Semantic Supervision and Representation Design in 3D Gaussian Splatting for Urban Scene Understanding

by

Hugo E. Chassagnette

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday April 2, 2026, at 09:00.

Student number:	4770897	
Faculty:	Faculty of Mechanical Engineering, Delft	
Track:	Robotics	
Thesis committee:	Dr Holger Caesar,	Department of Intelligent Vehicles, supervisor
	Michael Weinmannn,	Department of Intelligent Vehicles

TU Delft

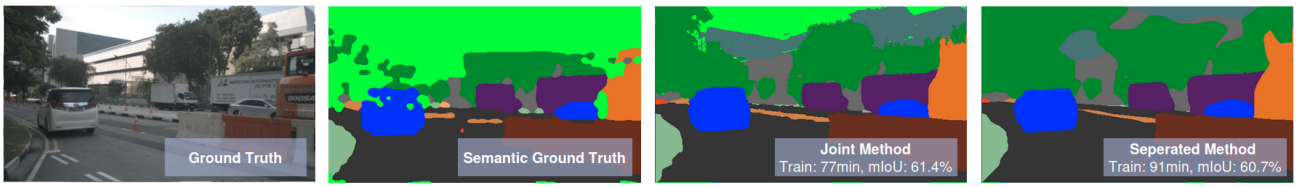


Figure 1: Visualisation of the semantic map renders obtained from the best configurations for the Joint and Separated Gaussian methods, along their average training time and average test dataset's mIoU. Compared to the RGB and semantic ground-truths.(Source: nuScenes dataset [1], used under CC BY-SA 4.0.)

Abstract. 3D Gaussian Splatting (3DGS) has recently emerged as an efficient method for high-fidelity scene reconstruction in autonomous driving environments. While semantic information has been incorporated into Gaussian based representations for scene understanding tasks, it's broader potential for influencing the training process remains unexplored. This thesis investigates how semantic supervision can be integrated into 3DGS training through several semantic-aware strategies, including alternative semantic loss functions, weighting schemes, and semantic-guided densification mechanisms. In addition, we explore different ways of organising RGB and semantic information within the representation. Since RGB appearance and semantic information differ in complexity, we compare a joint Gaussian representation, where RGB and semantic supervision act on the same primitives, with a separated Gaussian representation, where semantic information is modelled by an independent Gaussian set. Experimental results show that the choice of semantic classification loss is the dominant factor influencing semantic performance, while auxiliary strategies do not provide significant improvements. Furthermore, we observe a clear trade-off between representation designs: the joint representation achieves stronger semantic performance but at the cost of degradation in RGB reconstruction quality, whereas the separated representations preserves RGB fidelity with minimal degradation while still achieving good semantic performance. These findings highlight the trade-offs between representations and motivate the exploration of hybrid organisations that better balance RGB reconstruction quality and semantic performance.

1 Introduction

In the field of 3D scene reconstruction, neural scene representations have led to significant advances in recent years for reconstruction and novel view synthesis. Early implicit methods such as Neural Radiance Fields (NeRF)[2] have shown that photorealistic view synthesis can be achieved through volumetric rendering of neural implicit functions. However, the high computational cost of sampling a large number of points across the full 3D space limits their applicability in real-time or large-scale scenarios.

The introduction of 3D Gaussian Splatting (3DGS)[3] provided an explicit alternative by representing scenes as collections of anisotropic Gaussian primitives that are directly optimised in the 3D space. Through differentiable splatting and adaptive densification, 3DGS is able to achieve high-quality real-time rendering. Its efficiency and explicit nature have made it attractive for large-scale urban reconstruction and autonomous driving applications.

Recent works such as Street Gaussians[4] and CoDa-4DGS[5] demonstrate the applicability of Gaussian based representations to dynamic urban scenes. Frameworks such as OmniRe[6] further enable large-scale urban reconstruction pipelines, while simulators including SplatAD[7] highlight the potential of 3DGS for simulation and safety validation. These developments place Gaussian Splatting as a strong candidate for real-time, scalable scene modelling in safety-critical domains.

While the quality of renderings has been the primary focus of these recent works, many real-world applications require more than photorealistic reconstruction. In autonomous driving, robotics, and simulation, semantic understanding of the environment is essential for object-level reasoning, task-aware rendering, and downstream perception tasks.

1.1 Problem Statement

The integration of semantic information into Gaussian based representations and it's impact have not been widely studied.

Recent works have demonstrated that semantic information can be embedded into Gaussian primitives. Methods such as Semantic Gaussians[8], GSemSplat[9], HUGS[10], and Open-Ended 3D Metric-Semantic Representation Learning[11] incorporate explicit semantic labels directly into the Gaussian parameters. Other vision-language-based methods such as FMGS[12], Language Embedded 3D Gaussians[13],

and SceneSplat[14] embed high-level semantic representations derived from pre-trained vision-language models into the Gaussian parameters, enabling open-vocabulary or language-guided understanding.

However, these existing methods primarily focus on enabling semantic prediction rather than analysing how semantic integration and supervision influence the training dynamics and structural properties of Gaussian representations. In particular, the impact of different semantic loss formulations on both semantic and RGB rendering quality and the role of weighting strategies remain unexplored. Likewise, density control in 3DGS has not yet been examined from a semantic perspective. Moreover, current semantic Gaussian methods rely on a shared representation in which RGB and semantic objectives jointly optimise the same primitives, without evaluating alternative ways of structuring these representations. As a result, there is limited understanding of how semantic integration and supervision interacts with RGB and semantic reconstruction fidelity.

1.2 Research Motivation

The integration of semantics into 3D Gaussian Splatting requires more than injecting class labels into Gaussian primitives.

Making 3DGS semantically aware can be approached at multiple levels. Different semantic loss formulations introduce distinct gradient behaviours, particularly in class-imbalanced driving scenes. Weighting strategies determine how strongly semantic supervision influences optimisation and when it is introduced during training, while also being able to address class imbalance and boundary ambiguity through class-based or boundary-aware weighting. Densification strategies can use semantic importance as guidance, enabling the semantically informed allocation of Gaussian.

Finally, semantic supervision introduces a multi-task optimisation problem. Appearance reconstruction and semantic classification are fundamentally different objectives. When both tasks share the same Gaussians, semantic gradients may influence geometric and photometric parameters, potentially leading to objective conflicts. Alternatively, separating semantic and RGB Gaussian sets removes gradient interference but introduces representational redundancy and removes the potential mutual benefits of shared optimisation. Understanding this trade-off is essential to determine how semantics should be integrated into 3D Gaussian Splatting.

A systematic investigation of semantic-aware training strategies and representation organisation is therefore necessary to understand how semantic integration and supervision affects Gaussian reconstruction and how different representation structures influence task interaction and results.

1.3 Research Question

This thesis addresses the following research question:

How do different semantic-aware training strategies and representation organisations influence semantic and RGB rendering quality in 3D Gaussian Splatting?

To answer this question, this work evaluates multiple semantic loss formulations, weighting strategies, and semantic-aware densification strategies, as well as two Gaussian organisations: a joint Gaussian representation and a separated Gaussian representation.

The evaluation is conducted on the Street Gaussians implementation of the OmniRe framework on large-scale urban driving data, measuring both RGB rendering quality and semantic rendering performance.

1.4 Contribution

This thesis makes the following contributions:

1. A systematic study of semantic-aware integration strategies for 3D Gaussian Splatting, examining alternative loss formulations, weighting schemes, and confidence-based densification mechanisms.

2. A comparative analysis of representation organisation, contrasting joint Gaussians with separated Gaussians representations to assess task interference.

3. A comprehensive experimental evaluation framework measuring semantic rendering quality, RGB reconstruction fidelity, and computational efficiency in large-scale urban driving scenes.

Through this structured evaluation, the thesis provides an analysis of how semantic supervision interacts with Gaussian based scene representations and clarifies the trade-offs involved in integrating semantics into real-time 3D Gaussian Splatting.

2 Related Work

In this section we review prior works relevant to 3D Gaussian Splatting, semantic rendering, multi-task representation learning, and adaptive density control. Together, they provide the foundation for analysing how semantic integration interacts with Gaussian based scene reconstruction.

2.1 3D Gaussian Splatting

3D Gaussian Splatting (3DGS)[3], established an explicit and efficient alternative to implicit neural radiance field representations. Instead of representing scenes as continuous neural functions rendered through volumetric rendering, 3DGS represents scenes as collections of anisotropic Gaussian primitives directly optimised in 3D space and rendered via differentiable splatting, allowing real-time rendering while maintaining competitive visual quality.

After its introduction, numerous works have aimed to extend 3DGS capabilities to large-scale and dynamic environments. Street Gaussians[4] model dynamic urban scenes with object level decomposition, while others such as DrivingGaussian[15], VDG[16], and CoDa-4DGS[5] extend Gaussian representations to handle dynamic motion in autonomous driving scenarios. Frameworks like OmniRe[6] provide scalable pipelines for urban scene reconstruction. Meanwhile the applicability of Gaussian-based rendering for driving simulation is demonstrated by simulation oriented systems including GSAVS[17], SplatAD[7], and LiHi-GS[18].

Beyond driving contexts, research such as Deformable 3D Gaussians[19] and Periodic Vibration Gaussian[20] incorporate dynamic deformation modelling into Gaussian primitives, further illustrating the flexibility of Gaussian based representations.

These works collectively demonstrate the scalability, efficiency, and adaptability of 3DGS for complex real-world environments.

However, even with significant progress made in geometry, dynamics, and rendering efficiency, systematic integration and analysis of semantic supervision within Gaussian based representations remains limited.

2.2 Semantic Radiance Fields/Semantic NeRF

Semantic scene understanding has been widely studied in implicit radiance field models. Semantic NeRF variants [21], [22] extend appearance-based neural rendering by jointly predicting semantic labels and RGB values, typically supervised with pixel-based losses such as cross-entropy or focal loss. These works demonstrate that semantic supervision can be integrated into radiance field optimisation without fundamentally changing volumetric rendering pipelines.

With the transition to Gaussian-based representations, several semantic 3DGS methods have emerged. Semantic Gaussians [8] introduce open-vocabulary semantic distillation into Gaussian primitives through alignment with pre-trained models. Other approaches include GSemSplat [9], which enables semantic Gaussian Splatting from limited or uncalibrated inputs; HUGS [10], which integrates holistic scene understanding; Open-Ended 3D Metric-Semantic Representation Learning [11], which learns joint metric-semantic embeddings; and OpenGS-SLAM [23], which incorporates semantic Gaussians into dense SLAM pipelines.

Several works further enrich Gaussian representations using foundation model priors. Methods such as FMGS [12], Language Embedded 3D Gaussians [13], SceneSplat [14], and Feature 3DGS [24] embed pre-trained feature representations, often derived from CLIP-like models, into Gaussian parameters to enable open-vocabulary or feature-level reasoning.

Together, these studies show that semantic attributes and high-dimensional features can be distilled into Gaussian primitives. However, most focus on enabling semantic prediction rather than analysing how different semantic supervision strategies influence training dynamics or affect semantic and RGB rendering quality.

2.3 Multi-task Learning & Representation Sharing

Joint optimisation of RGB reconstruction and semantic classification introduces challenges commonly studied in multi-task learning. When multiple objectives share a representation, gradients from different tasks may interfere with each other, potentially causing optimisation instability or degraded performance for one or more objectives. Multi-task learning literature addresses this issue through techniques such as loss balancing and task-specific weighting to reduce negative interference [25], [26].

Most semantic NeRF and semantic 3DGS methods follow a shared representation paradigm, where semantic and RGB outputs are predicted from the same underlying scene parameters [8], [10]. This design enables semantic attributes to be learned alongside appearance reconstruction while keeping the architecture simple. However, it implicitly assumes that the objectives are compatible during optimisation and does not explicitly address potential task interference.

Alternative representation organisations remain largely unexplored in Gaussian Splatting. One possible direction is to maintain separate parameter sets for semantic and RGB modelling so each task can evolve independently. Although recent work has investigated decoupled semantic and RGB representations for improved memory efficiency through hierarchical feature compression [27], the focus is on representation efficiency rather than optimisation dynamics. Whether semantic and RGB objectives benefit more from shared primitives or task-specific representations therefore remains an open question in Gaussian-based scene modelling.

2.4 Adaptive Density/Importance Sampling

Adaptive densification is a core component of 3D Gaussian Splatting. The original method proposes gradient based duplication and pruning to allocate more Gaussians to regions with high photometric error. Since then, numerous works have refined densification strategies to improve reconstruction quality and computational efficiency.

Revising Densification in Gaussian Splatting [28] revisits the original densification strategy of 3DGS and proposes improved splitting and pruning heuristics to stabilise training and improve reconstruction quality. Pixel-GS[29] introduces pixel-aware gradient signals to guide density allocation. Color-Cued Efficient Densification[30] and Frequency-Aware Density Control[31] incorporate appearance based cues to better capture fine details. More recent approaches such as Efficient Density Control[32], Improved Adaptive Density Control[33], and Generative Densification[34] propose alternative heuristics or learned strategies for balancing representation compactness and fidelity.

Across these methods, densification criteria are primarily driven by geometric, gradient, or photometric importance measures. While these strategies effectively improve visual quality, they do not account for semantic relevance. Integrating semantic importance into Gaussian allocation therefore represents a natural yet unexplored extension of adaptive density control in 3DGS.

3 Method

In this section, the methodology used to address the research question is presented. First the base framework upon which this research is built is introduced. This is followed by a description of the Gaussian organisations evaluated in this study. Finally, the implemented semantic-aware methods are discussed, providing insight into their objectives and implementation.

3.1 Base Framework

This work builds upon the Street Gaussians framework[4], implemented within the OmniRe reconstruction pipeline[6].

In 3DGS[3], a scene is represented as a set of anisotropic Gaussian primitives:

$$\mathcal{G} = \{g_i\}_{i=1}^{N} \tag{1}$$

Each Gaussian g_i is parametrised by a spatial mean $\mu_i \in \mathbb{R}^3$, covariance matrix $\Sigma_i \in \mathbb{R}^{3\times3}$, opacity $\alpha_i \in [0,1]$, and view-dependent colour parameters c_i:

$$g_i = \{\mu_i, \Sigma_i, \alpha_i, c_i\}. \tag{2}$$

Rendering is performed through differentiable splatting, where Gaussians are projected onto the image plane and composited using alpha blending. Then the baseline photometric objective minimises the difference between rendered and ground-truth images:

$$\mathcal{L}_{rgb} = \|\hat{I}_{rgb} - I_{gt}\|_1. \tag{3}$$

During training, the Gaussian representation is then dynamically refined through adaptive densification, if an accumulated gradient magnitude exceeds a certain threshold, Gaussians may be duplicated or split to try increase the reconstruction quality in complex regions.

While this framework is optimised purely for RGB reconstruction, the goal of this work is to introduce semantic supervision and investigate how different semantic-aware strategies influence Gaussian optimisation and reconstruction quality.

3.2 Gaussian Organisation

To integrate semantic understanding into Gaussian Splatting, we investigate two representation organisations: joint Gaussians and separated Gaussians. A simplified diagram of the framework combining both possible organisation representations can be found in Figure 2.

3.2.1 Joint Gaussians Method

In the joint Gaussians method, the semantic information is directly embedded within the RGB Gaussians. Each Gaussian is injected with semantic logits:

$$g_i = \{\mu_i, \Sigma_i, \alpha_i, c_i, s_i\} \tag{4}$$

where $s_i \in \mathbb{R}^K$ represents class logits for K semantic categories.

Rendering therefore produces both an RGB image $\hat{I}_{rgb}$ and a semantic prediction map $\hat{I}_{sem}$. Since both outputs are predicted from the same Gaussian primitives, RGB and semantic losses both influence the optimisation of Gaussian geometry and appearance parameters. A flow chart illustrating this method can be found in Appendix B Figure 7.

This shared representation leads to a more compact scene representations but introduces optimisation interference between semantic and RGB objectives.

3.2.2 Separated Gaussians Method

To investigate the impact of task decoupling, we also evaluate a method in which semantic and RGB information are modelled by independent Gaussian sets.

Two Gaussian sets are initialised:

$$\mathcal{G}_{rgb}, \quad \mathcal{G}_{sem} \tag{5}$$

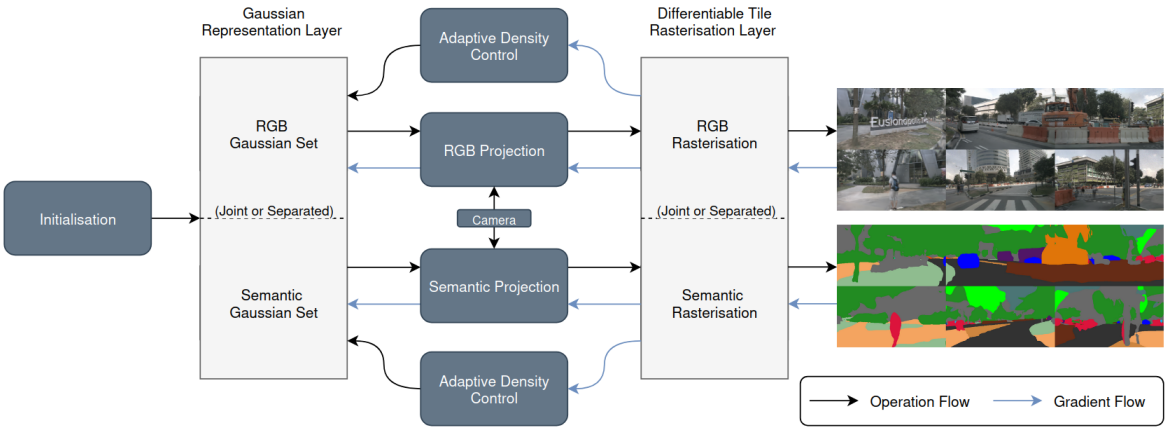


Figure 2: Simplified flow diagram unifying both the joint and separated Gaussian method in a single framework. This illustrate how the RGB and semantic renders are obtained and the training process. Depending on the method selected the representation and rasterisation layers will either be joint (single entity) or separated (two independent entities).(Source: nuScenes dataset [1], used under CC BY-SA 4.0.)

The RGB Gaussians are optimised using only the RGB loss, while semantic Gaussians are optimised using the semantic loss. This organisation ensures resulting gradients only interact with their respective Gaussian set, allowing both representations to evolve independently. A flow chart illustrating this method can be found in Appendix B Figure 8.

Additionally, we further analyse the representational capacity required for semantic reconstruction. Since semantic Gaussians are dedicated exclusively to classification rather than being combine with RGB rendering, their required density differs.

3.3 Training Objective

Both representation organisations are optimised using a combination of RGB reconstruction and semantic supervision. The overall training objective is defined as

$$\mathcal{L}_{total} = \mathcal{L}_{rgb} + \lambda_{main}(t)\mathcal{L}^{w}_{main} + \lambda_{clip}\mathcal{L}_{clip} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{depth}\mathcal{L}_{depth}. \tag{6}$$

Where the primary semantic classification objective $\mathcal{L}_{main}$ is selected as either cross-entropy or focal loss:

$$\mathcal{L}_{main} \in \{\mathcal{L}_{CE}, \mathcal{L}_{Focal}\}. \tag{7}$$

Additional auxiliary losses may be enabled depending on the evaluated configuration, this includes regularisation, CLIP-based feature alignment, and depth supervision.

3.4 Semantic-Aware Training Methods

We investigate several strategies for incorporating semantic supervision into Gaussian optimisation. These strategies modify the training objective, introduce auxiliary supervision signals, and leverage semantic information to guide densification. Mathematical formulations of all losses and weighting schemes are provided in Appendix C.

3.4.1 Semantic Loss

Semantic supervision is applied through pixel-wise classification losses that compare predicted semantic probabilities with ground-truth labels. As a baseline, we employ standard cross-entropy classification.

Driving datasets are highly imbalanced, with large classes such as road and vegetation dominating the pixel distribution. To mitigate this effect, we additionally evaluate focal loss, which down-weights well-classified samples and emphasises difficult or under-represented classes.

We further investigate regularisation strategies aimed at stabilising semantic predictions. Entropy regularisation discourages overly confident predictions by encouraging smoother probability distributions, while L2 regularisation constrains the magnitude of semantic logits.

Finally, we explore CLIP-based supervision as a higher-level semantic signal. In this setting, predicted semantic features are encouraged to align with feature embeddings extracted from a pre-trained vision–language model. This provides global semantic guidance that complements pixel-level supervision.

3.4.2 Depth Supervision

The original RGB Gaussian representation incorporates LiDAR depth supervision derived from rasterised Gaussian depth maps. We extend this supervision to the semantic Gaussian representation used in the separated Gaussian method.

Specifically, the semantic Gaussian set produces a depth map through the same rasterisation process as the RGB representation. This predicted depth is then compared against ground-truth LiDAR depth to enforce geometric consistency between the semantic and photometric representations.

3.4.3 Weighting Strategies

We evaluate several weighting strategies to improve the effectiveness of semantic supervision.

First, to address class imbalance, we apply class-dependent weighting to the semantic classification loss. This increases the contribution of rare classes during optimisation.

Second, we introduce boundary-aware weighting to emphasise pixels near semantic class transitions. A boundary mask is computed from the ground-truth segmentation map and used to increase the influence of pixels located near semantic edges. This encourages sharper semantic boundaries.

Finally, semantic supervision can destabilise early optimisation for the joint Gaussian representation where RGB and semantic parameters are trained simultaneously. To mitigate this, we optionally apply a warm-up schedule that gradually increases the weight of the semantic loss during the early stages of training.

3.4.4 Semantic-Aware Densification

Standard Gaussian densification relies primarily on RGB gradients to determine where representational capacity should be increased. However, semantic predictions provide an additional signal regarding the structural importance of different regions.

We therefore compute a semantic importance score for each Gaussian using its predicted class label, classification confidence, and class-frequency weighting. This score reflects how confidently the model assigns semantic meaning to a region while accounting for imbalance in the Gaussian class distribution.

The semantic importance score is used to guide three structural operations during training: culling, splitting, and duplication of Gaussians. Low-importance Gaussians may be removed, while high-importance Gaussians may be split or duplicated to increase representational capacity. Through this mechanism, semantic information directly influences how Gaussian capacity is allocated across the scene.

4 Experiment

In this section, the experiments conducted to evaluate the proposed approach are presented. It begins with a description of the experimental setup, including the data, implementation details, and relevant configurations. Next, the evaluation metrics used to assess performance are introduced. The section then examines the impact of the semantic-aware methods, followed by an analysis of semantic Gaussian capacity. A comparison of different Gaussian representations is subsequently provided. Finally, the limitation of the approach are discussed, offering context for the interpretation of the results.

4.1 Experimental Setup

All experiments were evaluated on 10 urban driving scenes from the nuScenes[1] dataset. For each scene, a train-test split was created by removing every 10th frame from the training sequence and assigning it to the test set. After training, the model is also used to generate a novel view, enabling an additional semantic evaluation on unseen perspectives beyond the initial camera trajectory.

Two baselines were used as references. The photometric baseline corresponds to the original StreetGS implementation within OmniRe, which we refer to as "Vanilla". The semantic baseline is based on the StreetGS implementation with injected semantic logits into the Gaussian primitives, but without semantic supervision during training, this baseline is referred to as "Vanilla Semantics". As expected, this configuration produces extremely poor semantic predictions and therefore serves only as a lower semantic bound.

Rather than evaluating all semantic-aware methods independently, the experiments follow a progressive evaluation strategy. Starting from the semantic baseline, different semantic-aware techniques are introduced sequentially. At each stage, the best-performing configuration is selected and used as the base configuration for the next experiment. This process allows the experimental setup to incrementally build toward an optimal configuration for each method.

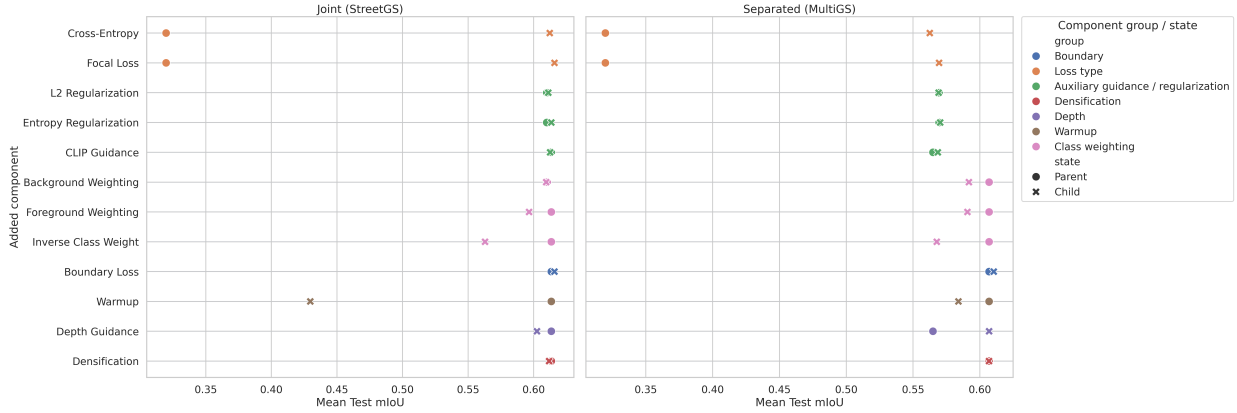
4.2 Evaluation Metrics

We evaluate both photometric reconstruction quality and semantic prediction accuracy.

Photometric quality is measured using Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Learned Perceptual Image Patch Similarity (LPIPS). SSIM measures structural similarity between rendered and ground-truth images, PSNR evaluates pixel-wise reconstruction fidelity, and LPIPS captures perceptual similarity using deep feature representations.

Semantic performance is evaluated using mean Intersection-over-Union (mIoU), which measures the average overlap between predicted and ground-truth semantic segmentation across all classes.

Figure 3: Graph visualising the impact of each semantic-aware component on the test dataset mIoU metric. For each component, the "parent" corresponds to the baseline configuration without that component (or the vanilla Gaussian baseline for CE and Focal), while the "child" corresponds to the best-performing configuration obtained by adding that component. Each parent–child pair is computed from matched configurations (same scene and underlying setup), so the difference directly reflects the isolated effect of introducing that component.



4.3 Impact of Semantic-Aware Methods

The detailed quantitative results for all tested semantic-aware methods and configurations, for both joint and separated Gaussian methods, are reported in tables inside Appendix A. A more concise visual summary of the results is provided in Figures 3 and 4.

The main classification loss, across all experiments, has the largest influence on semantic performance. Replacing cross-entropy with focal loss consistently improves both test and novel-view semantic mIoU, as can be seen for the former in Figure 5. Because of this strong and stable improvement, focal loss is selected as the base classification loss for all subsequent experiments. [Table 1]

Entropy and L2 regularisation are evaluated as mechanisms to stabilise semantic predictions. When added to the focal-loss baseline, both strategies produced minor variations in semantic performance. Improvements were inconsistent across scenes, demonstrating that regularisation has limited influence on the final performance of the model. [Table 2]

CLIP-based feature alignment is introduced to encourage semantic predictions to match pre-trained vision-language embeddings. Although this supervision improved semantic consistency in individual scenes, overall it instead lead to degradations of both semantic and RGB metrics, and was therefore not included in subsequent experiments. [Table 3]

Weighting strategies evaluated include class-based weighting, boundary-aware weighting, and warm-up scheduling. Overall, these strategies did not lead to measurable improvements in the global evaluation metrics. Instead, they produced degradations in both RGB and semantic quality across most scenes. This indicates that the additional weighting schemes introduce extra optimisation complexity without providing clear benefits for the final performance. [Tables 5, 8]

Additionally, applying a warm-up schedule in the joint Gaussian method improves RGB renderings quality but leads to a significant degradation in semantic performance, showing that semantic supervision is too weak in the later training stages making it unable to compensate it's delayed start. [Table 6]

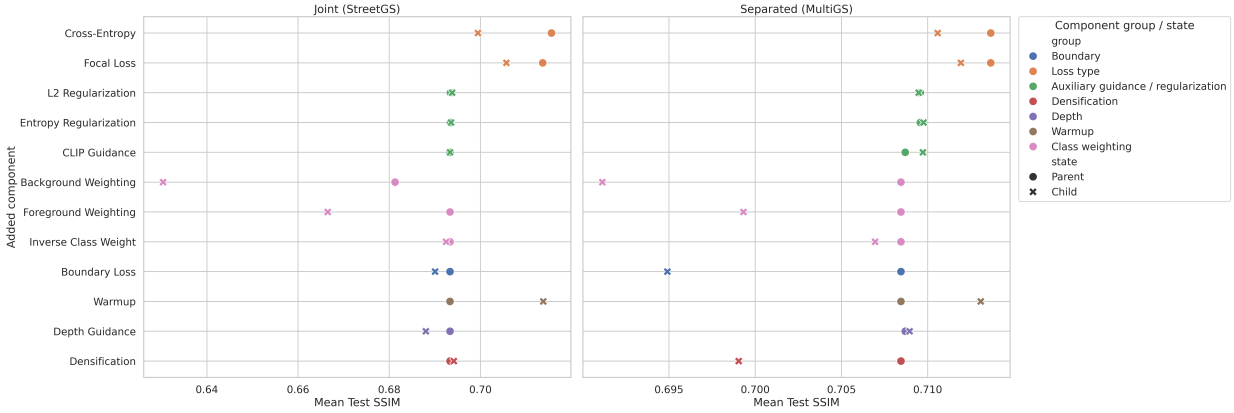
Depth supervision is already included in the original Vanilla implementation of the joint representation. To evaluate its importance, it was removed during training, which, as expected, resulted in a noticeable degradation across all metrics. For the separated Gaussian method, reintroducing depth supervision for the semantic Gaussian set led to measurable improvements, indicating that additional geometric guidance is beneficial. These results highlight the importance of explicit geometric constraints. [Table 4]

Semantic-guided densification produced only negligible changes for both the joint and separated Gaussian representations. This indicates that the existing optimisation signals from the gradients already provide sufficient guidance for Gaussian placement and density, limiting the additional benefit of explicit semantic densification strategies. [Table 7]

4.4 Semantic Gaussian Capacity Analysis

Semantic Gaussian capacity was evaluated by varying the number of initialised semantic Gaussians in the separated Gaussian method. As can be seen in Figure 6, the results show that the initial count can be reduced from 800,000 to 100,000 without any observable degradation in semantic performance. This stability is primarily observed on the training and testing datasets, while the novel-view performance remains

Figure 4: Graph visualising the impact of each semantic-aware component on the test dataset SSIM metric. For each component, the "parent" corresponds to the baseline configuration without that component (or the vanilla Gaussian baseline for CE and Focal), while the "child" corresponds to the best-performing configuration obtained by adding that component. Each parent–child pair is computed from matched configurations (same scene and underlying setup), so the difference directly reflects the isolated effect of introducing that component.



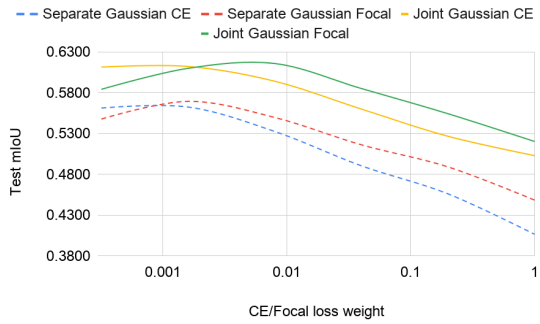


Figure 5: Parameter sweep plot for test dataset's mIoU against the loss weight for CE or Focal loss, for both joint and separated Gaussian method.

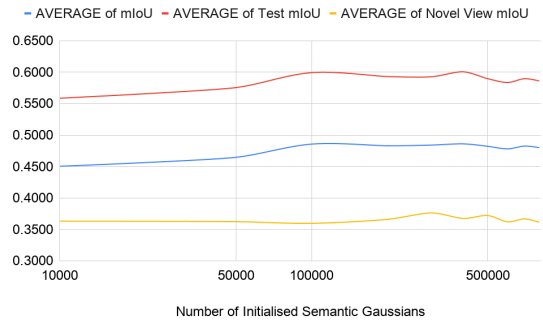


Figure 6: Evolution of the semantic mIoUs for each datasets, averaged over the 10 scenes evaluated, depending on the number of initialised semantic Gaussians.

unaffected. This reduction in capacity is consistent with the lower visual complexity of semantic maps compared to RGB renders, allowing the semantic representation to remain accurate with fewer Gaussians. Additionally, reducing the number of semantic Gaussians slightly decreases training time, providing a modest efficiency benefit. [Table 9]

4.5 Comparison of Gaussian Representations

We now compare the two Gaussian representation organisations used throughout the experiments with their final best configuration found in Table 10.

Across both representations, the choice of the main semantic classification loss remains the most influential factor for semantic performance. In particular, replacing cross-entropy with focal loss consistently produces the largest improvements in semantic metrics, while the other semantic-aware techniques have a smaller impact.

For the joint Gaussian representation, most additional semantic-aware methods do not provide significant benefits for the semantic performance, while introducing RGB degradations. This behaviour is due to the increased optimisation complexity and conflicts between semantic objectives and the RGB reconstruction loss, since both tasks act on the same set of Gaussian primitives.

In the separated Gaussian representation, most semantic-aware methods also do not produce improvements. However, structural guidance mechanisms, specifically depth supervision, plays a more important role. Because semantic Gaussians in this configuration are optimised independently from RGB reconstruction, they lack the structural cues normally provided by photometric gradients. These additional structural constraints therefore help compensate for the absence of RGB driven guidance.

From a qualitative perspective, the semantic map renders produced by the joint and separated Gaussian methods show distinct characteristics, as illustrated

in Figure 1. The joint method appears noisier but generally provides more precise semantic boundaries, whereas the separated method produces smoother results that appear less precise. This indicates once again that the RGB guidance in the joint method play an important role in refining semantic predictions. However, because semantic rendering is inherently less complex than RGB rendering, the separated method is still able to achieve strong results despite the absence of direct RGB guidance.

From a global perspective, the joint representation achieves slightly higher semantic performance and marginally faster training. However, this comes at the cost of a significant degradation in RGB reconstruction quality due to the interaction between semantic and photometric objectives. In contrast, the separated representation produces slightly lower semantic metrics and is marginally slower, but introduces only negligible degradation in RGB quality.

4.6 Limitation

During evaluation of the separated Gaussian method, small variations in RGB metrics were observed when modifying the semantic objective, suggesting that some parameters, specifically camera parameters, were still shared between the RGB and semantic pipelines. To evaluate this interaction, an additional experiment was conducted with fully detached the camera parameters between the two Gaussian sets. Contrary to expectations, this resulted in slightly worse performance for both RGB and semantic metrics. [Table 11] This indicates that sharing camera parameters provides a small stabilising effect during optimisation by maintaining consistent geometric alignment between the two representations, while the remaining interaction between Gaussian sets remains minimal.

5 Conclusion

This work provides additional insight into how semantic information can be effectively integrated into 3D Gaussian Splatting representations. Through a systematic evaluation of semantic-aware training strategies, the experiments highlight which components meaningfully influence semantic performance and which introduce unnecessary complexity.

For the joint Gaussian method, integrating semantics into an existing 3DGS pipeline proves straightforward. Because RGB reconstruction already provides strong structural guidance for the Gaussian primitives, complex semantic-specific supervision is unnecessary to obtain acceptable semantic predictions. However, sharing the same Gaussian representation for both RGB and semantic objectives leads to a noticeable degradation in RGB reconstruction quality, indicating a conflict between the two optimisation objectives.

In contrast, the separated Gaussian method removes this conflict by decoupling the RGB and semantic Gaussian sets. The experiments show that a complex semantic loss formulations provide limited benefit in this configuration. Instead, the most important factor for improving semantic performance is the structural organisation of the semantic Gaussians, with depth supervision providing the most consistent improvements.

Considering this trade-offs, the separated Gaussian method offers the most balanced solution. While slightly slower and producing relatively lower semantic metrics than the joint method, it maintains high RGB reconstruction quality while still achieving acceptable semantic performance. Combined with the reduced capacity required for semantic Gaussians, this organisation provides an efficient and stable way to incorporate semantics into 3DGS.

Future work could further improve the separated representation by reinforcing the structural guidance of semantic Gaussians, such as with the integration of stronger geometric constraints into semantic learning for Gaussian Splatting. Incorporating explicit geometric consistency, for example through surface regularisation, or multi-view geometric priors, could help stabilise semantic predictions and encourage semantically consistent structures across views. It could also explore whether semantic rendering requires the full complexity of current Gaussian splatting rasterisers. Since semantic output is inherently simpler than RGB reconstruction, a lighter 2D rasterisation or compositing strategy may be sufficient for semantic prediction. This could reduce computational cost and simplify the pipeline, although the trade-off with multi-view 3D consistency would need to be evaluated carefully.

References

[1] H. Caesar et al., *Nuscenes: A multimodal dataset for autonomous driving*, 2020. arXiv: 1903.11027 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1903.11027.

[2] X. Zhang, S. Bi, K. Sunkavalli, H. Su, and Z. Xu, *Nerfusion: Fusing radiance fields for large-scale scene reconstruction*, 2022. arXiv: 2203.11283 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2203.11283.

[3] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, *3d gaussian splatting for real-time radiance field rendering*, 2023. arXiv: 2308.04079 [cs.GR]. [Online]. Available: https://arxiv.org/abs/2308.04079.

[4] Y. Yan et al., *Street gaussians: Modeling dynamic urban scenes with gaussian splatting*, 2024. arXiv: 2401.01339 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2401.01339.

[5] R. Song et al., *Coda-4dgs: Dynamic gaussian splatting with context and deformation awareness for autonomous driving*, 2025. arXiv: 2503.06744 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2503.06744.

[6] Z. Chen et al., *Omnire: Omni urban scene reconstruction*, 2024. arXiv: 2408.16760 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2408.16760.

[7] G. Hess, C. Lindström, M. Fatemi, C. Petersson, and L. Svensson, *Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving*, 2025. arXiv: 2411.16816 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2411.16816.

[8] J. Guo, X. Ma, Y. Fan, H. Liu, and Q. Li, *Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting*, 2024. arXiv: 2403.15624 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2403.15624.

[9] X. Wang, C. Lan, H. Zhu, Z. Chen, and Y. Lu, *Gsemsplat: Generalizable semantic 3d gaussian splatting from uncalibrated image pairs*, 2024. arXiv: 2412.16932 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2412.16932.

[10] H. Zhou et al., *Hugs: Holistic urban 3d scene understanding via gaussian splatting*, 2024. arXiv: 2403.12722 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2403.12722.

[11] Y. Yan, C. Liang, W. Wang, and Y. Yang, *Open-ended 3d metric-semantic representation learning via semantic-embedded gaussian splatting*, 2024. [Online]. Available: https://openreview.net/forum?id=JGr4Qv9vbz.

[12] X. Zuo, P. Samangouei, Y. Zhou, Y. Di, and M. Li, *Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding*, 2024. arXiv: 2401.01970 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2401.01970.

[13] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, *Language embedded 3d gaussians for open-vocabulary scene understanding*, 2023. arXiv: 2311.18482 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2311.18482.

[14] Y. Li et al., *Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining*, 2025. arXiv: 2503.18052 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2503.18052.

[15] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, *Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes*, 2024. arXiv: 2312.07920 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2312.07920.

[16] H. Li et al., *Vdg: Vision-only dynamic gaussian for driving simulation*, 2024. arXiv: 2406.18198 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2406.18198.

[17] R. Wilson, *Gsavs: Gaussian splatting-based autonomous vehicle simulator*, 2024. arXiv: 2412.18816 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2412.18816.

[18] P.-C. Kung, X. Zhang, K. A. Skinner, and N. Jaipuria, *Lihi-gs: Lidar-supervised gaussian splatting for highway driving scene reconstruction*, 2024. arXiv: 2412.15447 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2412.15447.

[19] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, *Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction*, 2023. arXiv: 2309.13101 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2309.13101.

[20] Y. Chen, C. Gu, J. Jiang, X. Zhu, and L. Zhang, *Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering*, 2024. arXiv: 2311.18561 [cs.CV]. [Online].

Available: https://arxiv.org/abs/2311.18561.

[21] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, *In-place scene labelling and understanding with implicit scene representation*, 2021. arXiv: 2103.15875 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2103.15875.

[22] Z.-T. Chou, S.-Y. Huang, I.-J. Liu, and Y.-C. F. Wang, *Gsnerf: Generalizable semantic neural radiance fields with enhanced 3d scene understanding*, 2024. arXiv: 2403.03608 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2403.03608.

[23] D. Yang, Y. Gao, X. Wang, Y. Yue, Y. Yang, and M. Fu, *Opengs-slam: Open-set dense semantic slam with 3d gaussian splatting for object-level scene understanding*, 2025. arXiv: 2503.01646 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2503.01646.

[24] S. Zhou et al., *Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields*, 2024. arXiv: 2312.03203 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2312.03203.

[25] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997. [Online]. Available: https://api.semanticscholar.org/CorpusID:45998148.

[26] A. Kendall, Y. Gal, and R. Cipolla, *Multi-task learning using uncertainty to weigh losses for scene geometry and semantics*, 2018. arXiv: 1705.07115 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1705.07115.

[27] Z. Dai, T. Liu, and Y. Zhang, "Efficient decoupled feature 3d gaussian splatting via hierarchical compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2025, pp. 11 156–11 166.

[28] S. R. Bulò, L. Porzi, and P. Kontschieder, *Revising densification in gaussian splatting*, 2024. arXiv: 2404.06109 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2404.06109.

[29] Z. Zhang, W. Hu, Y. Lao, T. He, and H. Zhao, *Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting*, 2024. arXiv: 2403.15530 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2403.15530.

[30] S. Kim, K. Lee, and Y. Lee, "Color-cued efficient densification method for 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2024, pp. 775–783.

[31] Z. Zeng, Y. Wang, L. Ju, and T. Guan, *Frequency-aware density control via reparameterization for high-quality rendering of 3d gaussian splatting*, 2025. arXiv: 2503.07000 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2503.07000.

[32] X. Deng, C. Diao, M. Li, R. Yu, and D. Xu, *Efficient density control for 3d gaussian splatting*, 2025. arXiv: 2411.10133 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2411.10133.

[33] G. Grubert, F. Barthel, A. Hilsmann, and P. Eisert, "Improving adaptive density control for 3d gaussian splatting," in *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, SCITEPRESS - Science and Technology Publications, 2025, pp. 610–621. DOI: 10.5220/0013308500003912. [Online]. Available: http://dx.doi.org/10.5220/0013308500003912.

[34] S. Nam, X. Sun, G. Kang, Y. Lee, S. Oh, and E. Park, *Generative densification: Learning to densify gaussians for high-fidelity generalizable 3d reconstruction*, 2025. arXiv: 2412.06234 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2412.06234.

A Research Results

In this appendix can be found the tables compiling the results obtained during the research for each semantic-aware method evaluated. In the following tables, the number found at the end of each method's name represent the weight λ given to the semantic-aware method in the total loss formulation, unless stated otherwise in the table's caption.

Table 1: Results for the selection of the main loss between CE and Focal, for both joint and separated Gaussian method. In yellow the best overall separated Gaussian method configuration. In blue the best overall joint Gaussian method configuration.

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_CE_0,00032	0.7947	26.6403	0.2436	0.4497	0.7106	24.6336	0.2610	0.5612	1.4615	0.3318
Separated_Gaussian_CE_0,0016	0.7893	26.5059	0.2472	0.4610	0.7018	24.4385	0.2654	0.5625	1.5358	0.3236
Separated_Gaussian_CE_0,008	0.7724	26.0805	0.2607	0.4559	0.6807	23.8911	0.2799	0.5325	1.7662	0.3015
Separated_Gaussian_CE_0,04	0.7626	26.0429	0.2732	0.4357	0.6883	24.1413	0.2907	0.4904	1.9583	0.2775
Separated_Gaussian_CE_0,2	0.7559	25.6385	0.2774	0.4076	0.6971	24.1538	0.2924	0.4560	2.2289	0.2613
Separated_Gaussian_CE_1	0.7553	25.6500	0.2791	0.3680	0.7074	24.3649	0.2927	0.4063	2.5814	0.2217
Separated_Gaussian_Focal_0,00032	0.7953	26.6581	0.2428	0.4376	0.7119	24.6701	0.2603	0.5477	1.3932	0.3260
Separated_Gaussian_Focal_0,0016	0.7943	26.6312	0.2435	0.4564	0.7096	24.6044	0.2611	0.5695	1.4045	0.3315
Separated_Gaussian_Focal_0,008	0.7860	26.4201	0.2496	0.4553	0.6965	24.2772	0.2682	0.5497	1.5788	0.3197
Separated_Gaussian_Focal_0,04	0.7683	25.9682	0.2643	0.4453	0.6796	23.8166	0.2829	0.5162	1.8046	0.2948
Separated_Gaussian_Focal_0,2	0.7589	25.7200	0.2742	0.4317	0.6890	23.9700	0.2906	0.4890	2.0247	0.2774
Separated_Gaussian_Focal_1	0.7556	25.6120	0.2782	0.3996	0.7006	24.2164	0.2930	0.4483	2.2929	0.2541
Joint_Gaussian_CE_0,00032	0.7758	26.1126	0.2635	0.4957	0.6995	24.3812	0.2795	0.6115	1.2236	0.3314
Joint_Gaussian_CE_0,0016	0.7354	24.7359	0.3176	0.4988	0.6682	23.3742	0.3315	0.6123	1.4084	0.3542
Joint_Gaussian_CE_0,008	0.6237	21.4111	0.4978	0.4847	0.5835	20.6532	0.5082	0.5942	1.5539	0.3380
Joint_Gaussian_CE_0,04	0.5522	19.1981	0.6109	0.4634	0.5313	18.6626	0.6204	0.5600	1.4363	0.3175
Joint_Gaussian_CE_0,2	0.5204	18.0613	0.6627	0.4415	0.5079	17.6210	0.6713	0.5266	1.3544	0.3085
Joint_Gaussian_CE_1	0.5107	17.5814	0.6802	0.4247	0.4960	17.0955	0.6901	0.5029	1.5809	0.2891
Joint_Gaussian_Focal_0,00032	0.7892	26.3656	0.2473	0.4855	0.7057	24.4358	0.2645	0.5842	1.1297	0.3414
Joint_Gaussian_Focal_0,0016	0.7731	25.7827	0.2656	0.5006	0.6934	24.0671	0.2814	0.6100	1.2253	0.3499
Joint_Gaussian_Focal_0,008	0.7120	23.8895	0.3550	0.4991	0.6521	22.6700	0.3675	0.6159	1.4505	0.3562
Joint_Gaussian_Focal_0,04	0.6006	20.7552	0.5346	0.4792	0.5656	20.0711	0.5448	0.5852	1.5473	0.3336
Joint_Gaussian_Focal_0,2	0.5410	18.8458	0.6285	0.4587	0.5236	18.3434	0.6375	0.5543	1.4526	0.3153
Joint_Gaussian_Focal_1	0.5181	17.9440	0.6666	0.4367	0.5057	17.5043	0.6747	0.5202	1.5378	0.3034
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

Table 2: Results for the analysis for loss regularisation entropy vs l2, for both joint and separated Gaussian method. In yellow the best overall separated Gaussian method configuration. In blue the best overall joint Gaussian method configuration. In light yellow and blue the previous best overall used as base for this method analysis.

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_entropy_0.00000032	0.79440	26.63496	0.24337	0.45717	0.70928	24.59420	0.26108	0.56950	1.44183	0.33524
Separated_Gaussian_entropy_0.0000016	0.79424	26.63642	0.24377	0.45602	0.70922	24.60015	0.26171	0.57028	1.45061	0.33614
Separated_Gaussian_entropy_0.000008	0.79410	26.62739	0.24376	0.45392	0.70957	24.61277	0.26119	0.56605	1.44258	0.32970
Separated_Gaussian_entropy_0.00004	0.79424	26.63703	0.24373	0.45796	0.70976	24.62459	0.26117	0.57015	1.40617	0.33158
Separated_Gaussian_entropy_0.0002	0.79430	26.64208	0.24359	0.45533	0.70969	24.61255	0.26122	0.56875	1.44583	0.33176
Separated_Gaussian_entropy_0.001	0.79416	26.63895	0.24358	0.45596	0.70949	24.61448	0.26118	0.56654	1.43872	0.33073
Separated_Gaussian_Focal_0,0016	0.79435	26.63119	0.24346	0.45640	0.70957	24.60441	0.26113	0.56946	1.40447	0.33147
Separated_Gaussian_l2_0.00000032	0.79416	26.63545	0.24356	0.45340	0.70900	24.59888	0.26142	0.56512	1.40864	0.33380
Separated_Gaussian_l2_0.0000016	0.79436	26.63420	0.24334	0.45680	0.70939	24.60251	0.26103	0.56911	1.41239	0.33239
Separated_Gaussian_l2_0.000008	0.79443	26.63269	0.24317	0.45540	0.70924	24.58437	0.26104	0.56854	1.42322	0.33228
Separated_Gaussian_l2_0.00004	0.80733	26.84789	0.22613	0.45840	0.72229	24.80132	0.24392	0.56848	1.43858	0.33788
Separated_Gaussian_l2_0.0002	0.79442	26.64528	0.24349	0.45038	0.70942	24.61250	0.26134	0.56187	1.40133	0.32943
Separated_Gaussian_l2_0.001	0.79429	26.64579	0.24371	0.44248	0.70948	24.62168	0.26151	0.55374	1.38708	0.32972
Joint_Gaussian_entropy_0.00000032	0.77283	25.75397	0.26598	0.50054	0.69353	24.03948	0.28172	0.61056	1.24000	0.34988
Joint_Gaussian_entropy_0.0000016	0.77280	25.76396	0.26575	0.50131	0.69338	24.05974	0.28175	0.61178	1.25653	0.35027
Joint_Gaussian_entropy_0.000008	0.77302	25.77601	0.26550	0.50204	0.69332	24.06144	0.28143	0.61348	1.26564	0.35099
Joint_Gaussian_entropy_0.00004	0.77282	25.76152	0.26582	0.50028	0.69326	24.05536	0.28155	0.60740	1.26631	0.35071
Joint_Gaussian_entropy_0.0002	0.77295	25.75305	0.26566	0.50105	0.69347	24.04654	0.28136	0.60964	1.25583	0.35072
Joint_Gaussian_entropy_0.001	0.77297	25.76749	0.26567	0.50170	0.69355	24.05386	0.28149	0.61077	1.28031	0.35017
Joint_Gaussian_Focal_0,0016	0.77309	25.78268	0.26559	0.50060	0.69340	24.06708	0.28142	0.60998	1.22528	0.34991
Joint_Gaussian_l2_0.00000032	0.77295	25.77651	0.26582	0.50069	0.69334	24.07380	0.28178	0.61030	1.23714	0.35192
Joint_Gaussian_l2_0.0000016	0.77293	25.77440	0.26568	0.50048	0.69351	24.07181	0.28155	0.60912	1.23447	0.35106
Joint_Gaussian_l2_0.000008	0.77308	25.76255	0.26497	0.50073	0.69337	24.03963	0.28093	0.61090	1.23553	0.35101
Joint_Gaussian_l2_0.00004	0.77308	25.75511	0.26552	0.50099	0.69319	24.03883	0.28181	0.61116	1.28272	0.35182
Joint_Gaussian_l2_0.0002	0.77306	25.75813	0.26542	0.49870	0.69343	24.04466	0.28156	0.60787	1.23997	0.35095
Joint_Gaussian_l2_0.001	0.77312	25.75269	0.26546	0.49790	0.69379	24.05162	0.28106	0.60730	1.24111	0.35077
Vanilla	0.80559	27.46432	0.24149	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.85339	#DIV/0!
VanillaSemantics	0.79672	26.72037	0.24042	0.32105	0.71365	24.72948	0.25802	0.31974	1.13203	0.24971

Table 3: Results for the analysis for CLIP feature alignment, for both joint and separated Gaussian method. In yellow the best overall separated Gaussian method configuration. In blue the best overall joint Gaussian method configuration. In light yellow and blue the previous best overall used as base for this method analysis.(if no method with light color exist, this means this semantic-aware method was not able to improve the previous best results)

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_CLIP_0,000008	0.7942	26.6244	0.2435	0.4550	0.7094	24.5945	0.2611	0.5685	1.4276	0.3337
Separated_Gaussian_CLIP_0,00004	0.7943	26.6393	0.2433	0.4549	0.7094	24.6137	0.2610	0.5663	1.4318	0.3310
Separated_Gaussian_CLIP_0,0002	0.7943	26.6303	0.2435	0.4469	0.7093	24.5961	0.2610	0.5537	1.4641	0.3245
Separated_Gaussian_CLIP_0,001	0.7943	26.6490	0.2432	0.4290	0.7097	24.6196	0.2606	0.5191	1.4709	0.3115
Separated_Gaussian_l2_0.00004	0.8073	26.8479	0.2261	0.4584	0.7223	24.8013	0.2439	0.5685	1.4386	0.3379
Joint_Gaussian_CLIP_0,000008	0.7731	25.7611	0.2653	0.5005	0.6933	24.0396	0.2814	0.6098	1.3036	0.3506
Joint_Gaussian_CLIP_0,00004	0.7730	25.7766	0.2656	0.5013	0.6933	24.0714	0.2814	0.6126	1.3153	0.3494
Joint_Gaussian_CLIP_0,0002	0.7729	25.7656	0.2658	0.5009	0.6932	24.0472	0.2821	0.6099	1.3597	0.3489
Joint_Gaussian_CLIP_0,001	0.7669	25.8882	0.2707	0.4940	0.6892	24.2301	0.2860	0.6033	1.3776	0.3484
Joint_Gaussian_entropy_0.000008	0.7730	25.7760	0.2655	0.5020	0.6933	24.0614	0.2814	0.6135	1.2656	0.3510
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

Table 4: Results for the analysis for the depth loss, for both joint and separated Gaussian method. In yellow the best overall separated Gaussian method configuration. In blue the best overall joint Gaussian method configuration. In light yellow and blue the previous best overall used as base for this method analysis.(if no method with light color exist, this means this semantic-aware method was not able to improve the previous best results)

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_depth_0,00032	0.8099	27.1478	0.2266	0.4696	0.7286	25.1177	0.2435	0.5856	1.5284	0.3402
Separated_Gaussian_depth_0,0016	0.7837	26.4441	0.2571	0.4820	0.6968	24.4291	0.2754	0.6091	1.3876	0.3522
Separated_Gaussian_depth_0,008	0.7940	26.6196	0.2434	0.4805	0.7084	24.5910	0.2611	0.6069	1.5206	0.3439
Separated_Gaussian_depth_0,04	0.7925	26.5947	0.2447	0.4794	0.7089	24.6184	0.2622	0.6008	1.6740	0.3463
Separated_Gaussian_depth_0,2	0.7751	26.1968	0.2641	0.4792	0.7029	24.4914	0.2803	0.5840	2.4090	0.3542
Separated_Gaussian_l2_0.00004	0.8073	26.8479	0.2261	0.4584	0.7223	24.8013	0.2439	0.5685	1.4386	0.3379
Joint_Gaussian_Depth_off	0.7591	25.5117	0.2947	0.4937	0.6880	23.8591	0.3101	0.6025	1.1223	0.3502
Joint_Gaussian_entropy_0.000008	0.7730	25.7760	0.2655	0.5020	0.6933	24.0614	0.2814	0.6135	1.2656	0.3510
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

Table 5: Results for the analysis for the class based weight, for both joint and separated Gaussian method. For this 2 manual sets of weights (foreground and background) and 1 set of weights from the inverse frequency per classes, are applied to the per-pixel main semantic loss. In yellow the best overall separated Gaussian method configuration. In blue the best overall joint Gaussian method configuration. In light yellow and blue the previous best overall used as base for this method analysis.(if no method with light color exist, this means this semantic-aware method was not able to improve the previous best results)

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_background	0.7835	26.3644	0.2515	0.4717	0.6911	24.1475	0.2701	0.5918	1.8164	0.3385
Separated_Gaussian_depth_0,008	0.7940	26.6196	0.2434	0.4805	0.7084	24.5910	0.2611	0.6069	1.5206	0.3439
Separated_Gaussian_foreground	0.7892	26.5045	0.2471	0.4780	0.6993	24.3527	0.2658	0.5907	1.6921	0.3428
Separated_Gaussian_inverseW	0.7928	26.6039	0.2444	0.4490	0.7069	24.5720	0.2622	0.5677	1.5656	0.3124
Joint_Gaussian_background	0.6852	23.5378	0.3957	0.4889	0.6304	22.4354	0.4078	0.6096	1.2737	0.3554
Joint_Gaussian_entropy_0.000008	0.7730	25.7760	0.2655	0.5020	0.6933	24.0614	0.2814	0.6135	1.2656	0.3510
Joint_Gaussian_foreground	0.7357	24.4871	0.3184	0.4915	0.6665	23.1015	0.3325	0.5965	1.3850	0.3423
Joint_Gaussian_inverseW	0.7714	25.6809	0.2670	0.4638	0.6925	24.0077	0.2832	0.5629	1.2743	0.3174
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

Table 6: Results for the analysis for the warmup schedule, for the joint Gaussian method. For this each method has a starting and end step between that will activate and ramp-up the weight multiplied to the main semantic loss. In blue the best overall joint Gaussian method configuration. In light blue the previous best overall used as base for this method analysis.(if no method with light color exist, this means this semantic-aware method was not able to improve the previous best results)

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Joint_Gaussian_entropy_0.000008	0.7730	25.7760	0.2655	0.5020	0.6933	24.0614	0.2814	0.6135	1.2656	0.3510
Joint_Gaussian_Warmup_0-5k	0.7968	26.7307	0.2405	0.3711	0.7138	24.7208	0.2579	0.4297	1.1536	0.2857
Joint_Gaussian_Warmup_10-15k	0.7967	26.7230	0.2404	0.3579	0.7137	24.7164	0.2580	0.3990	1.1643	0.2781
Joint_Gaussian_Warmup_20-25k	0.7966	26.7196	0.2405	0.3441	0.7135	24.6994	0.2581	0.3647	1.1480	0.2665
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

Table 7: Results for the analysis for the semantic importance densification strategies, for both joint and separated Gaussian method. For this each method is testing separately or in combination semantic importance culling, splitting, duplicating densification strategies. These strategies are respectively annotated by the letters c, s and d in the methods names. In yellow the best overall separated Gaussian method configuration. In blue the best overall joint Gaussian method configuration. In light yellow and blue the previous best overall used as base for this method analysis.(if no method with light color exist, this means this semantic-aware method was not able to improve the previous best results)

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_densification_c	0.7884	26.5972	0.2477	0.4845	0.6990	24.4457	0.2659	0.6068	1.7931	0.3498
Separated_Gaussian_densification_csd	0.7895	26.6160	0.2464	0.4667	0.6983	24.4444	0.2650	0.5876	1.5232	0.3344
Separated_Gaussian_densification_d	0.7884	26.4798	0.2475	0.4811	0.6990	24.3903	0.2658	0.6056	1.5768	0.3541
Separated_Gaussian_densification_s	0.7882	26.4834	0.2478	0.4750	0.6987	24.3661	0.2659	0.5949	1.6784	0.3357
Separated_Gaussian_depth_0,008	0.7940	26.6196	0.2434	0.4805	0.7084	24.5910	0.2611	0.6069	1.5206	0.3439
Joint_Gaussian_densification_c	0.7732	25.7725	0.2652	0.5001	0.6942	24.0771	0.2807	0.6107	1.3514	0.3493
Joint_Gaussian_densification_csd	0.7732	25.7720	0.2654	0.5012	0.6938	24.0637	0.2812	0.6116	1.2991	0.3505
Joint_Gaussian_densification_d	0.7731	25.7847	0.2652	0.5013	0.6934	24.0714	0.2811	0.6118	1.2821	0.3495
Joint_Gaussian_densification_s	0.7730	25.7583	0.2651	0.5012	0.6934	24.0405	0.2812	0.6107	1.2805	0.3504
Joint_Gaussian_entropy_0.000008	0.7730	25.7760	0.2655	0.5020	0.6933	24.0614	0.2814	0.6135	1.2656	0.3510
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

Table 8: Results for the analysis for the semantic border loss, for both joint and separated Gaussian method. For this each method is associated to a weight that is multiplied to pixels on semantic borders in the main semantic per-pixel loss. In yellow the best overall separated Gaussian method configuration. In blue the best overall joint Gaussian method configuration. In light yellow and blue the previous best overall used as base for this method analysis.(if no method with light color exist, this means this semantic-aware method was not able to improve the previous best results)

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_Boundary_16	0.7820	26.3065	0.2529	0.4859	0.6940	24.2201	0.2710	0.6081	1.9422	0.3523
Separated_Gaussian_Boundary_4	0.7841	26.3663	0.2510	0.4854	0.6949	24.2546	0.2693	0.6104	2.0059	0.3533
Separated_Gaussian_Boundary_8	0.7829	26.3443	0.2522	0.4854	0.6943	24.2391	0.2703	0.6070	1.9395	0.3514
Separated_Gaussian_depth_0,008	0.7940	26.6196	0.2434	0.4805	0.7084	24.5910	0.2611	0.6069	1.5206	0.3439
Joint_Gaussian_Boundary_16	0.7644	25.4742	0.2725	0.5018	0.6879	23.8875	0.2880	0.6139	1.3108	0.3525
Joint_Gaussian_Boundary_4	0.7678	25.5796	0.2693	0.5027	0.6900	23.9657	0.2846	0.6159	1.3449	0.3529
Joint_Gaussian_Boundary_8	0.7659	25.5094	0.2709	0.5027	0.6890	23.9022	0.2864	0.6155	1.3254	0.3539
Joint_Gaussian_entropy_0.000008	0.7730	25.7760	0.2655	0.5020	0.6933	24.0614	0.2814	0.6135	1.2656	0.3510
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

Table 9: Results for the analysis for the semantic Gaussian capacity, for both joint and separated Gaussian method. For this each method is associated to the number of initialised semantic Gaussians.

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_SemCount_10000	0.7959	26.7920	0.2426	0.4503	0.7128	24.7880	0.2597	0.5587	1.2992	0.3629
Separated_Gaussian_SemCount_50000	0.7961	26.7760	0.2423	0.4645	0.7129	24.7696	0.2596	0.5755	1.3030	0.3622
Separated_Gaussian_SemCount_100000	0.7961	26.8004	0.2423	0.4856	0.7130	24.7775	0.2596	0.5994	1.3437	0.3595
Separated_Gaussian_SemCount_200000	0.7964	26.8061	0.2421	0.4830	0.7133	24.7976	0.2593	0.5931	1.3540	0.3659
Separated_Gaussian_SemCount_300000	0.7962	26.7945	0.2422	0.4841	0.7128	24.7812	0.2596	0.5929	1.3438	0.3760
Separated_Gaussian_SemCount_400000	0.7960	26.7610	0.2423	0.4860	0.7133	24.7699	0.2594	0.6007	1.3762	0.3673
Separated_Gaussian_SemCount_500000	0.7961	26.7732	0.2423	0.4821	0.7125	24.7252	0.2600	0.5897	1.3880	0.3720
Separated_Gaussian_SemCount_600000	0.7962	26.7916	0.2422	0.4780	0.7131	24.7741	0.2594	0.5837	1.4009	0.3618
Separated_Gaussian_SemCount_700000	0.7960	26.7994	0.2424	0.4825	0.7125	24.7626	0.2601	0.5898	1.4148	0.3666
Separated_Gaussian_SemCount_800000	0.7961	26.7967	0.2421	0.4800	0.7126	24.7655	0.2599	0.5861	1.4339	0.3612

Table 10: Final results for the best configuration of both Joint Gaussians and Separated Gaussian method.

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_depth_0,008_l2_0.00004_Focal_0,0016	0.7940	26.6196	0.2434	0.4805	0.7084	24.5910	0.2611	0.6069	1.5206	0.3439
Joint_Gaussian_entropy_0.000008_Focal_0,0016	0.7730	25.7760	0.2655	0.5020	0.6933	24.0614	0.2814	0.6135	1.2656	0.3510
Vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
VanillaSemantics	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

Table 11: Comparison for separated Gaussian method with and without attached camera parameters.

method	AVERAGE of SSIM	AVERAGE of PSNR	AVERAGE of LPIPS	AVERAGE of MIOU	AVERAGE of SSIM_test	AVERAGE of PSNR_test	AVERAGE of LPIPS_test	AVERAGE of MIOU_test	AVERAGE of total_run_time_h	AVERAGE of novel_view_miou
Separated_Gaussian_with_attached_cam	0.7961	26.7886	0.2420	0.4855	0.7130	24.7681	0.2592	0.5970	1.4249	0.3656
Separated_Gaussian_detached_cam	0.7960	26.7808	0.2426	0.4836	0.7133	24.7645	0.2598	0.5904	1.4089	0.3630
vanilla	0.8056	27.4643	0.2415	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	0.8534	#DIV/0!
vanillaGaussians	0.7967	26.7204	0.2404	0.3210	0.7137	24.7295	0.2580	0.3197	1.1320	0.2497

B Joint vs Separated Gaussian Method Visualisation

This appendix contains simplified flow diagrams of both the joint and separated Gaussian method to help visualise the organisation of each method.

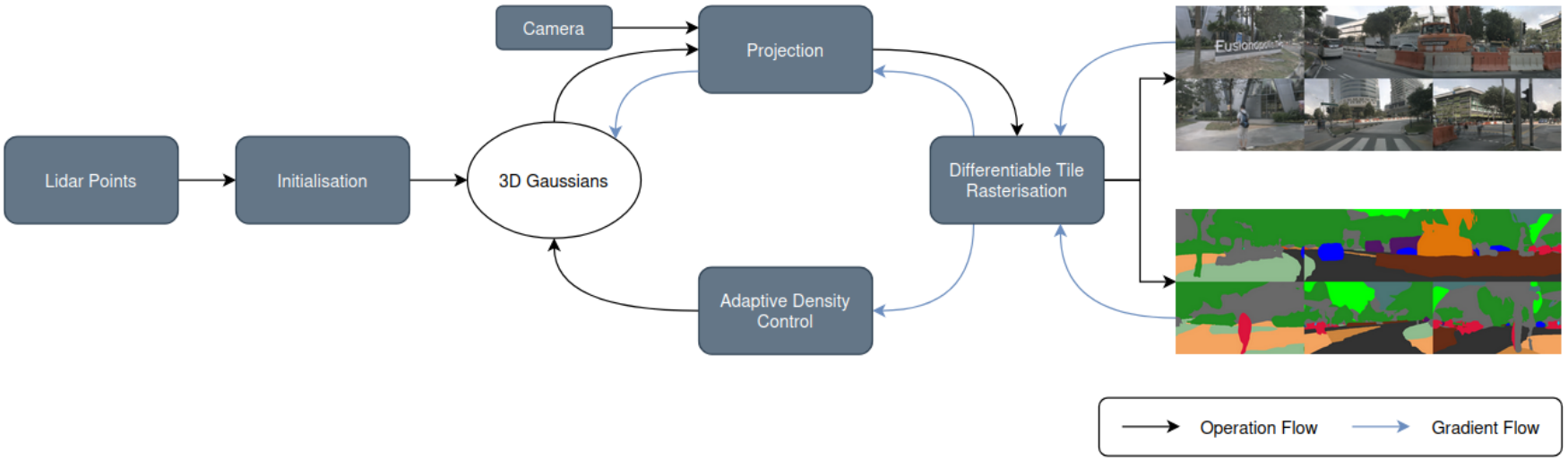


Figure 7: Flow diagram of the joint Gaussian method, this illustrate how both RGB renders and semantic maps are obtained from the same set of 3D Gaussians and how both of these reconstructions influence the same set of Gaussians thanks to the gradient flow during the back propagation. (Source: nuScenes dataset [1], used under CC BY-SA 4.0.)

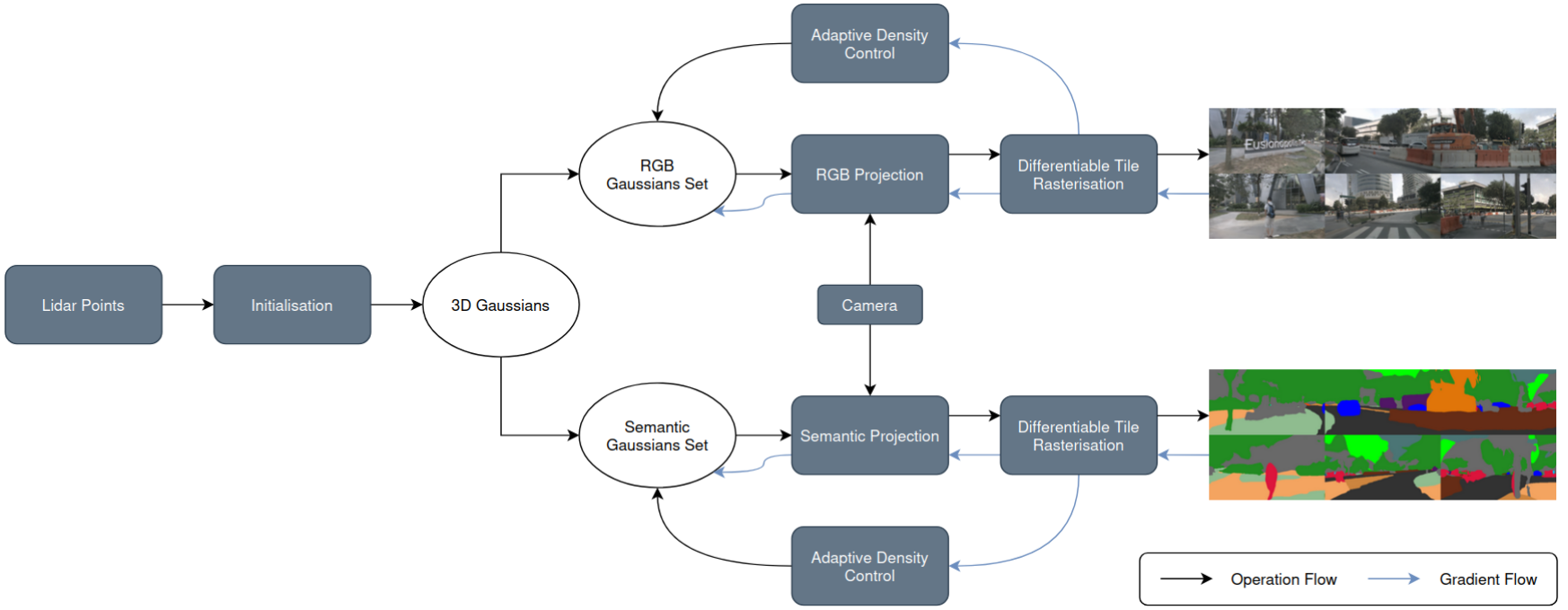


Figure 8: Flow diagram of the separated Gaussian method, this illustrate how both RGB renders and semantic maps are obtained from separate sets of 3D Gaussians and how each of these reconstructions influence their respective set of Gaussians thanks to the gradient flow during the back propagation. (Source: nuScenes dataset [1], used under CC BY-SA 4.0.)

C Training Losses and Formulations

This appendix provides the mathematical definitions of the losses and weighting strategies used in the semantic-aware training methods described in Section X.

C.1 Semantic Classification Losses

The cross-entropy loss is defined as

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{n=1}^{N}\log(\hat{p}_{n,y_n}) \tag{8}$$

where y_n is the ground-truth class label for pixel n and $\hat{p}_{n,y_n}$ is the predicted probability for that class.
To address class imbalance, focal loss is evaluated:

$$\mathcal{L}_{Focal} = -\frac{1}{N}\sum_{n=1}^{N}(1-\hat{p}_{n,y_n})^{\gamma}\log(\hat{p}_{n,y_n}) \tag{9}$$

where γ controls the strength of the focusing term.

C.2 Regularisation Losses

Entropy regularisation is defined as

$$\mathcal{L}_{entropy} = -\sum_{k=1}^{K}\hat{p}_k\log(\hat{p}_k) \tag{10}$$

which encourages smoother semantic probability distributions.
L2 regularisation on semantic logits is defined as

$$\mathcal{L}_{L2} = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}z_{n,k}^2. \tag{11}$$

C.3 CLIP Feature Alignment

CLIP-based supervision aligns predicted semantic features with CLIP image embeddings using cosine similarity:

$$\mathcal{L}_{CLIP} = 1 - \frac{f_{pred}\cdot f_{clip}}{\|f_{pred}\|\|f_{clip}\|}. \tag{12}$$

C.4 Depth Supervision

Depth supervision is applied using an L1 loss between the predicted semantic Gaussian depth map and ground-truth LiDAR depth:

$$\mathcal{L}_{depth} = \|\hat{D}_{sem} - D_{gt}\|_1. \tag{13}$$

C.5 Class-Weighted Cross Entropy

Class-weighted cross-entropy is defined as

$$\mathcal{L}_{WCE} = -\frac{1}{N}\sum_{n=1}^{N}w_{y_n}\log\hat{p}_{n,y_n}. \tag{14}$$

C.6 Boundary-Aware Weighting

Given a binary boundary mask b_n, the per-pixel weighting multiplier is

$$m_n = 1 + \lambda_{bd} b_n. \tag{15}$$

The weighted classification loss is computed as

$$\mathcal{L}_{main}^{w} = \frac{\sum_{n=1}^{N} m_n \ell_n}{\sum_{n=1}^{N} m_n}. \tag{16}$$

C.7 Semantic Loss Warm-Up

The semantic loss weight can be gradually introduced during training using

$$\lambda_{main}(t) = \lambda_{max} \cdot \min\left(1, \frac{t - t_0}{t_1 - t_0}\right). \tag{17}$$

C.8 Semantic Importance Score

For each Gaussian i, the predicted class and confidence are defined as

$$k_i = \arg\max_k \hat{p}_{i,k}, \qquad c_i = \max_k \hat{p}_{i,k}. \tag{18}$$

The semantic importance score is then computed as

$$S_i = w_{k_i}\, c_i^{\alpha}, \tag{19}$$

where w_{k_i} is the inverse-frequency class weight and α controls the influence of semantic confidence.