WHERE TO ALLOCATE GAUSSIANS: LEARNING SEMANTIC ALLOCATION FOR 3D OCCUPANCY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic 3D Gaussians provide a compact representation for 3D semantic occupancy prediction by rendering semantic primitives into a voxel volume under voxel-wise supervision. Recent methods have improved the modeling ability and efficiency of this representation through more flexible primitive shapes, geometry-guided initialization, and progressive densification. However, these advances mainly determine how primitives are represented, initialized, or added, and do not explicitly address how to select the most useful Gaussians when their total number must be limited to control memory and computation. This imbalance creates an allocation bottleneck: redundant Gaussians remain in simple regions, while difficult regions receive insufficient semantic support. We propose the Semantic Gaussian Allocation Transformer (SAGFormer), which uses Gaussian attributes and local geometric-semantic features to score candidates and select a budgeted final Gaussian set. Experiments on nuScenes-SurroundOcc and SSCBench-KITTI-360 show that SAGFormer improves occupancy prediction under the evaluated protocols and yields more semantically consistent and better-utilized Gaussian representations. Under similar final counts and raw coverage, it reduces semantic mixing, strengthens class-consistent voxel support, and produces fewer unused Gaussians. The results indicate that explicit capacity allocation is a useful complement to Gaussian refinement for semantic occupancy prediction.

1 INTRODUCTION

3D occupancy prediction estimates the geometry and semantic labels of the surrounding space, including both visible and occluded regions Wei et al. (2023); Tian et al. (2023). Dense voxel grids provide a direct scene representation, but their memory and computation increase rapidly with spatial range and resolution Li et al. (2023). Semantic Gaussian methods provide a sparse alternative. They represent a scene with semantic primitives and render these primitives into a voxel occupancy volume Huang et al. (2024b); Zhao et al. (2025).

Recent work improves sparse primitive representations along three main directions. SplatSSC uses depth cues to construct a compact initial Gaussian set Qian et al. (2026). QuadricFormer replaces ellipsoidal Gaussians with more flexible superquadrics and applies count-preserving pruning and splitting Zuo et al. (2025). PG-Occ progressively adds Gaussian queries to recover missing details for open-vocabulary occupancy Yan & Xu (2026). These methods focus on how primitives are represented, initialized, or progressively added. In contrast, we study which candidates should be kept under a Gaussian budget.

A compact Gaussian representation does not necessarily imply an effective allocation. Simple regions may contain several Gaussians with similar spatial support, whereas thin structures and class boundaries may receive insufficient semantically reliable support. Consequently, high geometric coverage can coexist with repeated Gaussians and semantic mixing. We refer to this mismatch as an *allocation bottleneck*. Rendering-oriented allocation is embedded in the photometric Gaussian generator rather than applied to an existing 3D Gaussian set Kim et al. (2026); Wan et al. (2026). Transferring it to occupancy therefore requires redesigning the allocation unit, semantic heads, and supervision. Semantic occupancy instead requires consistent geometry and class predictions at dis-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

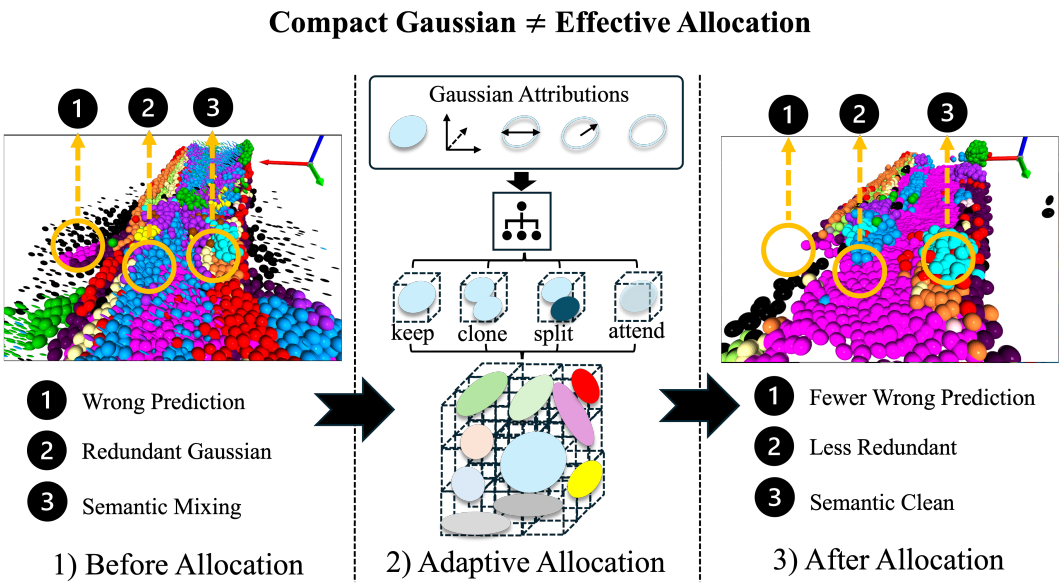


Figure 1: Schematic of semantic Gaussian allocation. Candidates compete under a nominal final budget; the panels illustrate occupancy prediction errors, redundant Gaussian support, and semantic mixing before and after allocation. Prediction accuracy and Gaussian support are evaluated separately in Section 4.

crete voxels. Candidate selection for this task should therefore account for semantic uncertainty, local density, geometric overlap, and redundancy.

To address this problem, we propose the Semantic Gaussian Allocation Transformer (SAGFormer). It encodes each Gaussian using its attributes and local geometric-semantic context, models relations among Gaussians, and generates keep, clone, split, and suppress candidates. These candidate operations are standard. The main contribution is an occupancy-oriented scoring and selection process that places all candidates in a shared pool. Global score-based selection materializes the final candidates under a nominal budget. We further analyze redundancy, semantic mixing, class-consistent voxel support, and Gaussian usage to characterize the resulting allocation.

The main contributions of this work are summarized as follows:

- We formulate semantic Gaussian allocation as a candidate-selection problem under a nominal final budget. Our analysis shows that high raw coverage can coexist with repeated Gaussians, mixed semantic support, and weak support for occupied voxels.
- We propose SAGFormer, which combines occupancy-related features, relations among Gaussians, standard candidate operations, and global score-based selection. We measure the final physical count after hard selection.
- We evaluate SAGFormer on two semantic occupancy benchmarks. Matched comparisons and allocation measures demonstrate improvements in both occupancy accuracy and the use of the Gaussian budget.

2 Related Work

Occupancy prediction and semantic Gaussians. Dense scene completion has been studied with monocular images, sparse voxel queries, multi-view features, and LiDAR or multi-modal networks Cao & de Charette (2021); Li et al. (2023); Wei et al. (2023); Tian et al. (2023); Roldao et al. (2020); Yang et al. (2024); Duan et al. (2025). GaussianFormer instead refines sparse semantic primitives before voxel decoding; GaussianFormer-2 improves their initialization and mixture representation

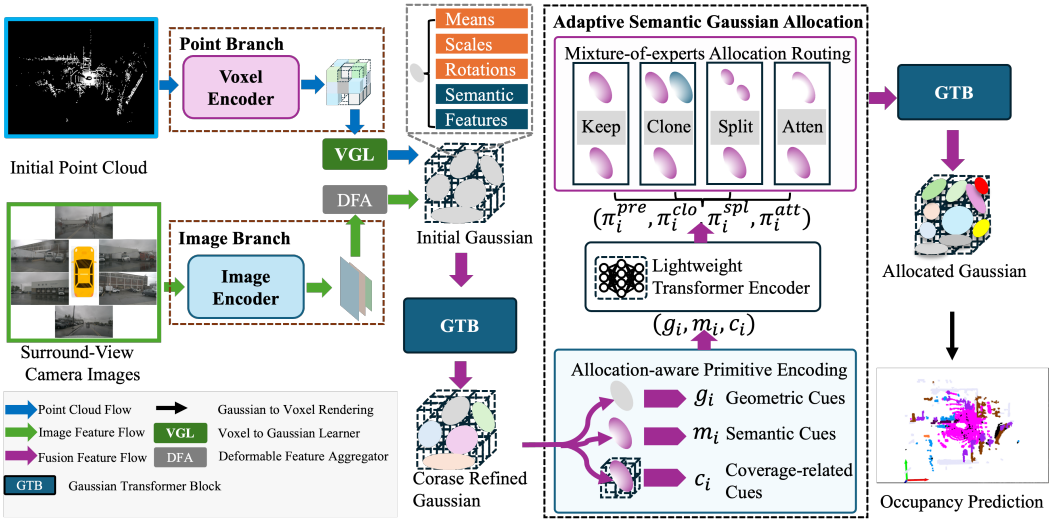


Figure 2: Overall framework of SAGFormer. Multi-modal features initialize semantic Gaussians. The allocation module uses intrinsic attributes and local geometric-semantic context to generate candidates and select the final set before voxel decoding.

Huang et al. (2024b;a). Subsequent work brings LiDAR geometry, image semantics, temporal information, and geometric completion into this representation Zhao et al. (2025); Doruk & Ates (2026); Lv et al. (2026); Song et al. (2025); Yan et al. (2025; 2026); Zhou et al. (2026); Shi et al. (2026); Chen et al. (2026).

Changing the primitive set. SplatSSC uses depth to initialize a compact set Qian et al. (2026). QuadricFormer prunes and splits superquadrics while preserving their count Zuo et al. (2025); PG-Occ adds queries progressively where depth evidence is missing Yan & Xu (2026). These studies make adaptive primitive placement a close point of comparison. Our question is narrower: given initialized semantic Gaussians, which generated candidates should occupy a per-scene final budget? Section 4 compares in-system rule adaptations at the same final count; they are not complete reproductions of the cited systems.

Allocation in Gaussian rendering. Generative Densification upsamples coarse Gaussian features, F4Splat allocates a global budget using spatial complexity and view overlap, and SplatWeaver predicts local counts Nam et al. (2024); Kim et al. (2026); Wan et al. (2026). Other work uses image gradients, detections, or reconstruction features to adjust placement Zhang et al. (2024); Ye et al. (2024); Jiang et al. (2025); Moreau et al. (2025). Semantic cues also guide pruning, boundary splitting, and redundancy removal in rendering and segmentation He et al. (2025); Zhang et al. (2025); Sheng et al. (2025). SAGFormer uses predicted Gaussian semantics and local overlap to rank candidates for voxel occupancy, then selects the final set globally. This differs from motion-specific mixture-of-experts routing Jin et al. (2026); Appendix Table 17 gives a compact comparison.

3 Method

3.1 Problem and Overview

Given multi-modal inputs, the model starts with initialized Gaussian set $\mathcal{G}^0 = \{\mathcal{G}_i^0\}_{i=1}^N$. Each Gaussian is $\mathcal{G}_i^0 = \{\boldsymbol{\mu}_i, \boldsymbol{s}_i, \boldsymbol{r}_i, \alpha_i, \boldsymbol{f}_i, \boldsymbol{p}_i\}$, where the entries denote its center, scale vector, rotation, opacity, feature, and semantic logits, respectively. To address the allocation bottleneck, SAGFormer expands this initial set into a shared candidate pool and ranks the candidates under a normalized budget B, with the goal of reducing redundant allocation and strengthening semantically reliable voxel support. The selected set $\hat{\mathcal{G}}$ is then used for voxel decoding.

The budget B sets a nominal growth target relative to the initial size N. Training uses a differentiable soft expected count, whereas inference ranks candidates and materializes a hard subset. We define the ranking score and nominal budget below. "Final #G" refers to the measured number of physical Gaussians after selection, not the soft count or the nominal target.

Figure 2 shows the full model. Image and point-cloud features first initialize semantic Gaussians. SAGFormer then builds allocation features and models relations among Gaussians. Finally, it generates keep, clone, split, and suppress candidates. These operations retain reliable support, add nearby support, divide a large or mixed region, or reduce a Gaussian's effect. Global selection decides which candidates enter the decoder.

3.2 Adaptive Semantic Gaussian Allocation

Given $\mathcal{G}^0$, SAGFormer has three steps: feature encoding, candidate generation and scoring, and global budgeted selection.

Reading the local context. A Gaussian's own attributes do not tell us whether its support is duplicated nearby or whether neighboring predictions disagree. For each input Gaussian, SAGFormer therefore reads its center, scale, rotation, opacity, and predicted class distribution, then summarizes up to eight neighbors within $2\,\mathrm{m}$. Four local cues capture the number of neighbors, their mean distance, scale-normalized overlap, and agreement between predicted class distributions. Count and distance describe crowding; overlap and semantic agreement help distinguish repeated support from a region that may need more candidates. The overlap cue uses axis-aligned scales and ignores rotation. These stop-gradient cues use predicted Gaussian states only, never ground-truth voxel labels. Their computation costs $O(NK_{\mathrm{nbr}}C)$ for C classes, excluding attention and decoding.

Let $\boldsymbol{g}_i$ contain normalized geometry, let $\boldsymbol{q}_i = \mathrm{Softmax}(\boldsymbol{p}_i)$, and let $\mathcal{N}_i$ contain at most eight neighbors within $R = 2\,\mathrm{m}$. With $\langle\cdot\rangle_i$ denoting their mean, the semantic and local vectors are

$$\begin{aligned} \boldsymbol{m}_i &= [\boldsymbol{q}_i,\ \max_c q_{i,c},\ -\boldsymbol{q}_i \odot \log(\boldsymbol{q}_i + \epsilon)], \\ \boldsymbol{c}_i &= \left[\frac{|\mathcal{N}_i|}{8},\ \left\langle \frac{\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_i\|_2}{R} \right\rangle_i, \langle \eta_{ij} \rangle_i,\ \langle \boldsymbol{q}_i^\top \boldsymbol{q}_j \rangle_i \right], \\ \eta_{ij} &= \exp\left(-\frac{1}{2} \sum_{d=1}^{3} \frac{(\mu_j^{(d)} - \mu_i^{(d)})^2}{(s_i^{(d)})^2 + (s_j^{(d)})^2 + \epsilon} \right). \end{aligned} \tag{1}$$

The overlap proxy η_{ij} ignores rotation. An MLP combines $\boldsymbol{g}_i$, $\boldsymbol{m}_i$, and $\boldsymbol{c}_i$ into $\boldsymbol{e}_i$. The original Gaussian feature $\boldsymbol{f}_i$ is then passed with $\boldsymbol{e}_i$ through two global self-attention Transformer blocks to obtain $\boldsymbol{z}_i$:

$$\boldsymbol{e}_i = \mathrm{MLP}_{enc}([\boldsymbol{g}_i, \boldsymbol{m}_i, \boldsymbol{c}_i]), \qquad \{\boldsymbol{z}_i\}_{i=1}^N = \mathrm{TransEnc}(\{[\boldsymbol{e}_i, \boldsymbol{f}_i]\}_{i=1}^N). \tag{2}$$

The neighborhood statistics use predicted states with stopped gradients and never use ground-truth labels.

Candidate generation and scoring. For each Gaussian $\mathcal{G}_i^0$, a small router predicts weights for the keep, clone, split, and suppress generators:

$$\boldsymbol{\pi}_i = \mathrm{Softmax}(\mathrm{MLP}_{op}(\boldsymbol{z}_i)) = [\pi_i^{pre}, \pi_i^{clo}, \pi_i^{spl}, \pi_i^{att}]. \tag{3}$$

The superscripts pre, clo, spl, and att denote keep, clone, split, and suppress, respectively, matching Fig. 2. Let $\mathcal{P}$ collect all generated candidates, with e indexing the generator and k a candidate from that generator. The ranking score and nominal count target are

$$\begin{aligned} \mathrm{score}(\mathcal{G}_{i,e,k}) &= \pi_i^e \alpha_{i,e,k}, \\ K_B^{\mathrm{nom}} &= \lfloor (1+B)N \rfloor. \end{aligned} \tag{4}$$

During training, all candidates contribute to occupancy decoding through continuous gate weights, enabling end-to-end optimization. Inference ranks candidates by the same gate-weighted opacity score and materializes a hard subset. Candidates compete individually, so selection may retain one or both outputs of a clone or split generator. The measured physical count may differ from K_B^{nom}; Table 6 reports the realized counts.

What each branch offers. The keep branch refines one Gaussian; clone offers the original plus one nearby child; split offers two independently updated children; suppress retains one Gaussian with lower opacity. For each generated child, a head predicts center offset $\boldsymbol{o}_i$, unconstrained scale $\tilde{\boldsymbol{\eta}}_i$, and opacity, feature, and logit residuals. With $\boldsymbol{\beta}_i = \mathrm{softplus}(\tilde{\boldsymbol{\eta}}_i) + \epsilon$, the update is

$$\mathcal{G}'_i = \{\boldsymbol{\mu}_i + \boldsymbol{o}_i,\ \boldsymbol{\beta}_i \odot \boldsymbol{s}_i,\ \boldsymbol{r}_i,\ \sigma(\mathrm{logit}(\alpha_i) + \Delta\alpha_i),\ \boldsymbol{f}_i + \Delta\boldsymbol{f}_i,\ \boldsymbol{p}_i + \Delta\boldsymbol{p}_i\}. \tag{5}$$

Clone offers the original and one child; split offers two independently updated children. Suppress keeps the other attributes and multiplies opacity by $\gamma_i = \sigma(\mathrm{MLP}_{att}(\boldsymbol{z}_i))$. The branches generate candidates rather than final decisions: both children, one child, or neither may survive global selection. A selected suppress candidate still occupies one physical slot.

Keep and suppress each produce one candidate, whereas clone and split each produce two. The normalized soft count first expands over all four branches and then simplifies because the routing weights sum to one:

$$\begin{aligned}\bar{\mathcal{C}}_\pi &= \frac{1}{N}\sum_{i=1}^{N}\left(\pi_i^{pre} + 2\pi_i^{clo} + 2\pi_i^{spl} + \pi_i^{att}\right) \\ &= 1 + \frac{1}{N}\sum_{i=1}^{N}\left(\pi_i^{clo} + \pi_i^{spl}\right) \in [1, 2].\end{aligned} \tag{6}$$

The base term accounts for one candidate per Gaussian, while clone and split contribute the expected extras. This is a router-weighted *expected* count: all six candidates per input Gaussian participate in training, so the soft penalty does not bound training memory. We penalize expected counts above $1 + B$. At inference, K_B^{nom} is the nominal physical-count target for each input set, while the final materialized count is measured directly.

3.3 Training and Final Selection

Training decodes the gate-weighted candidate pool so that the router and generators receive occupancy gradients before a discrete selection is made. The main occupancy loss combines voxel cross-entropy with Lovasz-Softmax. Two auxiliary terms encourage confident candidate semantics and penalize predicting empty at occupied ground-truth voxels. A fourth term penalizes only soft expected counts above the chosen budget:

$$\mathcal{L} = \mathcal{L}_{\mathrm{occ}} + \lambda_{\mathrm{ent}}\mathcal{L}_{\mathrm{ent}} + \lambda_{\mathrm{rec}}\mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{bud}}\max(0, \bar{\mathcal{C}}_\pi - (1 + B)). \tag{7}$$

Here $\mathcal{L}_{\mathrm{occ}} = \mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{lov}}\mathcal{L}_{\mathrm{lov}}$. For C classes and $\hat{N}$ weighted-pool candidates, let $\rho_{i,c}$ be candidate i's predicted class probability. Let $\mathcal{V}_{\mathrm{occ}}$ contain occupied ground-truth voxels and $\hat{p}_v^{\mathrm{emp}}$ be the predicted empty probability. The auxiliary terms are

$$\begin{aligned}\mathcal{L}_{\mathrm{ent}} &= -\frac{1}{\hat{N}\log C}\sum_{i=1}^{\hat{N}}\sum_{c=1}^{C}\rho_{i,c}\log(\rho_{i,c} + \epsilon), \\ \mathcal{L}_{\mathrm{rec}} &= -\frac{1}{|\mathcal{V}_{\mathrm{occ}}|}\sum_{v\in\mathcal{V}_{\mathrm{occ}}}\log(1 - \hat{p}_v^{\mathrm{emp}} + \epsilon).\end{aligned} \tag{8}$$

The losses supervise predicted classes and occupancy, whereas the local allocation cues use no ground-truth labels. Neither auxiliary term directly optimizes the post-hoc GT-support diagnostics in Section 4. At inference, the weighted pool is replaced by a hard selected set ranked with the score in Eq. (4). Appendix A gives the loss weights and optimization settings.

4 Experiments

4.1 Experimental Setup

Datasets. We select nuScenes-SurroundOcc Caesar et al. (2020); Wei et al. (2023) and SSCBench-KITTI-360 Liao et al. (2022); Li et al. (2024) as complementary benchmarks for testing SAG-Former across different view configurations and scene distributions. We use the official training/validation splits, spatial ranges, voxel resolutions, and semantic categories from prior occupancy

Table 1: nuScenes comparison at 16.6K final Gaussians. Opacity Top-K changes only scoring; "style" rows adapt cited rules. – denotes an unmeasured diagnostic.

Allocation policy	IoU ↑	mIoU ↑	$P_{\text{pen.}}$ ↑
No allocation (16.6K)	40.4±0.12	27.33±0.06	0.6484
Opacity Top-K	40.1±0.04	27.33±0.05	–
Random	40.5±0.13	27.53±0.05	0.6492
QuadricFormer-style Zuo et al. (2025)	40.5±0.11	27.56±0.05	0.6497
F4Splat-style Kim et al. (2026)	40.9±0.12	27.89±0.06	0.6503
PG-Occ-style Yan & Xu (2026)	41.0±0.12	28.03±0.05	0.6553
MLP controller	41.0±0.11	28.01±0.04	0.6544
SAGFormer	41.7±0.11	**28.47±0.06**	**0.6593**

work. nuScenes uses six surround-view images and LiDAR points in the LiDAR coordinate system, whereas KITTI-360 follows the SSCBench protocol with the left-view image and the corresponding LiDAR sweep.

Evaluation Metrics. We report occupancy IoU, semantic mIoU, and per-class IoU. To inspect the final representation, we assign each occupied ground-truth voxel to Gaussians within Mahalanobis radius $\tau_d = 3$. *Raw coverage* is the fraction reached by at least one Gaussian; *unused* is the fraction of Gaussians reaching none. For each used Gaussian, we find the majority ground-truth class among the voxels it reaches. *Mix-G* is the fraction whose majority share is below $\rho = 0.6$; *Sem-Sup* is the fraction of occupied voxels reached by a Gaussian whose majority share exceeds that threshold and whose majority class matches the voxel label. P_{valid} is the average majority share among used Gaussians, and $P_{\text{pen.}}$ penalizes unused ones. These measures describe geometric support relative to ground truth, *not* the accuracy of the Gaussian's predicted class. They are computed after inference and never enter allocation. Appendix A defines them formally; Appendix C tests the threshold.

Implementation Details. For our controlled variants and reproduced baselines, we match the data splits, voxelization, augmentation, and evaluation; controlled allocation-policy variants also share the initializer architecture, encoder, decoder, and training objectives. SAGFormer is trained for 20 epochs on nuScenes and 30 on KITTI-360. Mean and standard deviation use seeds 42, 37, and 91 where shown; the camera-only plug-in is a single run. Appendix A gives encoder, optimization, and hardware details.

SAGFormer Configuration. SAGFormer uses two global-attention Transformer blocks and the default budget $B = 0.6$. At inference, it collects all candidates, ranks them by gate-weighted opacity, and materializes a hard set under the nominal target $K_B^{\text{nom}} = \lfloor (1 + B)N \rfloor$. All occupancy and efficiency results use the measured final set. Appendix A gives the layer sizes, loss weights, and optimization settings.

Comparison Protocol. The primary test is raw single-frame inference: each validation sample supplies one scene frame as input. The SAGFormer initializer yields approximately 10.38K physical Gaussians per validation frame on average; at $B = 0.6$, allocation raises the average final count to 16.6K. "W/o Semantic Cues" keeps the same initializer, router, candidate generators, training, and nominal budget, but removes semantic logits, confidence, and entropy from the allocation input. The no-allocation control in the policy and diagnostic comparisons starts and ends with 16.6K physical Gaussians and reaches 27.33% mIoU (Tables 1 and 4). A separate 10.38K no-allocation version reaches 27.00% mIoU and appears in the efficiency comparison (Table 7); it is not a matched-mean-count accuracy control. Published multi-sweep and completion-enhanced results in Tables 2–3 provide context across different protocols, not paired comparisons. The 1^* setting follows the documented Gau-Occ training schedule Lv et al. (2026); Appendix A details each comparison.

4.2 What Changes at a Fixed Count?

At 16.6K final Gaussians, SAGFormer reaches 28.47±0.06% mIoU on nuScenes, against 27.33±0.05% for opacity Top-K, 28.03±0.05% for PG-Occ-style, and 28.01±0.04% for an MLP controller (Table 1). The 1.14-point gain over opacity holds the candidate pool and training fixed; the no-allocation row matches only final capacity. The margin over the strongest adapted policy is 0.44 point; its rule adaptation is not a full-system reproduction.

Table 2: nuScenes-SurroundOcc benchmark context, grouped by input protocol. C/L denote camera/LiDAR, and 1/M denote single-/multi-sweep LiDAR. * uses one raw LiDAR sweep at inference with temporally aggregated training supervision; † marks reported results, ‡ our reproduction, and – unavailable results. Available class-wise results are provided in Appendix B.

Method	Mod.	Frames	IoU ↑	mIoU ↑
GaussianFormer	C	1	29.83	19.10
GaussianFormer-2	C	1	31.74	20.82
GaussianFormer‡ (16.6K)	C	1	29.60	19.10
+ SAGFormer allocation (16.6K)	C	1	33.04	23.67
GaussianFormer3D†	C+L	M (10)	43.3	27.1
GaussianFormer3D‡	C+L	1	36.7	22.4
GaussianOcc3D†	C+L	M	–	28.9
DAOcc†	C+L	M (10)	45.0	30.5
DAOcc‡	C+L	1	40.1	25.6
SDG-Fusion†	C+L	M	45.2	31.7
SDG-Fusion‡	C+L	1	41.1	26.4
Gau-Occ†	C+L	1*	44.3	32.7
SAGFormer w/o Semantic Cues	C+L	1	39.47 ± 0.45	27.68 ± 0.12
SAGFormer	C+L	1	41.74 ± 0.11	28.47 ± 0.06
SAGFormer	C+L	1*	51.79 ± 0.12	33.16 ± 0.04

Removing the semantic inputs from the otherwise matched controller lowers mIoU from $28.47\pm0.06\%$ to $27.68\pm0.12\%$ on nuScenes and from $24.20\pm0.03\%$ to $23.76\pm0.03\%$ on KITTI-360 (Tables 2–3). The corresponding IoU gains are 2.27 and 0.96 points. These are aggregate improvements, not gains in every class: the semantic-cue variant declines on 5 of 16 nuScenes classes and 8 of 18 KITTI-360 classes (Appendix B). Small traffic and foreground object classes remain uneven.

On KITTI-360, opacity Top-K reaches $21.65\pm0.06\%$ mIoU at the same 18.6K final count, versus $24.20\pm0.03\%$ for SAGFormer. A no-allocation reference reaches 23.34% at 18.4K, without reported seed variability (Appendix Table 16).

4.3 Benchmarks and Transfer

Under raw single-frame camera–LiDAR input, SAGFormer reaches $41.74\pm0.11\%$ IoU / $28.47\pm0.06\%$ mIoU on nuScenes and $57.40\pm0.15\%$ / $24.20\pm0.03\%$ on KITTI-360. Tables 2–3 place these beside reproduced single-frame baselines and reported results with different temporal or training settings. In a separate, single-run camera-only GaussianFormer plug-in test at 16.6K Gaussians, adding SAGFormer raises mIoU from 19.10% to 23.67%; this suggests transfer to another initializer, pending multi-seed confirmation. With temporally aggregated training supervision (1^*), SAGFormer obtains $33.16\pm0.04\%$ on nuScenes and $26.80\pm0.02\%$ on KITTI-360. Those results are supplementary to the raw single-frame test. Appendix B gives available class-wise scores.

4.4 Where Does the Budget Go?

Table 4 separates reaching a voxel from using the primitive budget well. GaussianFormer reaches 99.88% of occupied voxels, yet 52.59% of its Gaussians reach no occupied voxel under the common assignment rule. Relative to a generator-matched control with nearly the same output count (16.5K versus 16.6K) and raw coverage (0.9104 versus 0.9103), SAGFormer lowers the unused fraction from 0.0854 to 0.0785 and the mixed-support fraction from 0.2963 to 0.2752. Class-consistent GT

Table 3: SSCBench-KITTI-360 benchmark context, with single-frame and temporally enhanced settings shown separately. C/L denote camera/LiDAR. * uses one raw LiDAR sweep at inference with temporally aggregated training supervision; † marks reported results, and – unavailable results. Available class-wise results are provided in Appendix B.

Method	Mod.	Frames	IoU ↑	mIoU ↑
MonoScene	C	1	37.87	12.31
VoxFormer	C	1	38.76	11.91
GaussianFormer	C	1	35.38	12.92
GaussianFormer-2	C	1	38.37	13.90
LMSCNet	L	1	47.53	13.65
GaussianFormer3D	C+L	10	54.5	21.3
Gau-Occ†	C+L	1*	58.9	25.8
SAGFormer with opacity Top-K	C+L	1	55.46 ± 0.08	21.65 ± 0.06
SAGFormer w/o Semantic Cues	C+L	1	56.44 ± 0.08	23.76 ± 0.03
SAGFormer	C+L	1	57.40 ± 0.15	24.20 ± 0.03
SAGFormer	C+L	1*	60.10 ± 0.09	26.80 ± 0.02

Table 4: Allocation and GT-support measures on nuScenes-SurroundOcc. Support purity uses majority *ground-truth* labels of covered voxels, not Gaussian class predictions. Final #G is the average physical output count measured after hard selection. FG, BG, and Macro are unweighted class-purity averages. ‡The generator-matched control uses the same keep, clone, split, and suppress candidate families as SAGFormer, but receives no geometric, semantic, or coverage cues and applies no explicit budget constraint.

Method	Allocation diagnostics						GT-support purity		
	mIoU ↑	Final #G	Raw Cov. ↑	Unused ↓	Mix-G ↓	Sem-Sup ↑	FG ↑	BG ↑	Macro ↑
GaussianFormer	19.10	25.6K	**0.9988**	0.5259	0.4133	0.5012	0.167	0.461	0.295
GaussianFormer3D†	27.10	25.6K	0.8967	0.1535	0.3169	0.6311	0.227	0.477	0.336
SAGFormer w/o Allocation	27.33	16.6K	0.9108	0.0867	0.3031	0.6411	0.207	0.484	0.328
Generator-matched control‡	27.43	16.5K	0.9104	0.0854	0.2963	0.6412	0.199	**0.488**	0.308
SAGFormer	**28.47**	16.6K	0.9103	**0.0785**	**0.2752**	**0.6627**	**0.234**	**0.488**	**0.345**

support rises from 0.6412 to 0.6627. That control has no explicit budget constraint, so the difference reflects the full allocation design, not semantic cues alone.

The category-balanced GT-support measure rises from 0.328 to 0.345, driven more by foreground support (0.199 to 0.234) than background (0.488 in both rows). These measures use the *majority ground-truth label of covered voxels*; they do not test whether a Gaussian predicts that label. Voxel mIoU and per-class IoU therefore remain the direct evidence for prediction quality. Appendix C provides distance bins, class-wise support, and threshold sensitivity.

4.5 Training Objectives and Candidate Availability

Table 5 tests the losses: occupancy supervision alone reaches 28.12% mIoU, and adding both auxiliary terms reaches 28.47%. Appendix Table 18 reports a candidate-family sweep with a common 10.38K initial count. Most rows have different final counts and are descriptive; the two 16.6K rows show a 0.42-point mIoU gain when suppress candidates are also available.

Table 6: nuScenes budget sensitivity. Final counts are post-allocation frame means; selection drop is the soft-to-hard mIoU loss. Full results appear in Appendix Table 20.

B	Final #G	mIoU ↑	Selection drop ↓
0.2	12.46K	27.69	0.321
0.4	14.53K	28.24	0.132
0.6	16.60K	**28.47**	0.031
0.8	18.68K	28.35	0.023
1.0	20.05K	28.31	0.022

Table 7: nuScenes accuracy and efficiency on shared hardware. The 16.6K mean-count-matched pair uses batch size 1; FPS is rounded. 1^* uses temporally aggregated training.

Method	Frames	Final #G	Memory (MB)	FPS	mIoU ↑
GaussianFormer	1	25.6K	4733	4	19.10
GaussianFormer3D	1	25.6K	6043	2	22.40
SAGFormer w/o Allocation	1	10.38K	2189	4	27.00
SAGFormer w/o Allocation	1	16.6K	2359	4	27.33
SAGFormer	1	16.6K	2456	4	28.47
SAGFormer-M	1^*	21.1K	3542	2	33.16

Table 5: nuScenes objective ablation. Entropy regularizes predicted semantics; recovery penalizes false empty voxels.

Training objective	IoU ↑	mIoU ↑
Occupancy only	41.1 ± 0.12	28.12 ± 0.05
+ entropy	41.3 ± 0.11	28.23 ± 0.06
+ recovery	41.4 ± 0.12	28.21 ± 0.05
Full objective	$\mathbf{41.7\pm0.11}$	$\mathbf{28.47\pm0.06}$

4.6 Budget, Materialization, and Cost

Table 6 reports measured counts, while K_B^{nom} is a target. At $B = 1.0$, 10.33K initial Gaussians yield 20.05K final, below the nominal 20.66K; the cause is unverified. $B = 0.6$ scores best, 0.16 mIoU point above $B = 1.0$. At $B = 0.6$, the soft count averages $1.63N$, and hard selection lowers mIoU by 0.031 point on nuScenes and 0.026 on KITTI-360 (Appendix D).

At the same 16.6K final count on RTX 4090 with batch size 1, SAGFormer uses 2456 MB versus 2359 MB without allocation, a 97 MB increase with 1.14 points higher mIoU (Table 7). Both round to 4 FPS, which cannot establish equal latency. The 10.38K reference is a separate capacity setting. Qualitative examples appear in Appendix E.

5 Limitations and Responsible Use

Gains are uneven. On nuScenes, semantic cues raise mean background IoU from 29.36% to 31.10%, but foreground-object IoU from 26.36% to 26.42% (Appendix Table 8); these class groups do not identify motion. Five nuScenes and eight KITTI-360 classes decline. The overlap cue ignores rotation, GT-support does not measure predicted-class correctness, and clone/split effects lack matched-count tests. Hard selection may retain low-opacity suppress candidates; its realized-count rule needs verification. Sparse moving objects, adverse weather, sensor degradation, and distribution shifts remain untested.

6 Conclusion

SAGFormer selects keep, clone, split, and suppress candidates under a Gaussian budget. It improves mIoU at matched mean final counts and reduces mixed and unused Gaussian support on nuScenes. Gains remain uneven for small foreground classes.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript organization, language editing, literature search and organization, and feedback on the presentation and interpretation of existing experimental results. All AI-assisted material was reviewed and verified by the authors. The authors also verified the manuscript's technical claims, equations, citations, experimental protocols, and reported results and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The method, experimental setup, matched comparisons, available implementation settings, and diagnostic definitions are described in Sections 3–4 and Appendix A. Additional class-wise results, ablations, and qualitative comparisons are provided in Appendices B–E. An anonymized code repository will accompany the final submission.

REFERENCES

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621–11631, 2020.

Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion, 2021. URL https://arxiv.org/abs/2112.00726.

Cheng Chen, Hao Huang, and Saurabh Bagchi. Vision-only gaussian splatting for collaborative semantic occupancy prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2796–2804, 2026.

Abdullah Enes Doruk and Hasan F. Ates. Gaussianocc3d: A gaussian-based adaptive multi-modal 3d occupancy prediction, 2026. URL https://arxiv.org/abs/2601.22729.

Zaipeng Duan, Chenxu Dang, Xuzhong Hu, Pei An, Junfeng Ding, Jie Zhan, Yunbiao Xu, and Jie Ma. Sdgocc: Semantic and depth-guided bird's-eye view transformation for 3d multimodal occupancy prediction, 2025. URL https://arxiv.org/abs/2507.17083.

Jingming He, Chongyi Li, Shiqi Wang, and Sam Kwong. Joint semantic and rendering enhancements in 3d gaussian modeling with anisotropic local encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28354–28363, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/He_Joint_Semantic_and_Rendering_Enhancements_in_3D_Gaussian_Modeling_with_ICCV_2025_paper.html.

Yuanhui Huang, Amonnut Thammatadatrakoon, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, and Jiwen Lu. Gaussianformer-2: Probabilistic gaussian superposition for efficient 3d occupancy prediction, 2024a. URL https://arxiv.org/abs/2412.04384.

Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction, 2024b. URL https://arxiv.org/abs/2405.17429.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views, 2025. URL https://arxiv.org/abs/2505.23716.

In-Hwan Jin, Hyeongju Mun, Joonsoo Kim, Kugjin Yun, and Kyeongbo Kong. On the design of mixture-of-experts for dynamic gaussian splatting, 2026. URL https://arxiv.org/abs/2607.08250.

Injae Kim, Chaehyeon Kim, Minseong Bae, Minseok Joo, and Hyunwoo J. Kim. F4splat: Feed-forward predictive densification for feed-forward 3d gaussian splatting, 2026. URL https://arxiv.org/abs/2603.21304.

Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M. Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion, 2023. URL https://arxiv.org/abs/2302.12251.

Yiming Li, Sihang Li, Xinhao Liu, Moonjun Gong, Kenan Li, Nuo Chen, Zijun Wang, Zhiheng Li, Tao Jiang, Fisher Yu, Yue Wang, Hang Zhao, Zhiding Yu, and Chen Feng. SSCBench: Monocular 3d semantic scene completion benchmark in street views. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Chengxin Lv, Yihui Li, Hongyu Yang, and YunHong Wang. Gau-occ: Geometry-completed gaussians for multi-modal 3d occupancy prediction, 2026. URL https://arxiv.org/abs/2603.22852.

Arthur Moreau, Richard Shaw, Michal Nazarczuk, Jisu Shin, Thomas Tanay, Zhensong Zhang, Songcen Xu, and Eduardo Pérez-Pellitero. Off the grid: Detection of primitives for feed-forward 3d gaussian splatting, 2025. URL https://arxiv.org/abs/2512.15508.

Seungtae Nam, Xiangyu Sun, Gyeongjin Kang, Younggeun Lee, Seungjun Oh, and Eunbyung Park. Generative densification: Learning to densify gaussians for high-fidelity generalizable 3d reconstruction, 2024. URL https://arxiv.org/abs/2412.06234.

Rui Qian, Haozhi Cao, Tianchen Deng, Shenghai Yuan, and Lihua Xie. SplatSSC: Decoupled depth-guided gaussian splatting for semantic scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8520–8528, 2026. doi: 10.1609/aaai.v40i10.37803. URL https://ojs.aaai.org/index.php/AAAI/article/view/37803.

Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, 2020. doi: 10.1109/3DV50981.2020.00021. URL https://doi.org/10.1109/3DV50981.2020.00021.

Yu Sheng, Jiajun Deng, Xinran Zhang, Yu Zhang, Bei Hua, Yanyong Zhang, and Jianmin Ji. Spatialsplat: Efficient semantic 3d from sparse unposed images, 2025. URL https://arxiv.org/abs/2505.23044.

Yunxiao Shi, Yinhao Zhu, Herbert Cai, Shizhong Han, Jisoo Jeong, Amin Ansari, and Fatih Porikli. Odg: Occupancy prediction using dual gaussians. *Advances in Neural Information Processing Systems*, 38:32104–32125, 2026.

Ke Song, Yunhe Wu, Chunchit Siu, and Huiyuan Xiong. Graphgsocc: Semantic and geometric graph transformer for 3d gaussian splating-based occupancy prediction, 2025. URL https://arxiv.org/abs/2506.14825.

Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving, 2023. URL https://arxiv.org/abs/2304.14365.

Yecong Wan, Fan Li, Mingwen Shao, and Wangmeng Zuo. Splatweaver: Learning to allocate gaussian primitives for generalizable novel view synthesis, 2026. URL https://arxiv.org/abs/2605.07287.

Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving, 2023. URL https://arxiv.org/abs/2303.09551.

Chi Yan and Dan Xu. Progressive gaussian transformer with anisotropy-aware sampling for open vocabulary occupancy prediction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mHFaflQv93.

Xiaoyang Yan, Muleilan Pei, and Shaojie Shen. St-gs: Vision-based 3d semantic occupancy prediction with spatial-temporal gaussian splatting, 2025. URL https://arxiv.org/abs/2509.16552.

Xiaoyang Yan, Muleilan Pei, and Shaojie Shen. Vg3s: Visual geometry grounded gaussian splatting for semantic occupancy prediction. *arXiv preprint arXiv:2603.06210*, 2026.

Zhen Yang, Yanpeng Dong, Jiayu Wang, Heng Wang, Lichao Ma, Zijian Cui, Qi Liu, Haoran Pei, Kexin Zhang, and Chao Zhang. Daocc: 3d object detection assisted multi-sensor fusion for 3d occupancy prediction, 2024. URL https://arxiv.org/abs/2409.19972.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details for 3d gaussian splatting, 2024. URL https://arxiv.org/abs/2404.10484.

Jiaxin Zhang, Junjun Jiang, Youyu Chen, Kui Jiang, and Xianming Liu. COB-GS: Clear object boundaries in 3DGS segmentation based on boundary-adaptive gaussian splitting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19335–19344, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhang_COB-GS_Clear_Object_Boundaries_in_3DGS_Segmentation_Based_on_Boundary-Adaptive_CVPR_2025_paper.html.

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting, 2024. URL https://arxiv.org/abs/2403.15530.

Lingjun Zhao, Sizhe Wei, James Hays, and Lu Gan. Gaussianformer3d: Multi-modal gaussian-based semantic occupancy prediction with 3d deformable attention, 2025. URL https://arxiv.org/abs/2505.10685.

Changqing Zhou, Yueru Luo, and Changhao Chen. Generalizing visual geometry priors to sparse gaussian occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28578–28587, 2026.

Sicheng Zuo, Wenzhao Zheng, Xiaoyong Han, Longchao Yang, Yong Pan, and Jiwen Lu. Quadricformer: Scene as superquadrics for 3d semantic occupancy prediction. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/4470d0b5795c3721e2a622a249ef2154-Abstract-Conference.html.

A Method, Diagnostic, and Reproducibility Details

A.1 Allocation Diagnostic Metrics

We report occupancy IoU, semantic mIoU, and class-wise IoU, and use GT-support label purity, coverage, and unused-Gaussian ratio as diagnostic representation metrics. For an occupied voxel v_j centered at $\boldsymbol{x}_j$ and a Gaussian $\mathcal{G}_i$ with covariance $\boldsymbol{\Sigma}_i = \boldsymbol{R}_i \operatorname{diag}((\boldsymbol{s}_i)^2)\boldsymbol{R}_i^\top$, coverage is measured by

$$d_{ij} = \sqrt{(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}. \tag{9}$$

An occupied voxel is covered when $d_{ij} < \tau_d$. Let $\mathcal{V}_i = \{v_j \in \mathcal{V}_{\text{occ}} \mid d_{ij} < \tau_d\}$ and y_j be the voxel label. The per-Gaussian and valid-Gaussian GT-support purities are

$$\begin{aligned} P_i &= \frac{\max_c |\{v_j \in \mathcal{V}_i \mid y_j = c\}|}{|\mathcal{V}_i|}, \\ P_{\text{valid}} &= \frac{1}{N_{\text{valid}}} \sum_{i:|\mathcal{V}_i|>0} P_i. \end{aligned} \tag{10}$$

These post-hoc diagnostics measure the GT-label homogeneity of each Gaussian's spatial support. They are distinct from $\mathcal{L}_{\text{ent}}$, which regularizes the entropy of predicted semantic distributions and does not directly enforce GT-support homogeneity or predicted-class correctness. For category-wise diagnostics, P_c averages $|\{v_j \in \mathcal{V}_i \mid y_j = c\}|/|\mathcal{V}_i|$ over Gaussians covering class c; foreground, background, and macro entries are unweighted class averages and are not a decomposition of P_{valid}. Let $\hat{c}_i = \arg\max_c |\{v_j \in \mathcal{V}_i \mid y_j = c\}|$ be the majority class of a valid Gaussian and $\mathcal{G}_{\text{clean}}(\rho) = \{i \mid |\mathcal{V}_i| > 0, P_i \geq \rho\}$ be the set of non-mixed valid Gaussians. Coverage, utilization, Mixed-Gaussian Ratio (Mix-G), and Class-consistent Semantic Support (Sem-Sup) are

$$\begin{aligned} C_{\text{ratio}} &= \frac{|\{v_j \in \mathcal{V}_{\text{occ}} \mid \min_i d_{ij} < \tau_d\}|}{|\mathcal{V}_{\text{occ}}|}, \\ U_{\text{ratio}} &= \frac{|\{i \mid \mathcal{G}_i \text{ covers no visible occupied voxel}\}|}{N}, \\ \text{Mix-G} &= \frac{1}{N_{\text{valid}}} \sum_{i:|\mathcal{V}_i|>0} \mathbb{1}[P_i < \rho], \\ \mathcal{S}_{\text{cc}} &= \{v_j \in \mathcal{V}_{\text{occ}} \mid \exists i \in \mathcal{G}_{\text{clean}}(\rho), \\ &\qquad d_{ij} < \tau_d,\ \hat{c}_i = y_j\}, \\ \text{Sem-Sup} &= \frac{|\mathcal{S}_{\text{cc}}|}{|\mathcal{V}_{\text{occ}}|}. \end{aligned} \tag{11}$$

We report penalized GT-support purity as $P_{\text{pen.}} = (1 - U_{\text{ratio}})P_{\text{valid}}$, together with final physical Gaussian count, memory, and FPS. For allocation diagnostics, we set $\rho = 0.6$ and use the same τ_d and evaluation procedure across methods. These GT-dependent diagnostics are used only for post-hoc evaluation, are never provided to the allocation module, and do not replace occupancy accuracy.

A.2 Implementation and Comparison Protocol

Across the full nuScenes validation set, with one raw scene frame per sample, 12% of initial Gaussians have more than eight other centers inside $R = 2\,\text{m}$ before truncation at the default $K_{\text{nbr}} = 8$. The 10.33K initial count refers to one checked frame; the validation-set average is approximately 10.38K.

Table 14 shows that, relative to $K_{\text{nbr}} = 8$, $K_{\text{nbr}} = 4$ lowers overall mIoU by 0.34 point and dense-stratum mIoU by 0.70 point. Increasing to $K_{\text{nbr}} = 16$ raises overall mIoU by 0.05 point and dense-stratum mIoU by 0.17 point, while sparse-stratum mIoU changes by -0.01 point. These measured differences do not establish the optimal neighbor count beyond the tested settings.

We use a ResNet-50/FPN image encoder and the sparse convolutional ResNet LiDAR encoder used in CenterPoint. Semantic Gaussians are initialized from non-empty voxels: the voxel center gives

Table 8: Full comparison on nuScenes-SurroundOcc. C/L denote camera/LiDAR, and 1/M denote single-/multi-sweep LiDAR. * uses one raw LiDAR sweep at inference with temporally aggregated training supervision following the documented Gau-Occ schedule; † marks reported results, ‡ our reproduction, and – unavailable results. SAGFormer variants report mean ± std. over seeds 42, 37, and 91; the two 16.6K camera-only plug-in rows are single-run results.

Method	Mod.	Frames	Venue	Class-wise IoU														
				Bar.	Bicy.	Bus	Car	C.Veh.	Moto.	Ped.	T.C.	Trail.	Truck	D.S.	O.F.	Side.	Terr.	Manm.	Veg.
Camera-only methods																			
SurroundOcc	C	1	ICCV'23	20.59	11.68	28.06	30.86	10.70	15.14	14.09	12.06	14.38	22.26	37.29	23.70	24.49	22.77	14.89	21.86
GaussianFormer	C	1	ECCV'24	19.52	11.26	26.11	29.78	10.47	13.83	12.58	8.67	12.74	21.57	39.63	23.28	24.46	22.99	9.59	19.12
GaussianFormer-2	C	1	CVPR'25	21.39	13.44	28.49	30.82	10.92	15.84	13.55	10.53	14.04	22.92	40.61	24.36	26.08	24.27	13.83	21.98
GaussianFormer‡ (16.6K)	C	1	Reproduced	18.65	11.90	25.62	29.63	10.70	15.61	12.82	9.36	12.83	20.52	39.18	21.83	24.33	22.24	10.68	19.70
+ SAGFormer allocation (16.6K)	C	1	Reproduced	23.05	16.52	28.75	33.68	15.98	20.40	24.29	12.95	17.66	27.61	31.55	16.25	21.02	23.52	27.84	37.58
Multi-modal methods																			
GaussianFormer3D†	C+L	M (10)	ICRA'26	26.9	15.8	32.7	36.1	18.6	21.7	24.1	13.0	21.3	29.0	40.6	23.7	27.3	28.2	32.6	42.3
GaussianFormer3D‡	C+L	1	Reproduced	21.56	14.03	28.63	29.83	15.21	17.20	20.64	6.54	16.61	22.66	34.18	16.88	22.91	25.06	28.95	37.51
GaussianOcc3D†	C+L	M	arXiv'26	28.7	16.8	34.8	38.5	19.8	23.1	25.7	13.9	22.7	30.9	43.3	25.3	29.1	30.1	34.7	45.1
DAOcc†	C+L	M (10)	arXiv'24	30.8	19.5	34.0	38.8	25.0	27.7	29.9	22.5	23.2	31.6	41.0	25.9	29.4	29.9	35.2	43.5
DAOcc‡	C+L	1	Reproduced	25.29	17.56	29.76	32.35	21.43	23.03	26.27	15.87	18.33	25.09	34.40	18.91	24.83	26.59	31.38	38.51
SDG-Fusion†	C+L	M	CVPR'25	31.2	15.6	39.7	42.4	22.2	26.9	29.7	22.7	24.2	33.2	45.9	25.1	31.8	33.5	38.9	44.5
SDG-Fusion‡	C+L	1	Reproduced	25.28	13.14	35.01	35.57	18.16	21.79	25.60	15.70	18.91	26.32	38.93	17.75	26.79	29.71	34.62	39.12
Gau-Occ†	C+L	1*	CVPR'26	33.1	16.5	41.1	43.9	21.9	26.6	31.1	23.1	24.5	34.0	49.0	26.3	32.9	33.4	39.3	45.7
SAGFormer w/o Semantic Cues (Ours)	C+L	1	–	28.51	21.72	33.04	36.68	19.09	26.98	28.47	20.53	20.80	29.96	36.00	21.26	25.19	25.68	30.48	38.43
SAGFormer (Ours)	C+L	1	–	29.82	21.13	33.82	37.30	18.59	26.25	27.83	18.74	22.93	31.16	39.19	23.19	27.21	27.15	31.25	39.88
SAGFormer (Ours)	C+L	1*	–	33.89	23.98	32.49	39.53	24.83	28.69	26.26	25.77	24.76	32.38	46.98	28.35	33.11	34.83	43.25	51.43

Table 9: Full comparison on SSCBench-KITTI-360. C/L denote camera/LiDAR. * uses one raw LiDAR sweep at inference with temporally aggregated training supervision following the documented Gau-Occ schedule; † marks reported results and – unavailable results. SAGFormer variants report mean ± std. over seeds 42, 37, and 91.

Method	Mod.	Frames	Venue	Class-wise IoU															
				Car	Bicy.	Moto.	Truck	O.Veh.	Pers.	Road	Park.	Side.	O.Gr.	Bldg.	Fence	Veg.	Terr.	Pole	T.Sign	O.Str.	O.Obj.
Camera-only methods																			
MonoScene	C	1	CVPR'22	19.34	0.43	0.58	8.02	2.03	0.86	48.35	11.38	28.13	3.22	32.89	3.53	26.15	16.75	6.92	5.67	4.20	3.09
VoxFormer	C	1	CVPR'23	17.84	1.16	0.89	4.56	2.06	1.63	47.01	9.67	27.21	2.89	31.18	4.97	28.99	14.69	6.51	6.92	3.79	2.43
GaussianFormer	C	1	ECCV'24	18.93	1.02	4.62	18.07	7.59	3.35	45.47	10.89	25.03	5.32	28.44	5.68	29.54	8.62	2.99	2.32	9.51	5.14
GaussianFormer-2	C	1	CVPR'25	21.08	2.55	4.21	12.41	5.73	1.59	54.12	11.04	32.31	3.34	32.01	4.98	28.94	17.33	3.57	5.48	5.88	3.54
LiDAR and multi-modal methods																			
LMSCNet	L	1	3DV'20	20.91	0.00	0.00	0.26	0.00	0.00	62.95	13.51	33.51	0.20	43.67	0.33	40.01	26.80	0.00	0.00	3.63	0.00
GaussianFormer3D	C+L	10	ICRA'26	36.95	2.60	1.55	22.08	6.74	4.96	63.80	15.99	43.32	4.68	47.79	9.59	41.00	27.62	22.32	18.97	10.06	3.38
Gau-Occ†	C+L	1*	CVPR'26	42.1	3.7	4.4	27.2	10.8	10.5	70.9	22.5	47.1	9.4	51.7	14.6	47.0	32.2	25.2	22.7	14.1	7.8
SAGFormer w/o Semantic Cues (Ours)	C+L	1	–	39.71	3.35	3.21	26.13	7.69	8.06	64.64	20.56	44.74	9.42	50.24	14.81	41.98	30.45	22.64	23.07	11.79	5.19
SAGFormer (Ours)	C+L	1	–	40.35	3.59	4.10	26.12	8.83	7.80	65.93	20.15	46.28	8.76	51.93	14.00	43.77	29.54	24.58	22.01	13.07	4.79
SAGFormer (Ours)	C+L	1*	–	42.98	6.78	8.19	28.99	9.67	8.78	69.59	21.17	48.91	9.12	54.61	16.27	46.70	34.78	28.58	23.40	16.18	7.70

Table 10: Camera-only plug-in generalization on nuScenes-SurroundOcc. The two single-run rows share the GaussianFormer image encoder, Gaussian initializer, voxel decoder, training objectives, and exact 16.6K final count; only the SAGFormer allocation module is added.

Camera-only host	Allocation	Final #G	IoU ↑	mIoU ↑
GaussianFormer	None	16.6K	29.60	19.10
GaussianFormer	SAGFormer	16.6K	33.04	23.67

Table 11: Category-wise class-conditioned GT-support purity on nuScenes-SurroundOcc.

Method	Bar.	Bicy.	Bus	Car	C.Veh.	Moto.	Ped.	T.C.	Trail.	Truck	D.S.	O.F.	Side.	Terr.	Manm.	Veg.
GaussianFormer	0.115	0.083	0.191	0.412	0.177	0.059	0.100	0.028	0.157	0.293	0.797	0.183	0.423	0.369	0.639	0.701
GaussianFormer3D	0.127	0.138	0.315	0.591	0.114	0.105	0.210	0.049	0.138	0.387	0.806	0.109	0.400	0.352	0.738	0.804
SAGFormer w/o Allocation	0.147	0.058	0.255	0.568	0.124	0.129	0.198	0.036	0.110	0.383	0.774	0.136	0.440	0.345	0.744	0.801
Candidate-family-matched non-semantic control	0.143	0.064	0.248	0.566	0.123	0.124	0.201	0.030	0.111	0.380	0.705	0.067	0.371	0.276	0.675	0.732
SAGFormer	0.134	0.086	0.232	0.583	0.238	0.130	0.208	0.047	0.156	0.425	0.796	0.164	0.447	0.352	0.728	0.797

the mean, and an MLP predicts scale, rotation, opacity, semantic scores, and primitive features from the voxel feature. All models use Adam with learning rate 2×10^{-4} and a one-cycle schedule. Experiments run on four RTX 4090 GPUs with PyTorch 2.12.0, CUDA 12.6, and cuDNN 9.

The allocation encoder has two global self-attention Transformer blocks with hidden width 128, eight attention heads, a 512-dimensional feed-forward layer, LayerNorm, and dropout probability

Table 12: Distance-wise allocation quality on nuScenes-SurroundOcc. Entries are GT-support purity / visible non-empty coverage; Far Avg. averages 30–50m.

Method	0–10m	10–20m	20–30m	30–40m	40–50m	Far Avg.
GaussianFormer	0.670 / 1.000	0.665 / 1.000	0.651 / 1.000	0.632 / 0.999	0.615 / 0.995	0.624 / 0.997
GaussianFormer3D	0.696 / 1.000	0.719 / 0.991	0.731 / 0.944	0.729 / 0.853	0.743 / 0.711	0.736 / 0.782
SAGFormer w/o Allocation	0.737 / 0.999	0.728 / 0.994	0.719 / 0.958	0.700 / 0.874	0.702 / 0.737	0.701 / 0.806
SAGFormer	**0.749** / 0.999	**0.733** / 0.994	**0.723** / 0.960	**0.713** / 0.874	**0.720** / 0.732	**0.716** / 0.803

Table 13: Sensitivity to Mahalanobis radius τ_d on nuScenes-SurroundOcc. Entries are Full / matched Transformer w/o Semantic Cues.

τ_d	P_{valid}	$P_{\mathrm{pen.}}$	Mean Cov.	Unused ↓
2	0.7242 / 0.6927	0.6728 / 0.6411	0.8986 / 0.8972	0.0927 / 0.0989
3	0.7155 / 0.6861	0.6593 / 0.6279	0.9103 / 0.9108	0.0785 / 0.0843
4	0.7135 / 0.6852	0.6538 / 0.6230	0.9189 / 0.9187	0.0669 / 0.0731

Table 14: Local-neighbor sensitivity on nuScenes-SurroundOcc at $R = 2\,\mathrm{m}$ and $B = 0.6$. Each initial Gaussian is grouped by raw LiDAR points within $2\,\mathrm{m}$ of its center: sparse < 4, dense ≥ 4 (reported group ratio 0.6/0.4). Stratum mIoU uses associated GT occupied voxels.

K_{nbr}	mIoU ↑	Sparse mIoU ↑	Dense mIoU ↑
4	28.13	28.62	27.39
8	28.47	**28.72**	28.09
16	**28.52**	28.71	**28.26**

Table 15: Existing budget-wise allocation diagnostics on nuScenes-SurroundOcc, measured on the final materialized physical set.

B	P_{valid} ↑	Mean Cov.	Unused ↓
0.2	0.7135	0.9049	**0.0713**
0.4	0.7144	0.9101	0.0734
0.6	**0.7155**	0.9103	0.0785
0.8	0.7147	0.9143	0.0792
1.0	0.7145	0.9193	0.0801

Table 16: Stability audit on SSCBench-KITTI-360. The w/o Allocation row is the non-allocation reference at the same input setting.

Method	B	Final Phys. #G	IoU ↑	mIoU ↑	Soft eff. ratio	Mat. mIoU drop
SAGFormer w/o Allocation	0.6	18.4K	55.34	23.34	1.61	0.031
SAGFormer	0.4	16.2K	57.10	24.01	1.44	0.072
SAGFormer	0.6	18.6K	57.40	24.20	1.63	0.026
SAGFormer	0.8	21.2K	57.43	24.31	1.87	0.025

0.01. The eight-neighbor summary used as input to these blocks does not restrict their attention range.

For the camera-only plug-in control, we use the GaussianFormer image encoder, Gaussian initializer, and voxel decoder, and insert SAGFormer between initialization and decoding. The capacity-matched reference and plug-in variant share the same 16.6K final Gaussian count, optimization, augmentation, objectives, evaluation code, and seeds; their only intended difference is adaptive allocation.

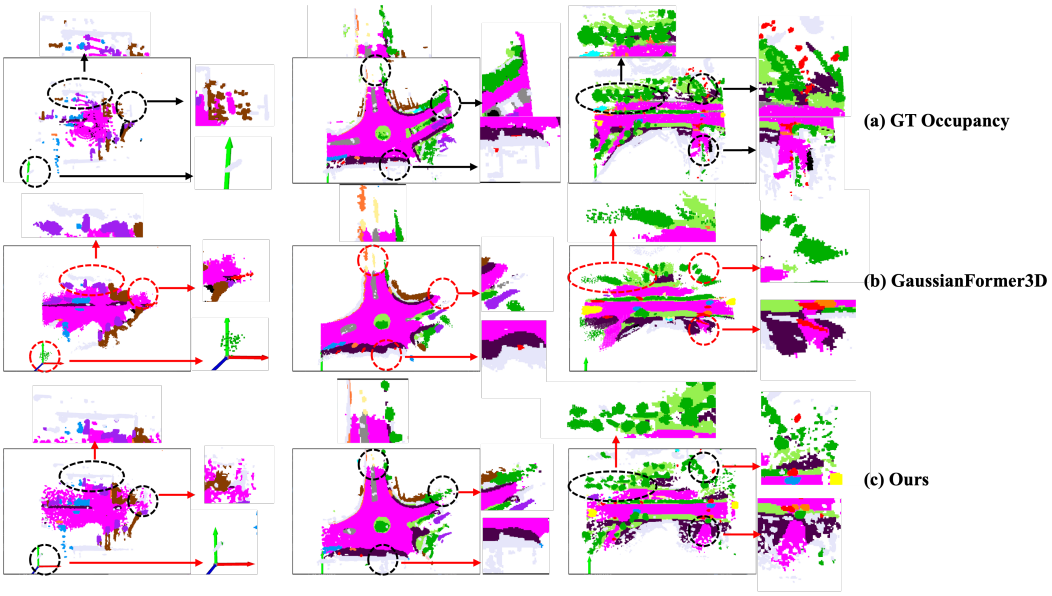


Figure 3: Qualitative comparison on nuScenes-SurroundOcc. Zoomed regions highlight small objects, boundaries, distant areas, and static structures where SAGFormer produces more complete occupancy and more coherent semantics.

Occupancy metrics, Final Phys. #G, memory, and FPS are measured on the final materialized physical set, while gate-weighted effective cardinality is reported only as a training-time allocation audit. We fix $\lambda_{\text{lov}} = 1.0$, $\lambda_{\text{ent}} = 0.1$, $\lambda_{\text{rec}} = 0.2$, $\lambda_{\text{bud}} = 0.3$, and $\tau_d = 3$ unless stated otherwise.

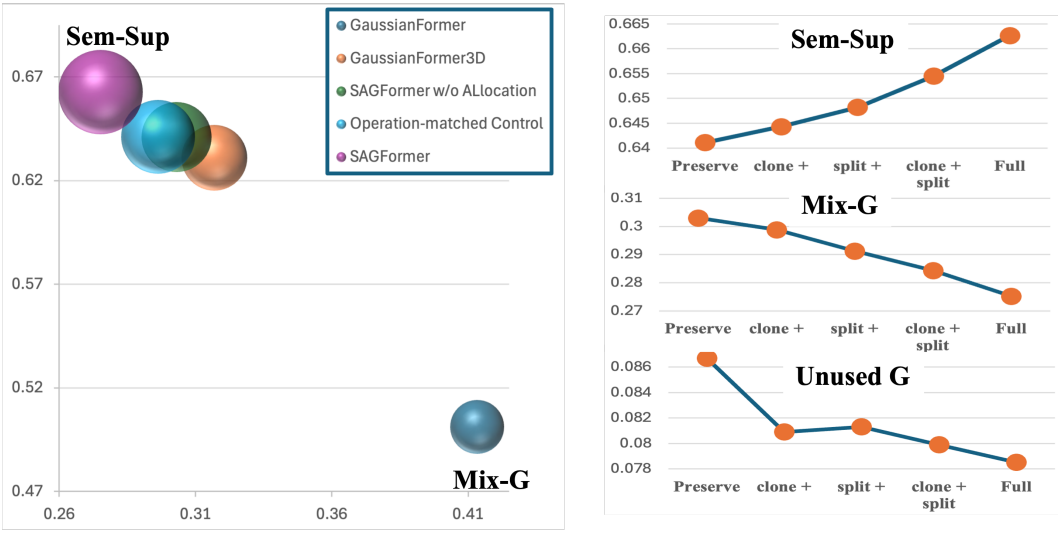
Because Gau-Occ has no public code, checkpoints, or complete training details, its completion-enhanced 1^* values are contextual cross-paper references rather than implementation-matched or paired comparisons. Hardware-efficiency comparisons are limited to implementations measured in our environment.

B Full Class-Wise Benchmark Results

In the single-run camera-only plug-in comparison, SAGFormer improves 13 of 16 classes but reduces driveable-surface, other-flat, and sidewalk IoU. In the matched raw multimodal comparison, adding semantic allocation cues improves 11 of 16 nuScenes classes and 10 of 18 KITTI-360 classes. NuScenes bicycle, construction vehicle, motorcycle, pedestrian, and traffic-cone IoU decline; KITTI-360 includes declines for person, parking, fence, terrain, and traffic sign. The aggregate mIoU gains therefore should not be read as uniform improvements. The completion-enhanced rows use a different training setting and are shown for context.

C Extended Allocation Diagnostics

Across distance bins, SAGFormer increases GT-support purity while keeping visible non-empty coverage comparable; the far-range purity over 30–50m rises from 0.701 to 0.716. For $\tau_d \in \{2, 3, 4\}$, it consistently improves valid GT-support purity by 0.0283–0.0315 and penalized GT-support purity by 0.0308–0.0317, with nearly identical mean coverage and lower unused ratio.



(a) Allocation Bottleneck Map **(b) Operation Diagnostic Trajectory**

Figure 4: Allocation diagnostics on nuScenes-SurroundOcc: (a) method-level Mix-G versus Sem-Sup and (b) candidate-family trajectories for Sem-Sup, Mix-G, and unused ratio. Only methods whose final Gaussian outputs can be inspected are included.

D Additional Ablations and Stability

Table 15 shows that nuScenes purity remains stable while coverage increases smoothly with the budget; the unused ratio rises only moderately. On KITTI-360, accuracy and the materialization gap change smoothly across budgets.

E Qualitative Results

The qualitative examples are consistent with the allocation metrics: SAGFormer improves boundary and background regions by assigning capacity to difficult local structures while suppressing ineffective primitives.

F Additional Comparison and Ablation Tables

The following tables provide the detailed comparisons and allocation ablations referenced in the main text.

Table 17: Comparison with closely related methods. They improve primitive shape, initialization, or progressive densification. SAGFormer studies candidate selection under a nominal final budget.

Method	Main goal	When primitives change	Main signal	Count rule	Task
SplatSSC Qian et al. (2026)	Compact initial set	Initialization	Image and depth features	Sparse initial set	Monocular SSC
QuadricFormer Zuo et al. (2025)	Flexible primitive shape	After initial training	Superquadric scale	Prune and split; count unchanged	Closed-set occupancy
PG-Occ Yan & Xu (2026)	Recover fine details	Progressive layers	Rendered and reference depth	Adds queries at each layer	Open-vocabulary occupancy
F4Splat/SplatWeaver Kim et al. (2026); Wan et al. (2026)	Generate rendering GS	Image-aligned generation	Photometric cues	Scale threshold / pixel-wise count	Novel-view synthesis
SAGFormer (ours)	Select useful candidates	Post-initialization	Geometry, local density, and semantic consistency	Global score-based selection	Closed-set occupancy

Table 18: Candidate-family sweep on nuScenes-SurroundOcc. All rows start from approximately 10.38K Gaussians and use the same training objectives, data, evaluation, and seeds. Final physical counts vary across rows; only the last two rows are matched at 16.6K.

Variant	Clone	Split	Suppress	Initial #G	Final #G	mIoU ↑
Preserve only	–	–	–	10.38K	10.38K	27.00 ± 0.06
+ Clone	✓	–	–	10.38K	13.32K	27.71 ± 0.04
+ Split	–	✓	–	10.38K	13.42K	27.69 ± 0.04
+ Clone + Split	✓	✓	–	10.38K	16.6K	28.05 ± 0.05
Full	✓	✓	✓	10.38K	16.6K	$\mathbf{28.47 \pm 0.06}$

Table 19: Comparison of matched Gaussian allocation policies on nuScenes-SurroundOcc. QuadricFormer-, F4Splat-, and PG-Occ-style rows are in-system rule adaptations inspired by the cited papers; they use our initial Gaussian set, occupancy backbone, decoder, training objectives, and input modality, and pass 16.6K final Gaussians to the decoder. They are not full-system reproductions. ⋆The capacity-matched w/o Allocation variant here and the diagnostic control in Table 4 use 16.6K final physical Gaussians. The efficiency reference in Table 7 uses 10.38K.

Variant	IoU ↑	mIoU ↑	Mean Cov.	P_{valid} ↑	$P_{\text{pen.}}$ ↑	Unused ↓
w/o Allocation⋆	40.4 ± 0.12	27.33 ± 0.06	0.9108	0.7100	0.6484	0.0867
Opacity Top-K	40.1 ± 0.04	27.33 ± 0.05	–	–	–	–
Random Allocation	40.5 ± 0.13	27.53 ± 0.05	0.9102	0.7101	0.6492	0.0833
QuadricFormer-style Zuo et al. (2025)	40.5 ± 0.11	27.56 ± 0.05	0.9103	0.7112	0.6497	0.0823
F4Splat-style Kim et al. (2026)	40.9 ± 0.12	27.89 ± 0.06	0.9102	0.7123	0.6503	0.0809
PG-Occ-style Yan & Xu (2026)	41.0 ± 0.12	28.03 ± 0.05	0.9103	0.7134	0.6553	0.0792
MLP Controller	41.0 ± 0.11	28.01 ± 0.04	0.9101	0.7132	0.6544	0.0799
Transformer Controller	41.7 ± 0.11	28.47 ± 0.06	0.9103	**0.7155**	**0.6593**	**0.0785**

Table 20: Budget sensitivity and the gap between soft routing and final selection on nuScenes-SurroundOcc. Final #G is the post-allocation frame mean. Soft count ratio is the validation mean of $\bar{C}_\pi$ in Eq. (6). Selection drop compares soft routing with the final materialized set. Results are mean $\pm$ std. where available.

	Final-count evaluation			Final-selection check	
B	Final #G	IoU ↑	mIoU ↑	Soft count ratio	Selection drop
0.2	12.46K	40.90	27.69	1.23	0.321
0.4	14.53K	41.20 ± 0.12	28.24 ± 0.05	1.41	0.132
0.6	16.6K	41.74 ± 0.11	28.47 ± 0.06	1.63	0.031
0.8	18.68K	41.80 ± 0.12	28.35 ± 0.06	1.82	0.023
1.0	20.05K	41.60	28.31	1.98	0.022