Gaussian Semantics: Lifting 2D Semantics to 3D

Noora Andreev-Ahmed, Damla Konur

Abstract

3D Gaussian Splatting (3DGS) enables efficient, high-fidelity reconstruction from posed RGB images, but it does not encode semantic meaning. To bridge this gap, we lift 2D semantic predictions into the 3D representation by using segmentations from a pretrained OneFormer model as render-space supervision. Our approach performs joint optimization of appearance/geometry and per-Gaussian semantic labels, encouraging view-consistent 3D semantics while preserving reconstruction quality. We evaluate the method on ScanNet++, highlighting both the benefits and the challenges of noisy 2D supervision.

1. Introduction

Consistent 3D semantic understanding is critical for robotics and AR, yet high-quality 3D labels are scarce. While 2D segmentation is well-solved, lifting these predictions to 3D is challenging due to temporal inconsistencies, where an image segmentation model mislabels objects across consecutive frames due to lighting, occlusion or model limitations. To address this, we augment 3D Gaussian Splatting (3DGS) [4] by injecting 2D segmentation cues directly into the optimization process. Our model learns view-consistent, per-Gaussian semantic logits alongside geometry, enabling semantic reconstruction from posed RGB images and 2D masks alone.

Our main contributions are as follows:

- We extend 3DGS with a joint optimization framework that learns per-Gaussian semantic logits alongside geometry to obtain view-consistent class annotations in 3D.
- We use render-space semantic supervision and systematically evaluate the method under both ground-truth (ScanNet++ [7]) and noisy pseudo-ground truth (OneFormer [3]) supervision to quantify the effect of 2D label inconsistencies on 3D semantic stability.
- We demonstrate that softmax-based probabilistic class modeling and entropy regularization improve visual and segmentation performance of joint optimization in semantically complex scenes.

2. Related Work

Recent work on semantics for 3DGS [4] can be grouped into two directions. First, feature-driven methods use 2D image features as input and attach them to Gaussians to enable querying or editing. For example, Semantic Gaussians [2] projects open-vocabulary features into Gaussians and trains an additional model for semantic queries, while Feature 3DGS [9] distills high-dimensional 2D features into Gaussian fields for text-guided manipulation. These approaches primarily target 2D/open-vocabulary querying and are commonly evaluated against other feature-based baselines.

Second, mask-supervised multi-view methods use 2D segmentation masks as supervision to obtain 3D-consistent semantics. SAGA [1] employs SAM-guided Gaussians for segmentation, while Gaussian Grouping [6] learns instance-aware Gaussians and reports both rendered 2D mask metrics and 3D panoptic/semantic scores. Unlike feature-projection methods [2, 9] which target open-vocabulary querying, we focus on learning class logits via render-space supervision, focusing on temporal consistency under noisy inputs.

3. Method

We propose an extension to the standard 3DGS [4] framework to jointly reconstruct the 3D geometry, radiance and semantics of a scene.

3.1. Scene Representation

Following the formulation by Kerbl et al., we represent the scene as a set of 3D Gaussians. To add semantic understanding, we extend the set of learnable attributes of each primitive. A single Gaussian G_i is defined as the tuple:

$$G_i = \{\mu_i, \Sigma_i, \alpha_i, c_i^{rgb}, s_i\},$$

where $\mu_i \in \mathbb{R}^3$ is the position, Σ_i is the covariance matrix, α_i is the opacity, c_i^{rgb} represents view-dependent color coefficients and $s_i \in \mathbb{R}^C$ is the newly added semantic vector, where C corresponds to the number of semantic classes in the scene. We model the semantic vector s_i as view-independent, assuming that an object's semantic identity does not change with viewing direction.

3.2. Semantic Rasterization

To learn the semantic parameters s_i, we modify the tile-based rasterizer of 3DGS [4] to render 2D semantic maps alongside the RGB images. The semantic rendering process follows standard alpha-blending.

For a given pixel p, the accumulated semantic vector $\hat{s}$ is computed by blending the semantic vectors of the ordered Gaussians overlapping that pixel:

$$\hat{s} = \sum_{i \in \mathcal{N}} \alpha_i T_i s_i \tag{1}$$

where α_i is the opacity, $T_i = \prod_{j=1}^{i-1}(1 - \alpha_i)$ is the transmittance and s_i is the semantic vector of the i-th Gaussian.

Weighted Average Normalization

Without normalization, the magnitude of the semantic prediction depends on the accumulated opacity. Rays passing through empty results in diluted aggregated semantic vectors. Hence, we normalize the output by the total accumulated weight. Using the recursive definition of transmittance $T_{i+1} = T_i(1 - \alpha_i)$, the sum of weights form a telescoping series $\sum_{i \in \mathcal{N}} \alpha_i T_i = 1 - T_{final}$ (assuming $T_1 = 1$). Thus, our final normalized rendering equation is

$$\hat{s}_{normalized} = \frac{\sum_{i \in \mathcal{N}} \alpha_i T_i s_i}{1 - T_{final}}. \tag{2}$$

This normalization ensures that the rendered semantic vector ignores empty space behind geometry.

3.3. Semantic Supervision and Class Mapping

To enable semantic reconstruction without ground truth annotations for each frame, we generate pseudo-ground truth masks using OneFormer [3], pre-trained on ADE20K [8] (150 classes). Each Gaussian is augmented with a learnable semantic logit vector of dimension C, where C is the number of semantic classes present in the scene. These vectors are initialized with small random values and optimized jointly with the geometric attributes during training.

Since ADE20K [8] and ScanNet++ [7] use different label vocabularies, we construct an explicit mapping between the two. Each ADE20K class is associated with semantically equivalent ScanNet++ class(es) (e.g., “chair” → {“chair”, “office chair”}), with word-boundary matching as a fallback. Outdoor classes without indoor equivalents (e.g., “sky”, “mountain”) are marked as unmapped and excluded from evaluation. This mapping enables us to evaluate our semantic predictions against ScanNet++ [7] ground truth annotations by restricting the comparison to the intersection of classes present in both label sets. We report metrics only over these common classes, ensuring a fair comparison that accounts for the inherent domain gap between OneFormer [3] and the ScanNet++ annotation schema.

3.4. Optimization and Loss Function

We optimize the parameters of all Gaussians using a combined loss function that penalizes discrepancies in visual appearance and semantic classification. By performing joint optimization, we allow semantic gradients to backpropagate into geometric attributes. This means that semantic information can improve the reconstruction of a scene's geometry.

The total loss $\mathcal{L}$ is a weighted sum of the photometric loss $\mathcal{L}_{RGB}$, the semantic loss $\mathcal{L}_{sem}$ and a regularization term $\mathcal{L}_{ent}$,

$$\mathcal{L} = \mathcal{L}_{RGB} + \lambda_{sem}\mathcal{L}_{sem} + \lambda_H \mathcal{L}_{ent} \tag{3}$$

where λ_{sem} and λ_H are hyperparameters. We trained with $\lambda_{sem} = 1$.

Photometric Loss and Semantic Loss

The photometric loss $\mathcal{L}_{RGB}$ follows the standard 3DGS formulation [4] combining L1 and SSIM [5]. For semantic supervision, we compare the normalized rendered semantic vectors $\hat{s}_{normalized}$ against the re-mapped pseudo-ground truth masks generated by OneFormer [3] using a standard Cross-Entropy loss.

Gaussian Entropy Regularization

To improve the consistency of the semantic predictions, we introduce an entropy regularization term applied directly to the per-Gaussian semantic vectors. To goal is to encourage Gaussians to make confident (low-entropy) semantic predictions, pushing the class distribution of each primitive to be peaked.

Each learnable semantic vector $s_i \in \mathbb{R}^C$ of Gaussian i is a logit. We compute the class probability distribution p_i via the softmax function

$$p_i = \text{softmax}(s_i) \in \mathbb{R}^C. \tag{4}$$

The entropy $H(p_i)$ for a single Gaussian is defined as

$$H(p_i) = \sum_{k=1}^{C} p_{i,k} \log(p_{i,k}) \tag{5}$$

where a high entropy indicates uncertainty (probabilities spread across classes) and low entropy indicates confidence (probability concentrated on a single class).

The final regularization term $\mathcal{L}_{ent}$ in Equation 3 is the expectation of the entropy over all N Gaussians in the scene,

$$\mathcal{L}_{ent} = \mathbb{E}_{\text{Gaussians}}[H(p)] = \frac{1}{N}\sum_{i=1}^{N} H(p_i). \tag{6}$$

Table 1. Per-scene results for two mask sources. RGB: PSNR↑, SSIM↑, LPIPS↓. Semantics: mIoU↑. Gaussian Grouping (GG) is only evaluated for OneFormer inputs.

Mask Source	Scene (#cls)	Gaussian Grouping (Baseline)				Ours (Joint Opt)			
		PSNR	SSIM	LPIPS	mIoU	PSNR	SSIM	LPIPS	mIoU
ScanNet++	3e8bba0176 (28)	—				26.37	0.889	0.23	0.92
	8e6ff28354 (22)	—				24.74	0.820	0.30	0.90
	7831862f02 (14)	—				30.62	0.900	0.18	0.99
	f36e3e1e53 (16)	—				26.69	0.910	0.20	0.94
OneFormer	3e8bba0176 (17)	15.09	0.59	0.58	0.13	25.75	0.790	0.24	0.56
	8e6ff28354 (15)	15.57	0.49	0.61	0.12	21.99	0.750	0.38	0.51
	7831862f02 (11)	15.49	0.54	0.59	0.17	23.49	0.850	0.27	0.49
	f36e3e1e53 (11)	16.94	0.65	0.54	0.21	22.53	0.830	0.34	0.54

This term encourages sharper, more consistent semantics at the primitive level. To prevent instability during early training iterations and to allow the model to converge on the geometry, we apply a linear warmup to the weight λ_H, growing it from zero over the first 5k iterations.

4. Results

We evaluated our method on four indoor scenes from the ScanNet++ [7] dataset (3e8bba0176, 8e6ff28354, 7831862f02, and f36e3e1e53). We selected these scenes to represent varying levels of semantic clutter and difficulty of geometry, e.g., offices are sparser, while living spaces are more cluttered and densely filled. To validate the results of our joint optimization pipeline, we evaluate our method in 2D against a baseline and against our method supervised with ScanNet++ [7] ground truth masks. We adapted the Gaussian Grouping [6] instance segmentation model as our baseline. Running our method with OneFormer [3] supervision tests our method's ability to account for temporally inconsistent 2D segmentation maps, i.e., to fuse noisy, inconsistent pseudo-ground truth masks into a consistent scene representation. The ScanNet++ supervision uses projected high-quality ground-truth annotations from ScanNet++ to establish an upper bound on our architecture's performance. For evaluation, we report mIoU (mean Intersection over Union) to measure semantic accuracy. To see the effect of jointly optimized semantics and visuals, we also report the standard visual metrics: PSNR, SSIM and LPIPS.

4.1. Quantitative Results

Table 1 summarizes per-scene results for both supervision sources (ScanNet++ [7] ground truth and OneFormer [3] predictions). Under ScanNet++ supervision, our joint optimization achieves consistently high semantic accuracy across all scenes while maintaining strong visual quality. Using OneFormer masks as supervision is substantially more challenging: mIoU drops across all scenes due to noisy and view-inconsistent pseudo-labels, but our method still produces stable semantics in the mid-range, indicating successful multi-view fusion of inconsistent 2D predictions.

We also compare against Gaussian Grouping as a baseline on OneFormer inputs. As shown in Table 1, Gaussian Grouping [6] yields markedly lower semantic scores on these scenes, while our joint optimization improves mIoU across all four scenes. Finally, we observe different trends in appearance quality depending on the supervision source. Under ScanNet++ [7] supervision, RGB metrics slightly improve compared to standard 3DGS, which we attribute to the higher Gaussian count in our joint optimization setting, increasing representational capacity. In contrast, with OneFormer [3] supervision the RGB metrics drop, likely because noisy and temporally inconsistent masks introduce conflicting gradients during joint optimization.

4.2. Qualitative Results

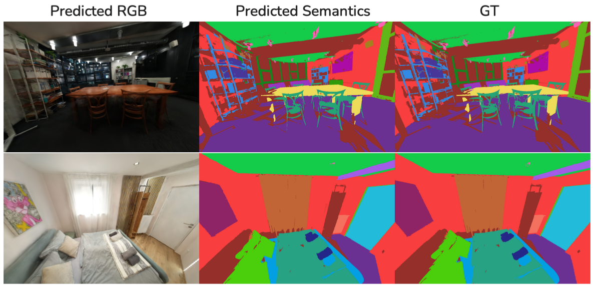
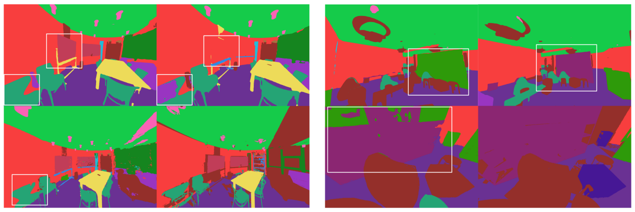


Figure 1. Qualitative results with ScanNet++ ground-truth supervision. Left: rendered RGB from our optimized 3DGS; middle: rendered semantic predictions; right: ScanNet++ ground-truth semantics. Top row: scene `8e6ff28354`. Bottom row: scene `f36e3e1e53`.

Figure 1 shows qualitative results under ScanNet++ ground-truth supervision. The rendered semantics closely match the ground truth for both scenes, and the RGB renderings remain sharp; consistent with our quantitative results, appearance quality does not degrade and can slightly



(**a**) Example 1

(**b**) Example 2

Figure 2. Effect of temporally inconsistent 2D supervision. Top: two consecutive OneFormer semantic predictions used as input to our method, illustrating frame-to-frame label flicker (highlighted regions). Bottom: our predicted semantics (left) vs. ScanNet++ ground truth (right).

improve due to the increased Gaussian capacity during joint optimization.

Figure 2 analyzes temporally inconsistent OneFormer supervision. In Example 1 (left), chair labels flicker between consecutive frames, but the joint multi-view optimization recovers a stable labeling (bottom-left). In Example 2 (right), the shelf is misclassified more consistently across views, and this error propagates into the optimized output, leading to remaining flicker in the bottom-right prediction. This highlights a key limitation: our method cannot correct cases where the 2D supervision is wrong in the majority of views.

4.3. Ablations studies

We conducted an ablation study evaluating the effects of Softmax placement and Gaussian Entropy Regularization.

In our baseline implementation, we store semantic logits on each Gaussian and apply Softmax to each logit after alpha-blending. We compared this to a variant in which we apply Softmax to each Gaussian's semantic vector before alpha-blending, making each primitive's semantic vector a valid probability distribution.

Additionally, we evaluate the effect of entropy regularization $\mathcal{L}_{ent}$ as defined in Section 3.4, which penalizes uncertainty and encourages primitives to make confident semantic predictions.

Table 2. Ablation Study: Softmax and Entropy Regularization

Scene	Method	PSNR	SSIM	mIoU (%)	Acc. (%)
38bba0176	Original (Baseline)	**24.74**	**0.822**	**90.1**	**97.9**
	+ Softmax	26.39	0.890	92.0	98.8
	+ Entropy ($w = 0.01$)	26.28	0.888	92.5	98.8
	+ Entropy ($w = 0.05$)	26.28	0.887	91.7	98.8
7831862f02	Original (Baseline)	**30.62**	**0.903**	**99.4**	**99.9**
	+ Softmax	27.42	0.828	95.4	99.2
	+ Entropy ($w = 0.01$)	27.32	0.827	95.9	99.1
	+ Entropy ($w = 0.05$)	27.33	0.826	96.0	99.1

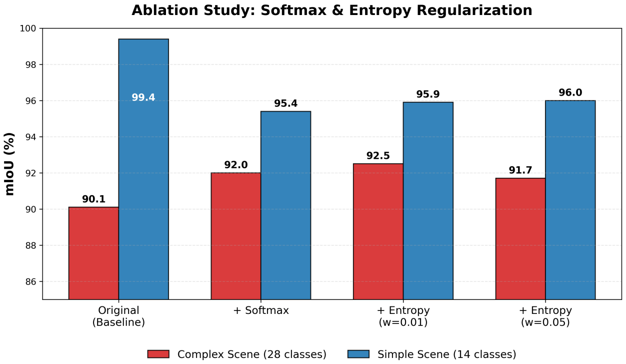


Figure 3. mIoU with Softmax placement and Entropy Regularization evaluated on two scenes.

These experiments were performed with ScanNet++ ground truth supervision to isolate the results from noisy 2D supervision. As shown in Table 2 and Figure 3, the effectiveness of these contributions depends on the geometric and semantic complexity of the scene. In the complex scene (38bba0176, 28 semantic classes), applying Softmax before aggregation stabilized training, while entropy regularization ($\lambda_H = 0.01$) sharpened semantic boundaries, achieving the best performance. However, in the simpler scene (7831862f02, 14 semantic classes), which was already solved by the base joint optimization ($> 99\%$ mIoU) these additional constraints caused a slight regression across visual metrics and mIoU. In simpler scenes, standard logit aggregation works well. This suggests that our regularization is helpful for complex scenes, but unnecessary for simpler scenes, where the base joint optimization is sufficient.

5. Conclusion

In this work, we presented a method to lift 2D semantic predictions into a consistent 3D Gaussian Splatting representation via joint optimization. While our approach successfully produces 3D-consistent semantics from multi-view supervision, several limitations remain. First, reducing temporal flicker via multi-view agreement assumes the 2D supervision is correct on average. If the 2D model consistently mislabels an object across most views, the optimization will reinforce the wrong class rather than recover the correct one. Second, there is a trade-off between semantic accuracy and photometric fidelity: gradients from the semantic loss $\mathcal{L}_{sem}$ can conflict with the reconstruction loss $\mathcal{L}_{RGB}$. In practice, strong semantic gradients near boundaries may trigger additional densification, introducing geometric artifacts and slightly lowering PSNR compared to standard 3DGS. Finally, our explicit representation stores a learnable C-dimensional vector per Gaussian, yielding a memory cost of $\mathcal{O}(N \cdot C)$. For large label sets and many Gaussians, this can require substantial VRAM. Code: https://github.com/damlakonur/sem3dgs

References

[1] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians, 2025. 1

[2] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting, 2024. 1

[3] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. 2023. 1, 2, 3

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 1, 2

[5] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 2

[6] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024. 1, 3

[7] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3

[8] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[9] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields, 2024. 1