UNIGAUSSIANGPT: COMPOSABLE AUTOREGRESSIVE SEMANTIC 3D GAUSSIANS MODELING

Anonymous authors
Paper under double-blind review

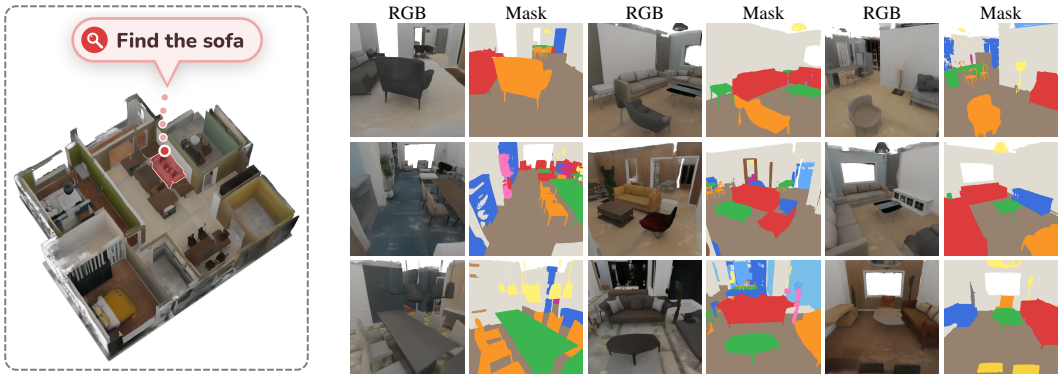

Figure 1: UniGaussianGPT generates 3D Gaussian scenes together with their semantic categories. **Left:** every decoded Gaussian carries a predicted category. **Right:** RGB renderings of scenes with the predicted semantic masks.

ABSTRACT

Recent advances in 3D generative modeling have enabled the synthesis of detailed scenes with coherent geometry and appearance, mostly through diffusion models that refine an entire scene holistically. Autoregressive formulations offer a complementary paradigm that constructs Gaussian scenes step by step, bringing a compositional inductive bias under which generation and completion follow from the same next-token prediction. However, these advances largely focus on producing renderable scenes, while recognizing their contents remains a separate task. Semantic information links geometry to the objects and regions it represents, providing a foundation for scene understanding and downstream applications such as robotics and AR/VR. We therefore present UniGaussianGPT, an autoregressive framework that unifies 3D Gaussian scene generation and semantic understanding by learning to predict scene contents and their semantics from shared contextual representations. To support this learning, we build labeled training scenes by composing Gaussian assets converted from complete furniture meshes with separately fitted empty rooms, and supervise Gaussian attributes directly so that occluded surfaces survive tokenization. Generated scenes are thus composable, consisting of semantically labeled and individually complete components. Extensive experiments on composed 3D-FRONT scenes show that UniGaussianGPT achieves state-of-the-art performance in both semantic understanding and scene generation and completion, while better retaining occluded content through tokenization.

1 INTRODUCTION

3D scene generation aims to synthesize coherent environments with coherent appearance and spatial structure. Gaussian scene representations make this content explicit and renderable from different viewpoints (Roessle et al., 2024; von Lützow et al., 2026). Diffusion and flow-based approaches synthesize scenes through iterative refinement of latent representations (Roessle et al., 2024; Wu et al., 2024c; Meng et al., 2025; Li et al., 2026). Yet scene construction also involves completing

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.

partial environments and extending existing ones, where each addition must fit the content already present. This makes the relationship between a partial scene and its continuation an important part of the generative formulation (von Lützow et al., 2026).

Autoregressive modeling expresses this relationship directly by conditioning each new spatial element on the preceding sequence. GaussianGPT realizes this process with discrete tokens that decode into Gaussian geometry and appearance (von Lützow et al., 2026). The generated continuation can then become context for further expansion, enabling outpainting beyond the training chunk. Resolving construction through conditional decisions also allows sampling to be constrained or revised as new content is added, providing a flexible basis for extending scenes while controlling how their content is constructed.

Existing generative formulation, however, describes what a scene looks like but not what it contains. The generated Gaussians carry geometry and appearance, yet nothing indicates their semantics. Local shape and surrounding arrangement are the evidence by which a region is recognized, while a category summarizes the geometry and layout its instances tend to share. A common remedy is to label the scene afterwards with a separate predictor, as existing Gaussian understanding methods do (Ye et al., 2024; Qin et al., 2024; Wu et al., 2024b; Li et al., 2025b). Such a decoupled pipeline treats recognition as annotation of finished geometry: the labeler cannot draw on the context from which the content was generated, errors in the generated geometry carry over into its labels, and the generator never benefits from the category structure that could organize its layouts. We instead study recognition and generation within a single autoregressive scene model, aiming to make category recognition a learned capability of the scene representation itself, available both when interpreting an existing scene and when generating new content.

We present UniGaussianGPT, to the best of our knowledge, the first autoregressive framework for semantic 3D Gaussian scene generation. Its central design is to learn category prediction and scene continuation from the same contextual states. A state is read after the current feature token has been consumed, so its semantic prediction can use the content actually represented at that location together with the preceding spatial context. The same state predicts the next scene token. Joint supervision therefore adapts the representation for both recognition and synthesis, rather than only fitting a classifier to frozen generative features. Spatial correspondence transfers the predicted categories to decoded Gaussians. No category tokens are added or fed back into the sequence, preserving the original autoregressive sampling procedure while making semantics an output of the scene model. The resulting scenes can then serve not only viewers but also agents that act on specific objects, such as robots that must locate, grasp, or navigate around them.

The structure a generative model can learn is shaped in part by what its training representation preserves. Rather than fitting a furnished room directly from its views, we convert complete furniture meshes into Gaussian assets, prune them individually, and combine them with separately fitted empty rooms. Each component therefore retains surfaces that would be hidden in the assembled room. We further supervise opacity, scale, and color directly to help these surfaces survive tokenization. Generated scenes are thus composable, consisting of semantically labeled and individually complete components.

Extensive experiments demonstrate that UniGaussianGPT achieves state-of-the-art performance across the board. It outperforms dedicated Gaussian and feed-forward understanding methods in semantic prediction on held-out scenes, surpasses existing scene generators in unconditional generation and completion, and better retains occluded Gaussian content through tokenization.

In summary, our contributions are threefold: (i) We present UniGaussianGPT, to the best of our knowledge, the first autoregressive framework for generating 3D Gaussian scenes with semantic categories. (ii) We achieve the state-of-the-art performance on all evaluated semantic understanding metrics and on appearance and layout metrics for generation and completion, together with improved retention of occluded content during tokenization. (iii) The generated scenes contain semantically identifiable components that can be duplicated, removed, or rotated, suggesting potential applications in interactive scene construction and future embodied learning.

2 RELATED WORK

3D scene generation. Scene generation has been studied through both asset composition and direct synthesis of 3D content. Procedural systems assemble reusable objects into environments (Raistrick et al., 2024). Learned layout models predict object categories and their spatial arrangements (Paschalidou et al., 2021; Tang et al., 2024). Image-to-asset pipelines recover separate scene components from observations (Huang et al., 2025; Yao et al., 2025; Qin et al., 2026). A complementary line learns scene geometry in structured latent spaces (Wu et al., 2024c; Meng et al., 2025; Bokhovkin et al., 2025). Gaussian generators additionally model appearance, enabling renderable scene synthesis through latent diffusion (Roessle et al., 2024) and spatial expansion through block-wise generation (Li et al., 2026). These approaches provide different ways to construct scene content and control its organization. We study semantic scene synthesis within a next-token formulation, generating Gaussian content and its category assignments from a shared autoregressive representation.

Autoregressive 3D modeling. Autoregressive models factorize scene construction into predictions conditioned on existing content. ATISS predicts object categories and bounding boxes from a partial scene represented as an unordered set (Paschalidou et al., 2021). Its semantic outputs describe object layouts rather than dense Gaussian geometry and appearance. GaussianGPT instead serializes vector-quantized Gaussian scenes into spatial and feature tokens, enabling direct scene generation and completion through a causal transformer (von Lützow et al., 2026). We build on this representation and sampling procedure. Our focus is to make the contextual states that drive scene continuation jointly support semantic prediction, and to preserve the labeled 3D content from which those states are learned.

Scene Understanding and segment. Gaussian scene understanding methods associate primitives with masks, categories, or language features through distillation (Ye et al., 2024; Qin et al., 2024; Wu et al., 2024b), multi-view feature lifting (Cheng et al., 2024; Joseph et al., 2025), or direct 3D prediction (Li et al., 2025b; Wu et al., 2024a). Feed-forward models also combine geometric reconstruction with scene understanding (Fan et al., 2024; Zhou et al., 2026; Li et al., 2025a; Qu et al., 2026). In particular, IGGT couples geometry and instance feature prediction through a shared transformer and 3D-consistent supervision (Li et al., 2025a). We bring category prediction into the causal scene representation, learning semantic understanding together with the generation of Gaussian geometry and appearance. The resulting categories describe the same explicit 3D content used for rendering. Semantic grouping also supports basic component edits, complementing existing Gaussian selection and restoration methods (Ye et al., 2024; Wang et al., 2024; Liu et al., 2026).

3 PRELIMINARIES

We build on GaussianGPT (von Lützow et al., 2026), which represents a scene as Gaussian primitives $g_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \alpha_j, \mathbf{c}_j)$ with center, covariance, opacity, and color. A sparse convolutional autoencoder with lookup-free quantization (Yu et al., 2024) compresses the scene into a discrete latent grid. Each occupied cell i is serialized as a position token p_i followed by a feature token f_i, giving $\mathbf{x} = (p_1, f_1, \ldots, p_N, f_N)$. A causal transformer models the sequence using separate position and feature predictors. Omitting sequence boundary tokens for clarity, its likelihood and objective are

$$p_\theta(\mathbf{x}) = \prod_{t=1}^{2N} p_\theta\big(x_t \mid x_{<t}\big), \qquad \mathcal{L}_{\mathrm{LM}} = -\frac{1}{2N} \sum_{t=1}^{2N} \log p_\theta\big(x_t \mid x_{<t}\big). \tag{1}$$

Let $\mathbf{H}_t$ denote the final transformer state after consuming x_t, from which x_{t+1} is predicted. The Gaussian decoder progressively upsamples the latent grid, predicting occupancy at each level, and reconstructs primitive attributes in the occupied base voxels. The autoencoder is trained with

$$\mathcal{L}_{\mathrm{AE}} = \lambda_{\mathrm{RGB}}\mathcal{L}_{\mathrm{RGB}} + \lambda_{\mathrm{perc}}\mathcal{L}_{\mathrm{perc}} + \lambda_{\mathrm{occ}}\mathcal{L}_{\mathrm{occ}} + \lambda_{\mathrm{LFQ}}\mathcal{L}_{\mathrm{LFQ}}, \tag{2}$$

where $\mathcal{L}_{\mathrm{RGB}}$ is an L1 color loss and $\mathcal{L}_{\mathrm{perc}}$ a VGG-based perceptual loss between renderings of the decoded and input Gaussians from sampled viewpoints, $\mathcal{L}_{\mathrm{occ}}$ is a binary cross-entropy on the occupancy predicted in the decoder upsampling layers, and $\mathcal{L}_{\mathrm{LFQ}}$ is the codebook entropy regularizer of lookup-free quantization. Rendering terms supervise a primitive only through its contribution to

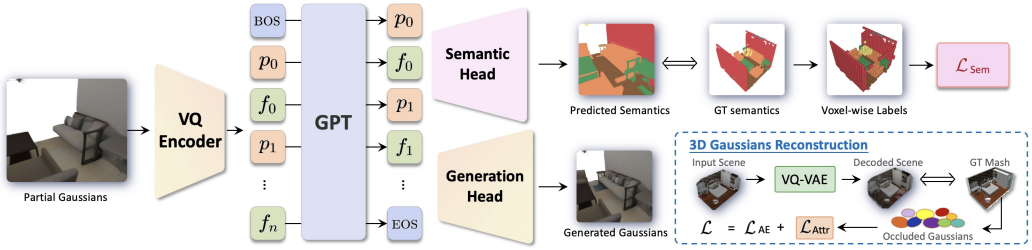

Figure 2: **Overview of UniGaussianGPT.** Given input Gaussians, a VQ encoder maps the scene to latent voxels serialized as position and feature tokens p_i, f_i, which a causal transformer models autoregressively. Its hidden states feed two heads: a generation head that predicts the next token, and a semantic head that predicts the category of each voxel from the state at its feature token (Sec. 4.2). Voxel-wise targets are majority-voted from ground-truth Gaussian categories and supervise the model through $\mathcal{L}_{\text{sem}}$. In the dashed box, the VQ-VAE is trained with the rendering-based $\mathcal{L}_{\text{AE}}$ and an additional attribute loss $\mathcal{L}_{\text{attr}}$ that supervises Gaussians directly from the composable assets, which helps to retain content occluded in furnished views (Sec. 4.3).

sampled views, while $\mathcal{L}_{\text{occ}}$ constrains where primitives exist but not their attributes. We retain the backbone and token format, with base and latent voxel widths of $2.5\,\text{cm}$ and $20\,\text{cm}$.

4 UniGaussianGPT

4.1 Overview

UniGaussianGPT learns to recognize and generate scene content through a shared causal representation. The model produces Gaussian geometry and appearance together with spatially aligned category predictions. The central design is to associate each semantic decision with the feature token describing the same region, while learning both category recognition and scene continuation from its contextual state. The same representation can therefore identify the contents of an encoded scene or generate new content together with its categories.

Figure 2 summarizes the method. We first establish the correspondence between feature tokens, semantic states, and decoded Gaussians in Sec. 4.2. These associations describe the content retained by the scene codes, whose training targets and attribute-preserving objective are introduced in Sec. 4.3. The tokenizer is trained before the transformer and held fixed during joint semantic and next-token learning.

4.2 Learning semantics from causal scene context

The latent grid partitions space, not objects. A cell may contain only part of an object, so category recognition can require evidence beyond its local feature code. A classifier applied only to a discrete code cannot distinguish occurrences of that code in different contexts. The autoregressive state provides a richer basis for this decision by integrating a cell's feature token with preceding scene content. The placement of the semantic readout determines whether it can use both signals.

Aligning recognition with realized content. After consuming p_i, the transformer predicts a distribution over f_i, but has not yet observed which feature will represent the cell. A semantic prediction at this point would condition on the location and preceding scene without the current feature value. We instead use the state after f_i has been consumed. Writing $h_i = \mathbf{H}_{2i}$, the category distribution is

$$q_\phi(y_i = c \mid h_i) = \left[\hat{\mathbf{y}}_i\right]_c, \qquad \hat{\mathbf{y}}_i = \text{softmax}\left(g_\phi(h_i)\right), \qquad h_i = \mathbf{H}_{2i}(x_{\le 2i}), \tag{3}$$

where g_ϕ is a two-layer MLP with a SiLU hidden activation. The readout thus interprets the reference feature during training and the sampled feature during generation. In both cases, it has access to the current position, the realized local code, and the preceding spatial context. Later cells remain inaccessible under causal attention.

Coupling category recognition and scene continuation. The state h_i also predicts the next scene token. Supervising its semantic interpretation therefore trains the same representation that determines how the scene continues. With θ denoting the shared transformer and token predictor, the model defines

$$p_{\theta,\phi}(\mathbf{x}, \mathbf{y}) = p_\theta(\mathbf{x}) \prod_{i=1}^{N} q_\phi\big(y_i \mid h_i(\mathbf{x}_{\leq 2i})\big). \tag{4}$$

Here $\mathbf{y} = (y_1, \ldots, y_N)$ contains the voxel categories. Categories are predicted alongside the scene tokens, without being inserted into the sequence or fed back as conditioning inputs. Recognition and continuation are coupled through the learned states rather than through sampled category labels.

Each labeled cell receives a target y_i by majority voting over its input Gaussian categories. For the set of supervised cells $\mathcal{I}$, we optimize

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \underbrace{\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \text{CE}(\hat{\mathbf{y}}_i, y_i)}_{\mathcal{L}_{\text{sem}}}. \tag{5}$$

Cells whose target falls in an excluded category are left out of $\mathcal{I}$, and excluded categories are masked during category selection. Semantic gradients update the contextual transformer as well as the readout, while the next-token objective continues to train scene synthesis. The frozen-GPT variant in Sec. 5.5 examines this joint adaptation.

Transferring semantics to explicit geometry. Categories are predicted per latent cell, while the decoder reconstructs finer Gaussians within it. Each decoded Gaussian j takes the prediction of the latent cell $v(\boldsymbol{\mu}_j)$ containing its final center, restricted to the allowed categories $\mathcal{C}$:

$$\hat{y}_j = \arg\max_{c \in \mathcal{C}} \left[\hat{\mathbf{y}}_{v(\boldsymbol{\mu}_j)}\right]_c. \tag{6}$$

Categories are thus attached to the primitives used for rendering, including content occluded in a given view, and are assigned by where a Gaussian is decoded rather than by its index. Semantic resolution remains that of the latent grid; the decoder supplies finer geometry and appearance.

4.3 Preserving Gaussian content through tokenization

The tokenizer determines which Gaussian content can be reconstructed from the codes modeled by the transformer. While rendered supervision alone does not resolve this ambiguity for hidden surfaces, whose contribution is attenuated by transmittance through foreground content. The reconstruction analysis in Sec. 5.4 demonstrates this distinction between recovered locations and retained opacity. We therefore supervise attributes directly in 3D, using targets that supply both object and room surfaces.

Composed Gaussian targets. We convert complete 3D-FUTURE furniture meshes (Fu et al., 2021b) into Gaussian assets and prune them in isolation (Hanson et al., 2025), then place them according to the source 3D-FRONT layouts (Fu et al., 2021a). With $\mathcal{A}_k$ denoting asset k, $\mathcal{T}_k$ its placement, and $\mathcal{B}$ a room background fitted with separable furniture removed, the scene is

$$\mathcal{G} = \mathcal{B} \cup \bigcup_{k=1}^{K} \mathcal{T}_k(\mathcal{A}_k). \tag{7}$$

Construction before assembly supplies surfaces that become occluded in the furnished room. Furniture categories follow source annotations, while background labels are estimated. The assembled scene is encoded as one sequence without source asset IDs or object-wise token partitions.

Matching attributes without primitive correspondences. Compression can change the number and locations of primitives, so input and reconstruction cannot be compared index-wise. We instead match them through their shared base-voxel grid. Let $u(\hat{g})$ denote the matching voxel of decoded

Gaussian $\hat{g}$, n_u the number of input Gaussians in voxel u before encoder subsampling, and $\bar{\mathbf{a}}_u$ their mean attributes. We minimize

$$\mathcal{L}_{\text{attr}} = \frac{1}{3}\sum_{k=1}^{3} \frac{\sum_{\hat{g}} n_{u(\hat{g})} \left|\mathbf{a}_k(\hat{g}) - \bar{\mathbf{a}}_{k,u(\hat{g})}\right|}{\sum_{\hat{g}} n_{u(\hat{g})}}, \qquad (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3) = (\alpha, \log \mathbf{s}/r, \mathbf{c}), \tag{8}$$

where $\mathbf{s}$ is Gaussian scale. The three groups determine whether a primitive contributes when exposed, how far it extends, and how it appears. The sums include decoded Gaussians whose matching voxels have input support, and weighting by n_u accounts for how many input primitives each target summarizes, without assuming one-to-one correspondence. Within each group, $|\cdot|$ averages absolute differences over channels, so that the channel count does not set the group's relative weight. Opacity is compared after the sigmoid, avoiding large logit differences near zero and one, and its target is the supplied value rather than a constant that would drive every primitive toward full opacity. Log-scale differences capture multiplicative changes in extent, with $r = 3$ balancing their magnitude against opacity and color. The tokenizer objective extends Eq. 2 as

$$\mathcal{L}_{\text{tok}} = \mathcal{L}_{\text{AE}} + \lambda_{\text{attr}}\mathcal{L}_{\text{attr}}. \tag{9}$$

While the inherited terms constrain spatial support and observed appearance, $\mathcal{L}_{\text{attr}}$ constrains the attributes of supplied hidden content, helping preserve it rather than synthesizing surfaces absent from the targets.

5 Experiments

We evaluate UniGaussianGPT along three axes: (i) semantic understanding of held-out scenes, (ii) unconditional scene generation and completion, and (iii) retention of occluded Gaussian content through tokenization. Ablations further examine how joint semantic learning affects both generation and understanding.

5.1 Experimental setup

Datasets. We train and evaluate on our composed 3D-FRONT scenes, following the house-level train and validation split of GaussianGPT (von Lützow et al., 2026). The tokenizer and transformer are initialized from released GaussianGPT weights. We first fine-tune the tokenizer on the composed scenes, then train the transformer and semantic head on its tokens.

Baselines. For semantic understanding, we compare with SceneSplat (Li et al., 2025b), which predicts categories directly on 3D Gaussians. 3DGS-GBP (Cheng et al., 2024; Joseph et al., 2025), which lifts LSeg features onto Gaussians by gradient-weighted back-projection. The feed-forward models IGGT (Li et al., 2025a) and SegVGGT (Qu et al., 2026); and the 2D segmentation model LSeg (Li et al., 2022). For a fair comparison, all semantic baselines are fine-tuned on the training split of our composed 3D-FRONT scenes with our category vocabulary before evaluation. For generation and completion, we compare with WorldGrow (Li et al., 2026) and GaussianGPT (von Lützow et al., 2026). For tokenization, we compare with the original GaussianGPT's VAE which has not been supervised by attribute loss.

Metrics. Semantic understanding is evaluated on 480 $4\,\text{m} \times 4\,\text{m} \times 4\,\text{m}$ held-out chunks. We report mIoU and mean class accuracy (mAcc) in 3D, where Gaussian predictions are majority-voted within $0.2\,\text{m}$ latent cells and compared with reference cell labels, and in 2D, where predicted and reference labels are rendered from shared views. Foreground (fg) variants exclude floor, wall, and ceiling. The *Residual* category, which groups content without a clear semantic class, is excluded from all semantic metrics (Appendix Table 7). For generation, appearance FID (Heusel et al., 2017) and KID (Bińkowski et al., 2018) are computed on horizontal orbit renderings, layout FID and KID on top-down depth maps of the bottom 60% of Gaussians, and geometry coverage (COV) and minimum matching distance (MMD) on 16k points under Chamfer distance. For completion, CLIP similarity (Radford et al., 2021) measures agreement between each completion and its reference chunk.

Table 1: **Semantic understanding.** 480 held-out chunks; all baselines are fine-tuned on our data. 3D metrics use $0.2\,\mathrm{m}$ latent cells and 2D metrics shared views; fg excludes floor, wall, and ceiling. **Blue**: best; green: our method; –: no 3D prediction.

Method	3D (latent cells)				2D (rendered views)			
	mIoU↑	mAcc↑	fg-mIoU↑	fg-mAcc↑	mIoU↑	mAcc↑	fg-mIoU↑	fg-mAcc↑
Image-based methods								
LSeg	–	–	–	–	0.406	0.451	0.312	0.358
SegVGGT	0.331	0.407	0.262	0.335	0.394	0.464	0.311	0.386
IGGT	0.343	0.404	0.238	0.299	0.435	0.476	0.334	0.382
3DGS-GBP w/ LSeg	0.348	0.411	0.242	0.306	0.441	0.488	0.342	0.396
Gaussian-based methods								
SceneSplat	0.385	0.425	0.278	0.321	0.387	0.427	0.280	0.323
UniGaussianGPT	0.578	0.731	0.509	0.685	0.547	0.719	0.476	0.676

Table 2: **Generation and completion.** Held-out scenes. Appearance: orbit renderings; layout: top-down depth maps of the bottom 60% of Gaussians; geometry: Chamfer COV/MMD on 16k points. CLIP applies only to completion. **Blue**: best within each setting.

Method	Appearance		Geometry		Layout		
	FID↓	KID↓	COV↑	MMD↓	FID↓	KID↓	CLIP↑
Unconditional generation							
WorldGrow	84.78	0.086	0.390	0.085	97.08	0.076	–
GaussianGPT	89.64	0.091	0.374	0.076	104.03	0.085	–
UniGaussianGPT	70.67	0.071	0.429	0.072	68.36	0.036	–
Completion, 25% context							
WorldGrow	–	–	–	–	–	–	–
GaussianGPT	79.36	0.063	0.970	0.056	129.40	0.064	0.887
UniGaussianGPT	73.84	0.056	0.940	0.065	101.73	0.029	0.892
Completion, 50% context							
WorldGrow	–	–	–	–	–	–	–
GaussianGPT	83.81	0.069	0.865	0.068	133.63	0.071	0.877
UniGaussianGPT	73.76	0.056	0.965	0.053	99.58	0.029	0.900

5.2 Semantic understanding

Table 1 reports semantic prediction on held-out scenes, evaluated both in 3D and in rendered views, with examples in the first row of Figure 3. UniGaussianGPT achieves the best result on all eight metrics.

3D understanding. UniGaussianGPT reaches a 3D mIoU of 0.578 and a 3D mAcc of 0.731, improving over the strongest baseline, SceneSplat, by 0.193 and 0.306. The margin is larger on mAcc than on mIoU, here and in 2D, showing that the baselines miss a considerably larger fraction of each category, which the class-averaged accuracy exposes directly. The advantage persists on foreground categories. 3D fg-mIoU reaches 0.509, 0.231 above SceneSplat, and 3D fg-mAcc reaches 0.685, compared with 0.335 for the strongest baseline on this metric. Excluding floor, wall, and ceiling lowers mIoU by 0.069 for UniGaussianGPT but by 0.107 for SceneSplat, so the improvement is not carried by large structural classes. These results are obtained with less context than the baselines receive: UniGaussianGPT labels each reference sequence in a single teacher-forced pass, where causal attention exposes each cell only to its own feature token and preceding cells, whereas the baselines observe the full window.

2D understanding. Rendered views reorder the baselines. Methods that predict from images, namely SegVGGT, IGGT, and 3DGS-GBP, all score higher in 2D than in 3D, and 3DGS-GBP becomes the strongest baseline on every 2D metric. SceneSplat, which operates directly on Gaussians, leads the baselines in 3D but has the lowest 2D mIoU (0.387). No baseline is strongest in both domains. UniGaussianGPT leads in both, with a 2D mIoU of 0.547 and a 2D fg-mIoU of 0.476,

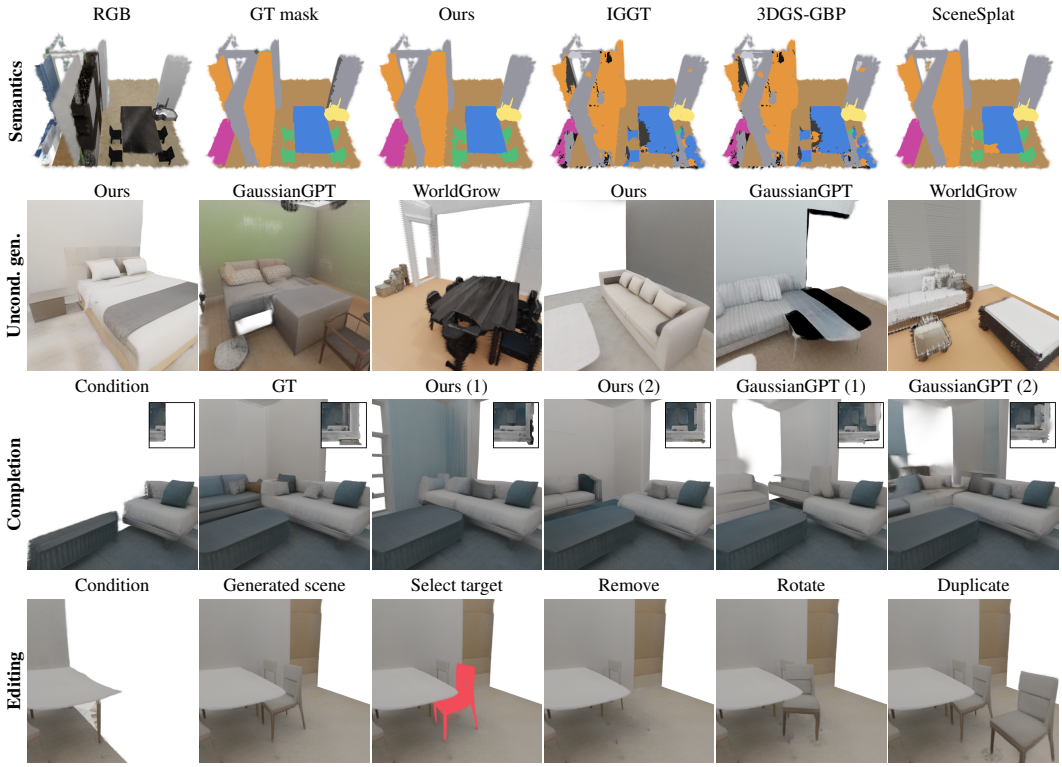

Figure 3: Qualitative results. **Semantics:** predicted categories on a held-out scene, compared with fine-tuned baselines. **Uncond. gen.:** two unconditional samples per method. **Completion:** two completions per method from the same partial condition. **Editing:** a scene generated from the condition is decomposed by its predicted categories; the selected object is removed, rotated, or duplicated while the rest of the scene is unchanged.

exceeding 3DGS-GBP by 0.106 and 0.134. Because each category is assigned to the Gaussians that are rendered (Sec. 4.2), rendered labels follow directly from the 3D prediction rather than from a separate image-space model.

5.3 Scene Generation and Completion

Table 2 reports unconditional generation and completion from 25% and 50% spatial context. Qualitative results are shown in Figure 3.

Unconditional generation. UniGaussianGPT achieves the best results on all appearance, geometry, and layout metrics. The largest gain is in layout quality, where layout FID drops to 68.36, compared with 97.08 for WorldGrow and 104.03 for GaussianGPT. Appearance FID likewise improves from 84.78 and 89.64 to 70.67. The contribution of semantic learning to this prior is isolated in Sec. 5.5.

Scene completion. The same next-token process completes partial scenes and again achieves the best appearance, layout, and CLIP scores in both settings. With 25% and 50% context, layout FID reaches 101.73 and 99.58, compared with 129.40 and 133.63 for GaussianGPT, and layout KID is more than halved. Geometry is slightly behind GaussianGPT at 25% context and clearly better at 50%. The layout advantage therefore carries over from unconditional to conditional generation.

5.4 Content Retention through Tokenization

We evaluate whether tokenization preserves the content of composed Gaussian scenes, including surfaces occluded in furnished views. Table 3 shows improved reconstruction and recall for both visible

Table 3: **Content retention through tokenization.** Recall: % of hidden/visible reference Gaussians with a decoded neighbor within $5\,\mathrm{cm}$ and opacity > 0.15. Δ_α: hidden recall lost to the opacity threshold.

	Reconstruction			Retention		
Tokenizer	PSNR↑	SSIM↑	LPIPS↓	Hid.↑	Vis.↑	Δ_α ↓
GaussianGPT	18.29	0.7819	0.4069	83.32	84.14	9.95
Ours	26.01	0.8685	0.3262	92.96	93.38	0.38

Table 4: **Effect of joint semantic learning.** Each row changes one component of the full model. CE: next-token cross-entropy; semantics: 3D metrics on held-out chunks.

	CE↓		Semantics↑	
Variant	Train	Val	mIoU	mAcc
Full model	2.122	2.213	0.578	0.731
w/o sem. head	2.113	2.221	–	–
frozen GPT	2.870	2.854	0.342	0.561
Lovász	2.294	2.317	0.584	0.722

and hidden Gaussians. Hidden recall rises from 83.32% to 92.96%, while the opacity gap falls from 9.95 to 0.38 percentage points. This suggests that direct attribute supervision retains occluded content with sufficient opacity, rather than merely reconstructing low-opacity Gaussians near the correct locations. Together with predicted categories and spatial connectivity, this retained content supports the selection and transformation of components in generated scenes, including removal, rotation, and duplication (Figure 3). In particular, preserving surfaces hidden in the assembled scene helps maintain scene content when an object is removed or transformed.

5.5 Ablation study

We ablate joint semantic learning in Table 4, which reports next-token cross-entropy (CE) on training and validation sequences as a measure of the generative prior, together with 3D mIoU and mAcc for semantic understanding. *W/o semantic head* removes the semantic objective and trains the transformer with next-token prediction alone under the same setting. *Frozen GPT* keeps the transformer fixed and trains only the semantic head. *Lovász* replaces the cross-entropy in $\mathcal{L}_{\mathrm{sem}}$ (Eq. 5) with the Lovász-Softmax loss (Berman et al., 2018), a surrogate that directly optimizes per-class IoU. All other settings are kept unchanged.

Table 4 examines how semantic supervision affects the shared states used for both scene continuation and category prediction. Removing the semantic head slightly improves training CE (2.122 to 2.113), but increases validation CE (2.213 to 2.221). Although the validation difference is small, it suggests that semantic supervision need not compete with next-token prediction and may provide a benefit beyond the training sequences. The semantic head also needs an adaptable backbone: freezing the GPT raises training and validation CE to 2.870 and 2.854, while reducing 3D mIoU and mAcc from 0.578 and 0.731 to 0.342 and 0.561. A trained readout on fixed contextual states therefore cannot recover the performance of joint training. Finally, Lovász-Softmax achieves a slightly higher mIoU (0.584 versus 0.578), but lowers mAcc and increases training and validation CE by 0.172 and 0.104. Its small mIoU gain comes with a clearer cost to scene continuation. We therefore adopt the per-cell semantic cross-entropy in Eq. 5 as the better overall choice for this model.

6 Conclusion

We presented UniGaussianGPT, an autoregressive framework that learns semantic recognition and 3D Gaussian scene generation in a shared representation. The same contextual states predict categories for the content already represented and model how the scene continues. Composed furniture and room backgrounds provide labeled Gaussian targets, and direct attribute supervision helps preserve their occluded content through tokenization. Experiments show state-of-the-art performance in both semantic understanding and scene generation and completion, together with better opacity-aware retention of hidden content. The generated representation also supports scene decomposition and basic component operations. These results connect autoregressive scene synthesis with explicit semantic prediction, suggesting a direction towards interactive world modelling and future robot learning from generated, semantically labeled environments.

AI USE STATEMENT

In this work, we used generative AI tools for **Drafting and editing:** language models drafted and revised the text, including the organization of the argument and the wording of claims, under the authors' direction.

REFERENCES

Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4413–4421. IEEE, 2018.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

Aleksey Bokhovkin, Quan Meng, Shubham Tulsiani, and Angela Dai. Scenefactor: Factored latent 3d diffusion for controllable 3d scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 628–639, 2025.

Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's lgs: An efficient approach for language gaussian splatting. *arXiv preprint arXiv:2412.01807*, 2024.

Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. *Advances in neural information processing systems*, 37:40212–40229, 2024.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10913–10922. IEEE, 2021a.

Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129 (12):3313–3337, 2021b.

Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5354–5363. IEEE, 2024.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5949–5958. IEEE, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23646–23657. IEEE, 2025.

Joji Joseph, Bharadwaj Amrutur, and Shalabh Bhatnagar. Gradient-weighted feature back-projection: A fast alternative to feature distillation in 3d gaussian splatting. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

Hao Li, Zhengyu Zou, Fangfu Liu, Xuanyang Zhang, Fangzhou Hong, Yukang Cao, Yushi Lan, Manyuan Zhang, Gang Yu, Dingwen Zhang, et al. Iggt: Instance-grounded geometry transformer for semantic 3d reconstruction. *arXiv preprint arXiv:2510.22706*, 2025a.

Sikuang Li, Chen Yang, Jiemin Fang, Taoran Yi, Jia Lu, Jiazhong Cen, Lingxi Xie, Wei Shen, and Qi Tian. Worldgrow: Generating infinite 3d world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6433–6441, 2026.

Yue Li, Qi Ma, Runyi Yang, Huapeng Li, Mengjiao Ma, Bin Ren, Nikola Popovic, Nicu Sebe, Ender Konukoglu, Theo Gevers, et al. Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4961–4972. IEEE, 2025b.

Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Ping Luo, and Yang Cao. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5726–5738, 2026.

Quan Meng, Lei Li, Matthias Nießner, and Angela Dai. Lt3sd: Latent trees for 3d scene diffusion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 650–660. IEEE, 2025.

Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in neural information processing systems*, 34:12013–12026, 2021.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20051–20060. IEEE, 2024.

Minghan Qin, Yuang Wang, Xiuyu Yang, Yushi Long, Yujian Zhang, Ruihuan Wang, Kai Ye, Yangang Zhang, and Hang Li. Lucida: Parse, generate, and place for composable real-to-sim scene modeling. *arXiv preprint arXiv:2608.30821*, 2026.

Jinyuan Qu, Hongyang Li, and Lei Zhang. Segvggt: Joint 3d reconstruction and instance segmentation from multi-view images. *arXiv preprint arXiv:2603.19926*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21783–21794. IEEE, 2024.

Barbara Roessle, Norman Müller, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, Angela Dai, and Matthias Nießner. L3dg: Latent 3d gaussian diffusion. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20507–20518. IEEE, 2024.

Nicolas von Lützow, Barbara Rössle, Katharina Schmid, and Matthias Nießner. Gaussiangpt: Towards autoregressive 3d gaussian scene generation. *arXiv preprint arXiv:2603.26661*, 2026.

Joanna Waczyńska, Piotr Borycki, Sławomir Tadeja, Jacek Tabor, and Przemysław Spurek. Games: Mesh-based adapting and modification of gaussian splatting. *Pattern Recognition*, pp. 114561, 2026.

Yuxin Wang, Qianyi Wu, Guofeng Zhang, and Dan Xu. Learning 3d geometry and feature consistent gaussian splatting for object removal. In *European conference on computer vision*, pp. 1–17. Springer, 2024.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851. IEEE, 2024a.

Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024b.

Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, et al. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. *ACM Transactions on Graphics (ToG)*, 43(4):1–17, 2024c.

Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. *ACM Transactions on Graphics (TOG)*, 44(4):1–19, 2025.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European conference on computer vision*, pp. 162–179. Springer, 2024.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *International Conference on Learning Representations*, volume 2024, pp. 765–783, 2024.

Chaoyi Zhou, Run Wang, Feng Luo, Mert D Pesé, Zhiwen Fan, Yiqi Zhong, and Siyu Huang. Ff3r: Feedforward feature 3d reconstruction from unconstrained views. *arXiv preprint arXiv:2604.09862*, 2026.

A Data construction

GaussianGPT fits one set of Gaussians to each furnished room, so every surface that is hidden in the furnished room (the floor under a sofa, the back of a wardrobe) is either missing or filled with whatever the optimizer found cheapest, and no Gaussian knows which object it belongs to. We instead build each scene from two separately represented parts: the empty room, fitted to renders of the room with its furniture removed, and every piece of furniture, converted from its complete mesh. Composing the two with the 3D-FRONT layout gives scenes in which hidden surfaces exist as Gaussians and every Gaussian carries a category and an instance index. Figure 4 summarizes the steps; this section gives their parameters.

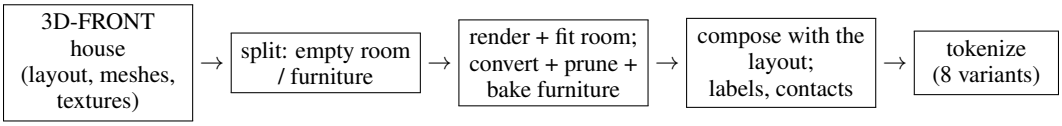

Figure 4: Per-house data construction.

Splitting furniture from the room. An instance is *extracted* as furniture if its footprint is at least $0.15\,\mathrm{m}^2$ or its height at least $0.6\,\mathrm{m}$; railings and build elements are never extracted. Lamps with emissive materials are kept in the empty scene as light sources but made invisible to the camera, and their geometry is composed back as Gaussians like other furniture. Layout transforms are taken from Blender's world matrices and include reflections and anisotropic scale (3.5% of 3D-FRONT instances have a negative scale).

Rendering. All renders use BlenderProc with Cycles at 960×960 pixels and 16 samples per pixel, with a black world so that the room is lit only by the 3D-FRONT ceiling emitters and lamps. Cameras follow a walk through every room at standing height: the camera looks at a point 3 m ahead on the path, orbits each piece of furniture from four directions, and receives 150 additional poses chosen to cover surfaces the walk misses. The empty room is rendered with 175 views per $100\,\mathrm{m}^2$ of floor (at least 120, at most 600).

Empty-room fitting. The empty room is fitted on the tokenizer's 2.5 cm grid with at most one Gaussian per voxel. A Gaussian's center is its voxel anchor plus an offset $1.5\, v \tanh(\cdot)$ and its scale is $2v\, \sigma(\cdot)$, with $v = 2.5$ cm, so every Gaussian stays representable by the tokenizer (extent at most 5 cm). The fit is initialized from Gaussians sampled on the room mesh with the mesh albedo as color and runs for 35,000 iterations, with pruning until iteration 17,500 and opacity resets until iteration 8,750; 68 views are held out. Gaussians initialized from the mesh are never pruned, so walls and floors stay closed even where they are rarely seen. Structural Gaussians have a minimum scale of 1.8 cm (applied to structure only, since a global minimum inflates small objects). Gaussians observed in fewer than 12 views are handled explicitly: their colors are frozen if observed in fewer than 8 views, higher-order color is frozen below 32 observations and clamped to 0.3, and structural colors are set to the mesh albedo times a smooth per-object gain field (kernel regression with $\sigma = 0.25$ m, weighted by observation count). Color gradients are applied only to Gaussians at the first surface along each ray (depth within two voxels).

Furniture conversion. Each 3D-FUTURE mesh used by a layout is converted once into Gaussians without optimization. Points are sampled on each triangle in proportion to its area, at 2×140^2 samples per m^2. Each Gaussian is a flat disc in the triangle plane, as in SuGaR (Guédon & Lepetit, 2024) and GaMeS (Waczyńska et al., 2026): its rotation is the triangle frame, its in-plane extents follow the local sample spacing with an overlap factor of 2.0, its aspect ratio is at most 4, its extent at most 2 cm, and its opacity 0.99. Because the mesh is complete, the asset includes its back and underside. This gives 11,034 assets with 3.70×10^9 Gaussians in total.

Furniture colors. Colors are transferred per placed instance from the furnished renders: every Gaussian is projected into the renders and accepted where its depth agrees with the rendered depth, inside an eroded instance mask; accepted observations are averaged. Gaussians that no camera observes take their texture color scaled by the gain of observed neighbors.

Furniture pruning. Converted assets are about ten times denser than fitted rooms. Each asset is pruned with PUP 3D-GS (Fisher-information scores, two rounds, each followed by 5,000 fine-tuning steps on renders of the isolated asset from an orbit, degree-0 color, white background). We change two things. First, pruning is stratified over 2.5 cm voxels: Gaussians are ranked within their voxel and survivors are taken from every voxel in turn, instead of by a global score, which would concentrate survivors on silhouettes and hollow out flat regions. Second, scales are clamped to 5 cm after every fine-tuning step, the tokenizer's limit. Scores depend on color, so an asset is pruned in a lit version (the baked colors of its instance of median brightness), and every surviving Gaussian keeps the index of its source Gaussian, so each placed instance recovers its own baked colors. Pruning reduces the library from 3.70×10^9 to 2.71×10^8 Gaussians (7.3%; median 2.33 Gaussians per occupied voxel; none above 5 cm). On 36 test assets, the stratified, clamped variant reaches 37.3 dB PSNR against 32.8 dB for unmodified PUP and 20.6 dB for random pruning to the same count.

Composition and contacts. Each asset is placed with its layout transform; covariances are transformed exactly, and reflected transforms are handled through the SVD of $A\, R\, \mathrm{diag}(s)$. Where furniture meets the room, three fixes remove artifacts of fitted floor Gaussians: floor Gaussians within 10 cm of furniture are replaced by flat up-facing discs (1.5 cm) colored from the undisturbed floor around them; room Gaussians within 3 cm of furniture are removed; and room Gaussians within 10 cm have their extent capped at 2 cm. When the tokenizer keeps one Gaussian per voxel, it keeps the most opaque one; in voxels shared by furniture and room, furniture is given priority by raising its opacity logit just above the room's. Furniture Gaussians take the category and instance of their source object; room Gaussians take the category of the room-mesh element they were initialized from. Instances that 3D-FRONT stores inside the room mesh cannot be removed and keep the `structure` label. Table 5 summarizes the resulting data.

B Implementation details

Table 6 lists the full configuration of the reported models.

Table 5: Statistics of our composed 3D-FRONT scenes.

Quantity	Value
Held-out chunks for semantic evaluation	480
Converted furniture assets	11,034
Furniture Gaussians before / after pruning	3.70×10^9 / 2.71×10^8
Token variants per house	8

Tokenizer. The tokenizer keeps GaussianGPT's architecture and is fine-tuned from the released 3D-FRONT weights on our composed 3D-FRONT scenes. We use the official GaussianGPT's losses, and our own attribute loss $\mathcal{L}_{\text{attr}}$ with weight 3.0 on 2.5 cm cells.

Transformer and category head. The transformer is initialized from the released GaussianGPT weights and fine-tuned on tokens of the fine-tuned tokenizer. Each latent voxel is a position token followed by a feature token; the position vocabulary enumerates the 20^3 cells of a chunk, and sampling enforces increasing positions. The category head is a two-layer MLP ($1024 \to 512 \to 23$, SiLU) on the final hidden state of each feature token, trained jointly with next-token prediction.

Table 6: Configuration of the reported models.

	Item	Value
Tokenizer	Base / latent voxel	2.5 cm / 20 cm (three $2\times$ stages); one Gaussian per base voxel
	Encoder / decoder	sparse 3D CNN (MinkowskiEngine), embedding dim. 64, at most 512 channels
	Quantization	LFQ, 4,096 codes, one token per latent voxel
	Gaussian attributes	offset, opacity, scale, rotation, base color (degree 0)
	Optimizer	AdamW, lr 10^{-4}, no weight decay, cosine to $0.1\times$, no warm-up, bf16
	Batch	1 scene per GPU, 2 GPUs, gradient accumulation 4 (8 scenes per step)
Transformer	Architecture	24 layers, width 1024, 16 heads, 3D RoPE on latent coordinates
	Context	16,384 tokens; chunk of 20^3 latent voxels (4 m), x-major order
	Vocabulary	8,000 position + 4,096 feature tokens + special tokens
	Optimizer	Muon for matrices (lr 0.02), AdamW $(\beta_1, \beta_2) = (0.8, 0.95)$ for embeddings (0.2), output layer (0.004) and scalars (0.5), AdamW rates scaled by $(1024/768)^{-1/2}$
	Weight decay	$0.2 \cdot (12/24)^2 = 0.05$ on Muon groups, decayed linearly to 0
	Schedule	cosine to $0.1\times$, no warm-up, bf16
	Category head	MLP $1024 \to 512 \to 23$ (SiLU), lr 10^{-3}, $\lambda = 1$
	Batch	4 chunks per GPU, 4 GPUs, accumulation 4 (64 chunks per step)
	Augmentation	4 rotations $\times$ reflection of x; chunk locations with $\geq$30% occupancy

C SEMANTIC CLASSES

We fold the 99 categories of 3D-FRONT into 23 classes. Table 7 lists the 3D-FRONT categories each class contains.

Table 7: Semantic classes and the 3D-FRONT categories folded into each. Colors are those of every mask in the paper.

Class	3D-FRONT categories
Structure	
`floor`	Floor, SlabBottom
`ceiling`	Ceiling, SmartCustomizedCeiling, CustomizedCeiling, ExtrusionCustomizedCeilingModel, SlabTop
`wall`	WallInner, WallOuter, WallTop, WallBottom, Baseboard, Pocket, CustomizedFeatureWall, CustomizedBackgroundModel, ExtrusionCustomizedBackgroundWall, Customized_wainscot
`door`	Door
`window`	Window, BayWindow
`stair`	Stair
Furniture and fixtures	
`sofa`	sofa, two-seat sofa, three-seat / multi-person sofa, L-shaped sofa, chaise longue sofa, lazy sofa
`chair`	chair, dining chair, armchair, lounge chair / book-chair / computer chair, dressing chair, classic Chinese chair
`stool`	footstool / sofastool / bed end stool / stool, pier / stool, barstool
`bed`	bed, double bed, single bed, kids bed, bunk bed, bed frame
`table`	table, dining table, tea table, round end table, corner / side table, dressing table
`desk`	desk
`cabinet`	cabinet, kitchen cabinet, children cabinet, storage unit, media unit, TV stand, nightstand, drawer chest / corner cabinet, sideboard / side cabinet / console, cabinet / shelf / desk, cabinet / lightband
`shelf`	shelf, bookcase / jewelry armoire
`wardrobe`	wardrobe
`lamp`	ceiling lamp, pendant lamp, lighting, lightband
`appliance`	appliance, electric, electronics, wine cooler
`sanitary`	basin, bath
`decor`	plants, art, accessory, mirror
`furniture_other`	customized furniture, customized fixed furniture, outdoor furniture
Residual	
`void`	no category
`other`	others, recreation, CustomizedPersonalizedModel, 200 – on the floor, 300 – on top of others, 400 – attach to wall, 500 – attach to ceiling
`structure`	Beam, Column, Flue, Cornice, SewerPipe, Hole, Build Element, SlabSide, CustomizedPlatform, Front, Back

D ADDITIONAL QUALITATIVE RESULTS

Figure 5 shows predicted categories on eight held-out chunks, and Figure 6 shows scene completions from spatial context.

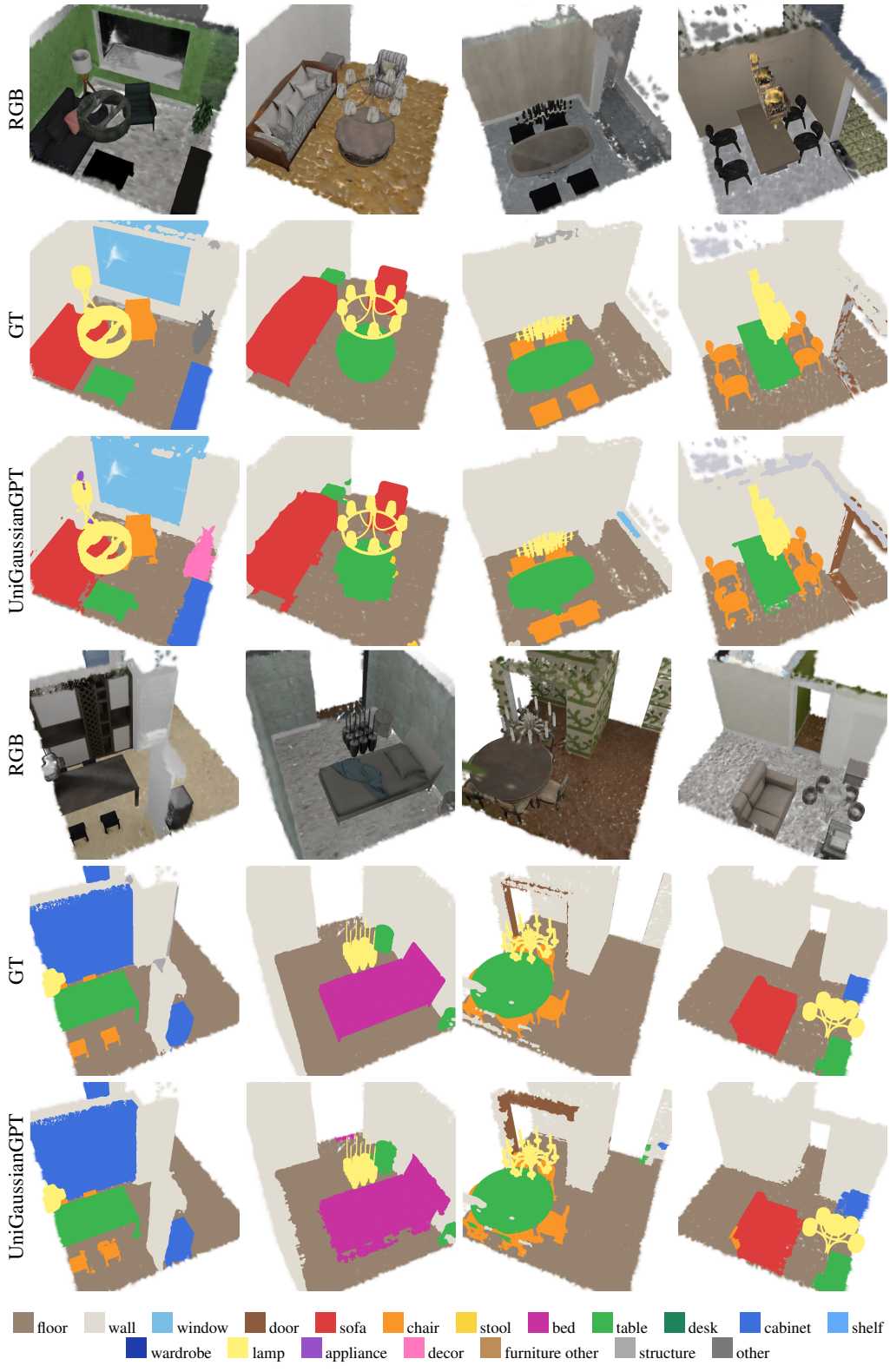

Figure 5: Categories of held-out chunks: RGB, ground truth and UniGaussianGPT (teacher forcing) on the same Gaussians and camera, ceiling removed. Each column is one case: RGB (top), ground truth (middle) and UniGaussianGPT (bottom).

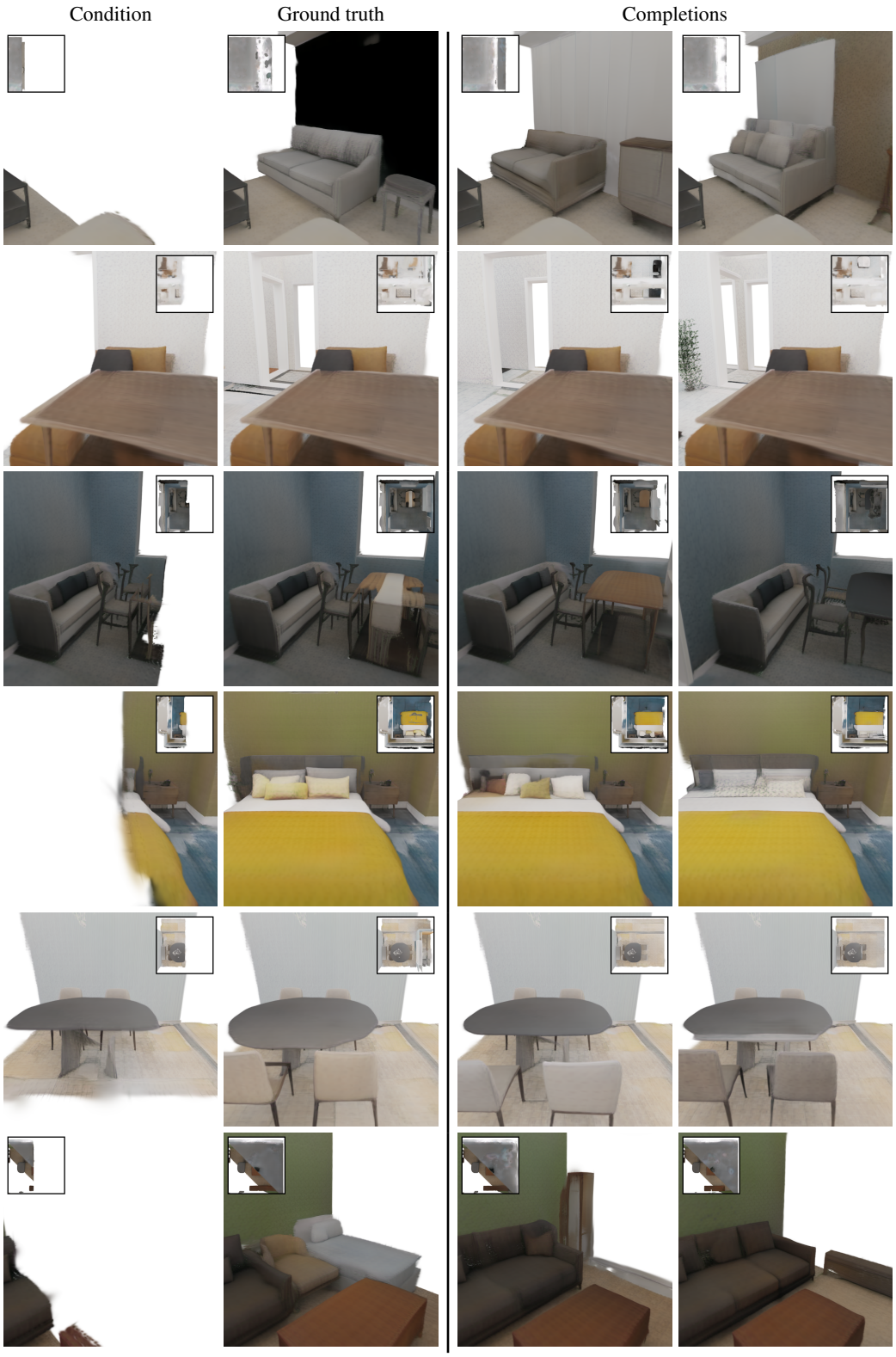

Figure 6: Scene completion from spatial context: the condition, the ground truth and two completions by UniGaussianGPT (selected examples). Insets: top-down views with the ceiling removed.