiv.2411.10133
- [18] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and Temples: benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017. [Online]. Available: <https://doi.org/10.1145/3072959.3073599>
- [19] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," vol. 37, no. 6, pp. 257:1–257:15, 2018.
- [20] B. Kerbl, A. Meuleman, G. Kopanas, M. Wimmer, A. Lanvin, and G. Drettakis, "A hierarchical 3D gaussian representation for real-time rendering of very large datasets," *ACM Transactions on Graphics*, vol. 43, no. 4, July 2024. [Online]. Available: <https://repo-sam.inria.fr/fungraph/hierarchical-3d-gaussians/>
- [12] Y. Ban, S. Ba, X. Alameda-Pineda, and R. Horaud, "Tracking multiple persons based on a variational bayesian model," in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 52–67.
- [13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:6628106>
- [14] W. Morgenstern, F. Barthel, A. Hilsman, and P. Eisert, "Compact 3D scene representation via self-organizing gaussian grids," in *Computer Vision – ECCV 2024*. Cham: Springer Nature Switzerland, 2025, pp. 18–34. [Online]. Available: <https://fraunhoferhhi.github.io/Self->