Scenes as Objects, Not Primitives: Instance-Structured 3D Tokenization from Unposed Views

Mijin Yoo[1*] **In Cho**[1*] **Subin Jeon**[2] **Jiwoo Lee**[1] **Eunbyung Park**[1] **Seon Joo Kim**[1†]

[1]Yonsei University [2]Seoul National University

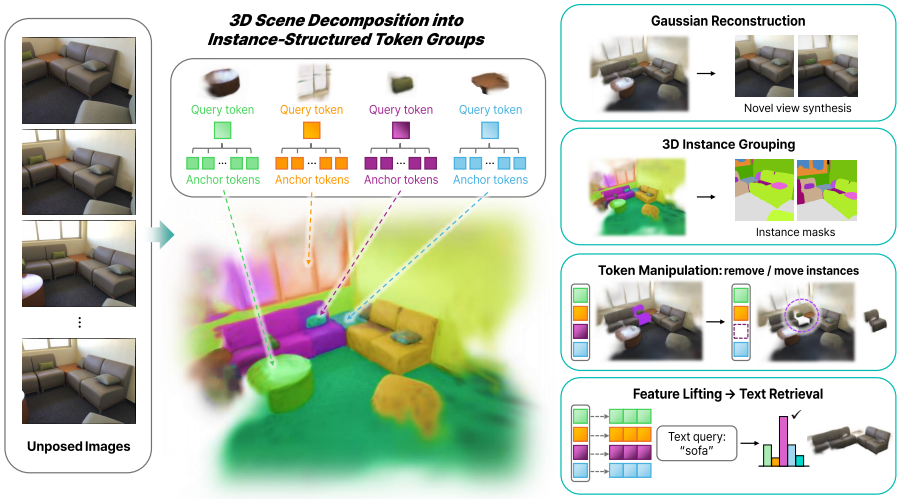


Figure 1: Our model maps unposed multi-view images to instance-structured 3D token groups, which make instances a native interface of the 3D representation. The token groups support novel-view synthesis, 3D instance segmentation, instance-level manipulations and open-vocabulary retrieval.

Abstract

A 3D scene is understood through its objects, not the primitives that compose them. Yet feed-forward reconstruction methods output dense, unstructured sets of points or Gaussians, leaving object-level structure to be recovered after the fact. We propose a feed-forward framework that decomposes a scene into instance-structured 3D token groups directly from unposed multi-view images — compact object-centric units from which reconstruction, segmentation, and manipulation all follow. Each token group pairs an instance token capturing entity-level identity with anchor tokens that encode local geometry and appearance, which are decoded into a set of 3D Gaussians. This two-level factorization decouples object identity from local appearance, making object instances a native interface of the representation rather than a derived product. The token groups are learned through differentiable rendering with joint reconstruction and segmentation supervision, requiring no 3D annotations. Our feed-forward model surpasses per-scene optimization baselines in class-agnostic instance segmentation while remaining competitive in novel view synthesis. Beyond these metrics, the same token groups directly unlock instance-level scene editing — removing, translating, or inserting objects by operating on their groups — as well as efficient open-vocabulary 3D instance retrieval, where retrieval complexity scales with the number of instances rather than primitives. Project page: `https://yoomimi.github.io/instok3d`

*Equal contribution.

†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

A 3D scene is not a bag of primitives. It is a composition of objects whose identities and boundaries give the scene its structure. Yet the dominant paradigm for feed-forward 3D reconstruction produces exactly that: dense, unstructured collections of points or Gaussians, with no notion of what belongs together [28, 27, 5, 6, 11, 1]. For a representation to support object-level reasoning, the entities themselves — not just the primitives that depict them — must be first-class units of the representation.

Recent feed-forward 3D reconstruction methods have made remarkable progress in predicting detailed geometry from unposed multi-view images [28, 13, 27, 30, 24, 11, 1], and a natural next step has been to enrich these reconstructions with semantics by attaching feature vectors from 2D foundation models to each primitive. While effective for local annotation, this strategy does not change the unit of representation. Object-level information remains scattered across many primitives [14, 9, 25, 1], the same semantic label is stored redundantly for every element of an entity, and any operation defined over objects — querying, editing, reasoning — still requires post-hoc grouping or aggregation [31, 35, 26, 23, 4, 18].

We view this limitation as a representation mismatch rather than a lack of expressive features. A primitive is a local geometric fragment; regardless of what feature is attached to it, it cannot supply the entity-level context needed to interpret the object it belongs to or provide a meaningful interface for interacting with it. For high-level 3D understanding, the representation should make semantic entities first-class units while preserving access to fine-grained details within each entity.

In this paper, we propose to restructure the representation itself around objects. Given unposed multi-view images, our model decomposes a scene into a compact set of instance-structured 3D token groups in a single forward pass (Figure 1). Each group pairs an instance token — which summarizes the identity and extent of an object instance — with a set of anchor tokens that encode local geometry and appearance, each decoding into a set of 3D Gaussians. The instance tokens expose object-level structure directly, while the anchor tokens preserve the fine-grained detail needed for rendering. This two-level factorization separates what belongs together from how each part looks, making object instances an explicit, manipulable interface of the scene representation.

We learn this representation with a joint reconstruction-segmentation framework built on top of a 3D foundation model. Both objectives are supervised through differentiable 2D rendering: RGB images at novel viewpoints supervise reconstruction, while rendered instance masks supervise grouping. No 3D annotations are required — the 3D instance structure emerges entirely from 2D supervision.

Once the instance structure is learned, it also provides a natural unit for integrating semantics. Rather than lifting high-dimensional features independently to every Gaussian, we distill 2D foundation model features into the token groups: each group stores a shared instance-level embedding, with lightweight anchor-level residuals capturing spatially varying detail. This yields a compact semantic representation that supports text-based retrieval at the entity level while preserving local specificity.

We evaluate on indoor scene benchmarks across reconstruction, feature lifting, and class-agnostic instance segmentation. Our feed-forward model surpasses per-scene optimized baselines in instance segmentation, while achieving competitive reconstruction quality. Beyond these results, the same token groups naturally support instance-level scene editing in 3D space — removing, translating, or inserting objects by directly operating on their token groups, as well as open-vocabulary 3D instance retrieval. These results suggest that structuring 3D representations around objects, rather than primitives, opens a more natural interface for both understanding and interacting with 3D scenes.

2 Related work

Feed-forward 3D Gaussian reconstruction. 3D Gaussian Splatting [12] represents scenes with efficiently renderable Gaussian primitives, but the original formulation requires per-scene optimization from posed images. Feed-forward methods remove this optimization by predicting Gaussians directly from input views: pixelSplat [5] and MVSplat [6] assume calibrated views and produce pixel-aligned Gaussians, while recent geometry foundation models [28, 13, 27] enable pose-free reconstruction from unposed image collections. Building on these models, NoPoSplat [30], Splatt3R [24], and AnySplat [11] extend Gaussian prediction to uncalibrated settings, and Uni3R [25] and C3G [1] adapt feed-forward Gaussian representations for scene understanding. Despite these advances, the

representation unit remains low-level: pixels, point-map elements, Gaussian queries, or individual Gaussians. Such units are well suited for rendering, but not for human-aligned scene understanding, where semantics are organized around coherent entities. Our token groups shift the primary semantic units from these primitives to entities: we introduce anchor tokens that generate local Gaussians and instance tokens that group anchors into instances.

Semantics and instances for multi-view 3D scenes. Instance-aware 3DGS methods often attach identity information directly to the Gaussian representation. Gaussian Grouping [31] optimizes per-Gaussian identity features, while ObjectGS [35] reduces alpha-blending ambiguity using one-hot object-ID channels inherited from object-aware anchors. These methods produce consistent masks, but require scene-specific reconstruction and labeling. Other methods attach semantics to an already formed 3D representation by lifting 2D masks or features [26, 29, 20, 23, 4, 18], or predict feature-augmented Gaussians in a feed-forward manner [9, 25, 1]. IGGT [15] and PanSt3R [36] further move multi-view instance or panoptic segmentation into the network itself. However, these outputs are still dense primitive-level masks or features; persistent 3D object handles require lifting, merging, or rendering after the fact. In contrast, our token groups are the scene representation itself: anchor tokens decode local Gaussians, instance tokens bind anchors into instance-level units, and the resulting groups provide shared handles for feature lifting and manipulation.

Object-centric 3D scene reconstruction. Object-centric representation learning models scenes as coherent entities rather than independent local features, as reflected in slot-based and set-prediction models [17, 3, 7, 33]. In 3D, SlotLifter [16] learns object-centric radiance fields through slot-guided decomposition, while GOCL [10] introduces object-centric supervision for per-scene optimized 3DGS using a scene-agnostic object codebook. These works highlight the value of object-level 3D structure, but the object variables are induced within radiance fields or optimized Gaussian scenes. In contrast, we learn instance-structured token groups directly from unposed multi-view images: anchor tokens generate local Gaussians, and supervised instance tokens group anchors into human-aligned object-scale units that share semantic features and serve as entity-level interfaces.

3 Method

We present instance-structured token groups, a feed-forward 3D scene representation built from unposed multi-view images. Each group pairs an instance token capturing entity-level identity with anchor tokens that encode local geometry and appearance, which are decoded into a set of 3D Gaussians. Figure 2(a) shows the model architecture: a frozen geometry foundation model extracts multi-view features and pointmaps, fused into context tokens. An image-anchor transformer ($\mathcal{D}_{\text{anchor}}$) cross-attends to these context tokens to produce anchor tokens, and an anchor-grouping transformer ($\mathcal{D}_{\text{group}}$) cross-attends to the anchor tokens to produce group tokens, which compete for anchor ownership via softmax assignment (Sec. 3.1). Figure 2(b) shows the training supervision: RGB rendering losses shape the anchor tokens while instance mask losses shape the group tokens (Sec. 3.2). Figure 2(c) shows how 2D foundation model features are distilled into the token groups, decomposed into a shared group-level embedding and low-dimensional anchor-level residuals (Sec. 3.3).

3.1 Instance-structured 3D tokenization

We tokenize unposed multi-view images into instance-structured 3D token groups: a set of anchor tokens that decode into 3D Gaussians, each assigned to one of the learned group tokens.

Multi-view feature encoding. Given V unposed RGB images $\mathcal{I} = \{I_i\}_{i=1}^{V}$, a frozen 3D foundation model [27] extracts multi-view features $F_i \in \mathbb{R}^{H' \times W' \times C}$ and pointmaps $P_i \in \mathbb{R}^{H \times W \times 3}$. We downsample each pointmap with stride p to obtain patch-aligned 3D coordinates $\tilde{P}_i \in \mathbb{R}^{H' \times W' \times 3}$. To provide additional appearance and geometry cues, we linearly patchify the RGB images and pointmaps and add them to the foundation model features. The resulting features are flattened across all views and patches into the context features $X = \{x_j\}_{j=1}^{VH'W'}$, where each $x_j \in \mathbb{R}^C$ is the fused feature at one patch of one view. X serves as the multi-view context for the token group decoder.

Token group initialization. Anchor tokens are initialized from the patch-aligned 3D coordinates $\tilde{P}$ and their corresponding context features. We apply farthest point sampling over all patch coordinates

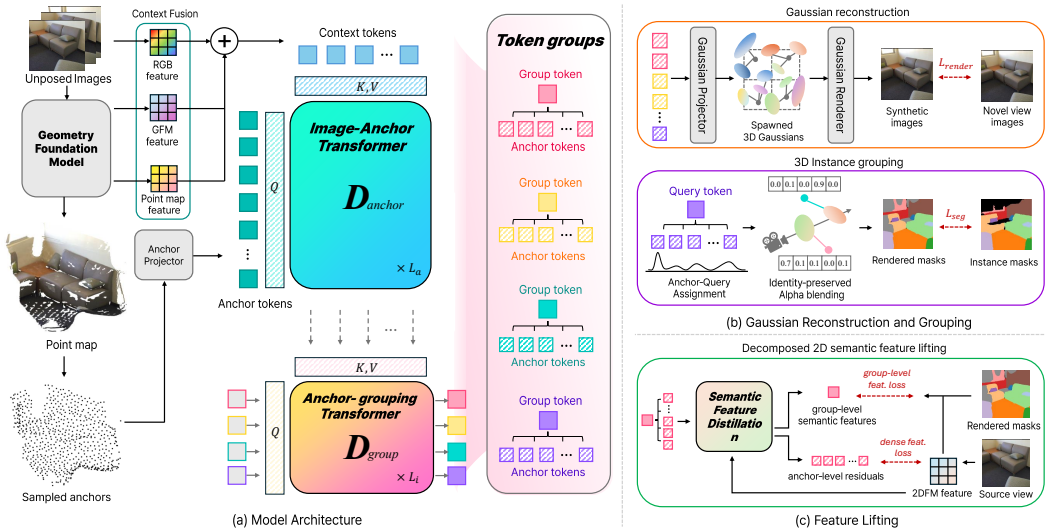


Figure 2: **Overview of the 3D token group framework. (a)** Multi-view features and pointmaps from a 3D foundation model are fused into context tokens. The image-anchor decoder $\mathcal{D}_{\text{anchor}}$ decodes anchor tokens from them, and the anchor-grouping decoder $\mathcal{D}_{\text{group}}$ produces group tokens defining instance-level assignments. **(b)** The framework is trained by 2D reconstruction-segmentation supervision: RGB images for anchor tokens, and instance masks for token grouping. **(c)** The token groups support semantic feature lifting decomposed into group- and anchor-level components.

to select K anchor positions $\{a_k\}_{k=1}^{K}$. The k-th anchor token at position a_k is initialized as

$$A_k^{(0)} = x_{a_k} + \phi_{\text{pos}}(a_k), \tag{1}$$

where $x_{a_k} \in \mathbb{R}^C$ is the context feature in X at the patch selected as anchor position a_k, and ϕ_{pos} is a 2-layer MLP projecting the 3D coordinate a_k to the feature dimension. We initialize L group tokens $G^{(0)} = \{G_\ell^{(0)}\}_{\ell=1}^{L}$ as learnable embeddings.

Token group decoding. The token group decoder consists of two cross-attention transformers: an image-anchor decoder $\mathcal{D}_{\text{anchor}}$ and an anchor-group decoder $\mathcal{D}_{\text{group}}$. The image-anchor decoder grounds the anchor tokens in the multi-view context by cross-attending to the context features X defined above. The anchor-group decoder then updates the group tokens by cross-attending to the decoded anchors, allowing each group token to aggregate object-level information:

$$A = \mathcal{D}_{\text{anchor}}(A^{(0)}, X), \qquad G = \mathcal{D}_{\text{group}}(G^{(0)}, A), \tag{2}$$

where $\mathcal{D}(Q, Z)$ denotes a cross-attention transformer that updates queries Q using context Z. The decoded anchor tokens are used to reconstruct Gaussian primitives, while the decoded group tokens serve as grouping queries for assigning anchors to object instances.

Anchor-to-group assignment. Each decoded anchor's assignment probability over the L groups is computed by a softmax over dot-product similarities with the group tokens:

$$\pi_{k,\ell} = \text{softmax}\big(\{\langle A_k, G_{\ell'}\rangle\}_{\ell'=1}^{L}\big)_\ell. \tag{3}$$

The softmax induces competition among group tokens for anchor ownership, encouraging each anchor to belong to a single group — analogous to the slot competition in [17].

Gaussian reconstruction. Each decoded anchor A_k at position a_k is mapped to N_g 3D Gaussians by a 2-layer MLP that predicts Gaussian attributes: position offsets relative to a_k, scale, rotation, opacity, and spherical harmonics. Each generated Gaussian inherits the assignment score π of its parent anchor, yielding L instance-level groups that can be independently rendered and manipulated.

3.2 Training via joint reconstruction and grouping supervision

The decoders are trained entirely through 2D supervision: RGB images supervise reconstruction quality via the anchor tokens, while instance masks supervise grouping via the group tokens.

Rendering supervision. The predicted Gaussians are rendered at target viewpoints and supervised against the ground-truth images with a combined MSE and perceptual loss:

$$\mathcal{L}_{\text{render}} = \mathcal{L}_{\text{mse}} + \lambda_{\text{lpips}}\,\mathcal{L}_{\text{lpips}}. \tag{4}$$

Grouping supervision via 2D instance segmentation. We cast anchor-to-group assignment as a 2D instance segmentation problem. Each Gaussian inherits the assignment probability $\pi_{k,\ell}$ of its parent anchor; rendering these probabilities through alpha compositing produces L instance probability maps $\{M_\ell\}_{\ell=1}^{L}$. Following standard 2D segmentation pipelines [3, 7], we perform Hungarian matching between $\{M_\ell\}$ and the ground-truth 2D instance masks $\{\hat{M}_n\}_{n=1}^{N}$. Specifically, the matching cost between group ℓ and GT instance n is computed on the (detached) rendered probability maps as the same Dice-and-BCE terms used in the mask loss, $\mathcal{C}(\ell, n) = \lambda_{\text{dice}}\big(1 - \text{Dice}(M_\ell, \hat{M}_n)\big) + \lambda_{\text{bce}}\,\text{BCE}(M_\ell, \hat{M}_n)$, and the optimal assignment is obtained by the Hungarian algorithm. For each matched pair (ℓ, n), we apply a per-pixel binary cross-entropy (BCE) loss $\mathcal{L}_{\text{bce}}$ and a Dice loss [19] $\mathcal{L}_{\text{dice}}$ between the predicted mask M_ℓ and the matched ground-truth mask $\hat{M}_n$:

$$\mathcal{L}_{\text{seg}} = \lambda_{\text{bce}}\,\mathcal{L}_{\text{bce}} + \lambda_{\text{dice}}\,\mathcal{L}_{\text{dice}}. \tag{5}$$

The anchor-group assignment is computed via a softmax over the L groups and an additional zero-valued void channel to account for anchors in non-instance regions. The softmax with the void channel implicitly discourages unmatched groups from acquiring anchors. This 2D segmentation objective drives the group tokens to represent coherent entities while organizing anchors into instance-level groups. The full training loss combines both objectives:

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \lambda_{\text{seg}}\,\mathcal{L}_{\text{seg}}. \tag{6}$$

As the reconstruction is not yet stable early in training, we apply a linear warm-up to λ_{seg} over the first few steps, ensuring the grouping supervision takes full effect once initial geometry has emerged.

3.3 Decomposed semantic feature distillation

The instance structure learned in Sections 3.1 and 3.2 provides a natural basis for integrating semantics. Rather than attaching a high-dimensional feature vector independently to every Gaussian — as prior methods do — our token groups enable a decomposed representation: a shared group-level embedding $s_\ell \in \mathbb{R}^D$ captures the dominant semantics of each instance, while a low-dimensional anchor-level residual $r_k \in \mathbb{R}^d$ ($d \ll D$) accounts for spatially varying detail within the group. This decomposition reduces semantic storage by orders of magnitude while preserving local specificity.

Semantic token encoding. Given per-view 2D foundation features $\{\Phi_i\}_{i=1}^{V}$, we reuse the trained image-anchor cross-attention $\mathcal{D}_{\text{anchor}}$ to aggregate them into anchor semantic tokens, following [1]. The group-level semantic tokens $\{s_\ell\}_{\ell=1}^{L}$ are then produced by an additional anchor-to-group cross-attention transformer whose queries are initialized from the learned group tokens G_ℓ via a linear projection. Intuitively, each group token queries the anchor semantic tokens to summarize the semantic content of its assigned instance. The low-dimensional projections of the anchor semantic tokens form the anchor-level residuals $\{r_k\}$, capturing local semantic variations within each group.

Group feature alignment. At rendering time, each anchor is hard-assigned to its highest-probability group, and the tuple $[\text{onehot}(\pi_{k,\ell}), r_k]$ is attached to every Gaussian spawned from A_k. Rendering produces per-view group assignment maps $\hat{S}_{v,\ell}$ and residual maps $\hat{R}_v$. The full per-pixel semantic feature is reconstructed as

$$F_v(u) = \sum_{\ell} \hat{S}_{v,\ell}(u)\, s_\ell + W_r\, \hat{R}_v(u), \tag{7}$$

where W_r projects the residual back to the foundation feature dimension. We optimize two complementary losses:

$$\mathcal{L}_{\text{distill}} = \sum_{v}\sum_{u}\big(1 - \cos\big(F_v(u),\, \Phi_v(u)\big)\big) + \sum_{v,\ell}\big(1 - \cos\big(s_\ell,\, \text{avg}_{M_\ell^v}(\Phi_v)\big)\big), \tag{8}$$

where $\text{avg}_{\hat{M}_\ell^v}(\Phi_v)$ denotes the average of the foundation features over the rendered mask prediction M_ℓ^v of group ℓ in view v. The first term drives the full reconstructed feature $F_v(u)$ to match the foundation model output at every pixel. The second term is a group-level alignment loss that directly supervises each s_ℓ to capture an object-level semantic summary, ensuring that the anchor residuals r_k need only model sub-instance variation rather than carrying the full semantic feature. Together, the two terms enforce a clean division of roles between the group and anchor levels.

Table 1: **Quantitative reconstruction and feature lifting results** with 2 context views. #Sem. units reports the number of primary semantic units (per Gaussian for the baselines, per instance group for our model). Feat. size reports the total number of stored feature scalars. †: Uni3R stores a compressed 64-dim feature per Gaussian. ‡: Our model stores 512-dim group features with 8-dim anchor residuals. **Bold** indicates the best result and underline indicates the second-best.

Method	Source view feature		Target view feature		Target view reconstruction			Representation cost		
	mIoU↑	Acc.↑	mIoU↑	Acc.↑	PSNR↑	SSIM↑	LPIPS↓	#Gauss	#Sem. units	Feat. size
LSeg [14]	0.470	0.789	0.482	0.793	–	–	–	–	–	–
LSM [9]	0.527	0.810	0.512	0.795	24.24	0.821	0.222	131,072	131,072	67.1 M
Uni3R [25]	0.540	**0.826**	0.558	**0.827**	**25.53**	**0.873**	**0.138**	131,072	131,072	8.4 M†
C3G [1]	0.542	0.803	0.513	0.783	23.89	0.770	0.285	**2,048**	2,048	1.0 M
Ours	**0.661**	0.786	**0.657**	0.789	25.28	0.771	0.238	32,768	**<100**	**59.4 K**‡

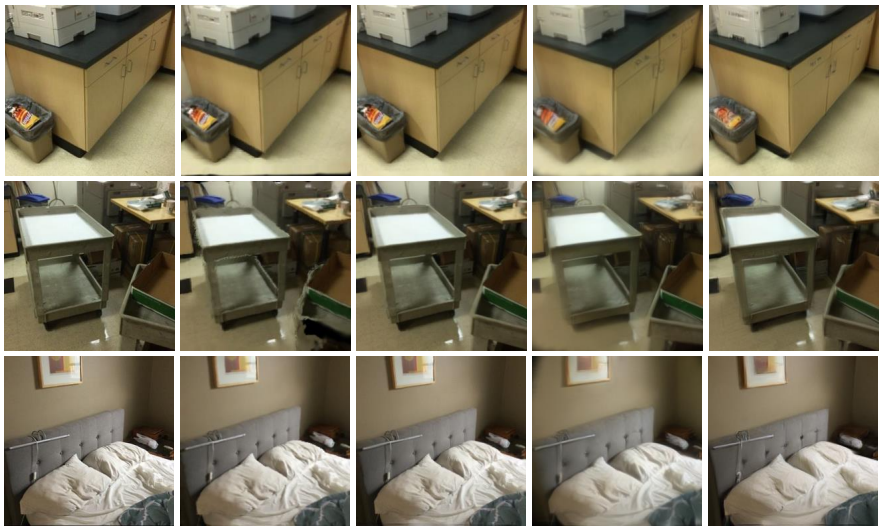


Figure 3: **Qualitative reconstruction results** with 2 context views.

4 Experiments

We evaluate the proposed token group representations and the tokenization framework on the ScanNet dataset [8]. We first assess the tokenizer in terms of reconstruction quality, feature lifting, and class-agnostic instance segmentation. We then demonstrate the broader applicability of our token groups through the instance-level token manipulations and text-based 3D instance retrieval.

Implementation details. We set $d = 8$ for low-dimensional anchor-level residuals, maximum number of groups $L = 100$, $N_g = 32$, and $K = 512$ anchor tokens. The image-anchor decoder consists of 6 transformer layers, and the anchor-group decoder of 4. We build our framework upon the pretrained VGGT [27] and follow the training protocol of Uni3R [25]. To resolve the scale misalignment between the pretrained VGGT and the ground-truth data, we first fine-tune VGGT with an auxiliary pixel-aligned Gaussian head for a few epochs, which takes less than 6 hours on 4 H200 GPUs. We then train our tokenizer with two input configurations: the 2-view setup is trained for approximately 20 hours on 4 RTX A6000 GPUs, while the 8-view setup is trained for approximately 12 hours on 4 H200 GPUs. We set $\lambda_{\text{lpips}} = 0.05$ and $\lambda_{\text{seg}} = 0.1$. We apply λ_{seg} warm-up for first 1,500 steps. Please refer to the supplements for additional implementation details.

4.1 Tokenization performance

We evaluate our tokenization framework on three tasks: feed-forward novel-view reconstruction, open-vocabulary feature lifting, and class-agnostic instance segmentation. For reconstruction and feature lifting, we follow the evaluation protocol and source/target camera sampling of LSM [9], reporting PSNR, SSIM, and LPIPS for reconstruction, and mIoU and pixel accuracy (Acc.) using

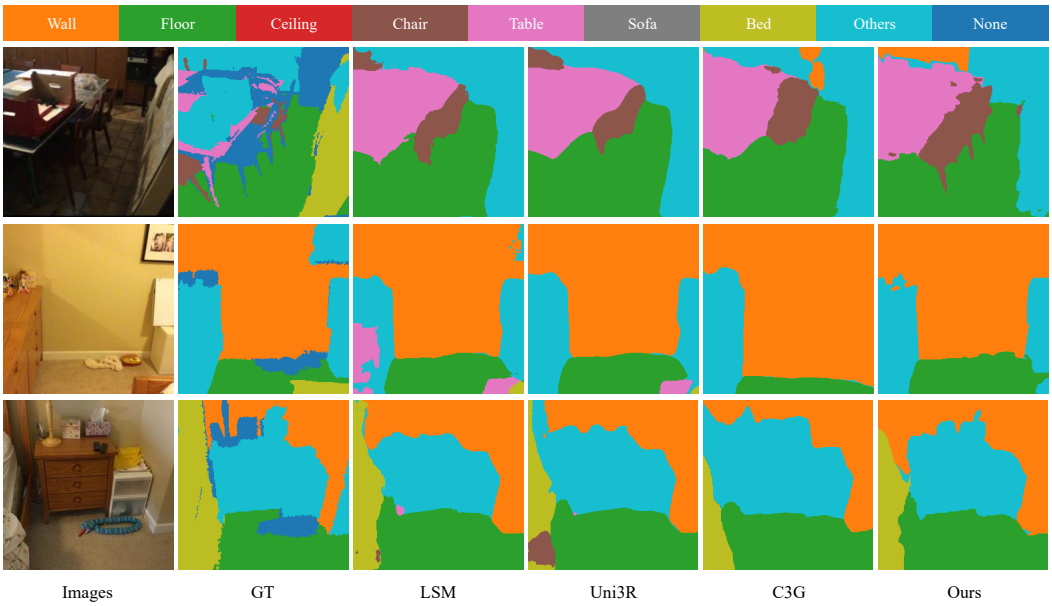


Figure 4: **Qualitative open-vocabulary novel view segmentation results** with LSeg features.

LSeg [14] for feature lifting, having $D = 512$ semantic features. For instance segmentation, we report target-view AP, AP50, and AP25 alongside reconstruction metrics on the same views. We also qualitatively evaluate token-level manipulation and open-vocabulary 3D instance retrieval.

Reconstruction and feature lifting results. Table 1 reports results on ScanNet with 2 context views. Our model achieves the best feature-lifting mIoU on both source and target views by a clear margin. This advantage reflects a fundamental property of our representation: rather than storing semantic features independently at every Gaussian, our token groups concentrate semantics at the instance level, reducing semantic storage from 8.4M scalars (Uni3R) to 59.4K. On reconstruction, Uni3R and LSM achieve stronger PSNR and SSIM at the cost of 131,072 unstructured per-pixel Gaussians, a scale our compact token-based representation does not match by design. Importantly, this reconstruction gap narrows considerably in zero-shot transfer to MipNeRF360 (Appendix F.2), suggesting that the anchor-group structure learns a more transferable scene prior. Qualitative reconstruction results are shown in Figure 3. Figure 4 shows open-vocabulary feature lifting results, where our token groups produce semantic maps with more coherent object boundaries than competing methods.

Class-agnostic instance segmentation results. Table 2 reports results with 8 context views. Our feed-forward model achieves the best segmentation across all AP metrics, surpassing both per-scene optimization baselines (Gaussian Grouping [31], ObjectGS [35]) and the feed-forward method with post-hoc optimization (IGGT [15] + LUDVIG [18]). That a fully feed-forward model trained only on 2D supervision outperforms methods that optimize per-scene suggests that native instance structure is a more effective inductive bias than post-hoc grouping. The qualitative comparisons in Figure 5 make this concrete — competing methods produce fragmented, noisy boundaries particularly on large surfaces, while our token groups yield clean, consistent instance decompositions. Reconstruction quality remains competitive given that all baselines rely on per-scene optimization or pixel-aligned Gaussians, while our model operates in a single forward pass.

Direct 3D evaluation. We additionally evaluate the predicted geometry and instance structure directly in 3D space. Our method achieves a Chamfer distance of 9.9 cm and a 3D instance mIoU of 0.487, substantially outperforming the compared baselines. These results further demonstrate that our instance-structured token groups capture coherent 3D geometry and object-level structure, beyond image-space rendering quality. Detailed results are provided in Appendix D.

4.2 Applications: entity-level interfaces and operations

The token groups produced by our framework are not merely a representational convenience — they expose a direct entity-level interface through which downstream operations follow naturally, without

Table 2: **Class-agnostic novel-view instance segmentation and reconstruction** with 8 context views. AP metrics evaluate target-view instance masks. Reconstruction metrics are reported on the same target views. **Bold** indicates the best result and underline indicates the second-best.

Type	Method	Instance Segmentation			Reconstruction		
		AP↑	AP_{50} ↑	AP_{25} ↑	PSNR↑	SSIM↑	LPIPS↓
Per-scene optimization	Gaussian Grouping [31]	0.139	0.288	0.440	<u>23.20</u>	<u>0.715</u>	0.325
	ObjectGS [35]	<u>0.178</u>	<u>0.337</u>	<u>0.489</u>	**24.34**	**0.733**	**0.310**
Feed-forward + optimization	IGGT [15] + LUDVIG [18]	0.122	0.265	0.442	22.75	0.712	<u>0.323</u>
Feed-forward	Ours	**0.235**	**0.438**	**0.564**	22.41	0.709	0.355



Figure 5: **Qualitative class-agnostic instance segmentation results** with 8 context views.

additional modules, masks, or optimization. We demonstrate two such operations: instance-level scene manipulation and open-vocabulary 3D instance retrieval.

Token group manipulation. Because each token group represents exactly one instance, scene editing reduces to selecting a group and applying an elementary operation directly to its tokens and associated Gaussians. We demonstrate four such operations: *group-wise rendering* (rendering only the Gaussians of a selected group), *removal* (discarding a group), *insertion* (adding a group from another scene), and *transformation* (applying a rigid transform to a selected group). As shown in Figure 6, edits stay strictly localized to the targeted instance in all cases — neighboring objects and the background are entirely unaffected. Crucially, none of these operations require manually provided masks, post-hoc processing, or any per-scene optimization; they act directly on the token groups produced in a single forward pass. This stands in contrast to prior methods, where object-level editing requires either per-scene optimization with identity supervision [31, 35] or explicit lifting and merging of 2D predictions into 3D [15]. We note that the demonstrated scenes contain relatively well-separated objects; how manipulation holds up under heavy occlusion or object contact is an interesting direction for future work.

Open-vocabulary 3D instance retrieval. Each token group stores a shared group-level semantic embedding lifted from a 2D foundation model, enabling retrieval by matching a text or image query directly against the group embeddings. Because retrieval operates at the instance level rather than the primitive level, complexity scales linearly with the number of instances — fewer than 100 in our representation — rather than with the number of Gaussians, which reaches 131,072 in pixel-aligned baselines. As shown in Figure 7, a query retrieves complete, spatially coherent instances rather than a scattered subset of primitives, a result that follows directly from the instance structure of the representation rather than from any post-hoc aggregation. While the demonstrations here use unambiguous queries (*e.g.*, sofa, toilet), the underlying mechanism generalizes to any feature expressible by the foundation model, including finer-grained or relational queries.



Figure 6: **Instance-level token manipulation results.** Our token groups directly offer an entity-level interface, enabling instance-level rendering, transformation, insertion, and removal.

4.3 Ablations

To examine the effects of our central design choices, we conduct ablation studies on joint training and decomposed feature lifting. All ablations are conducted on ScanNet with 2 context views.

Joint training. We evaluate our joint training scheme against two variants: (1) a *sequential* variant that first trains $\mathcal{D}_{\text{anchor}}$ with the rendering loss, then freezes it and trains $\mathcal{D}_{\text{group}}$ with the mask loss; and (2) a joint variant *without* λ_{seg} *warm-up*. As reported in Table 3, the sequential variant shows degraded performance on both reconstruction and segmentation, indicating that the two decoders must be trained together for token group structure to emerge properly. The interaction between reconstruction and grouping supervision is not one-directional: joint training shapes the anchor tokens toward geometry that is compatible with coherent grouping, rather than optimizing reconstruction in isolation. Removing the warm-up degrades performance even further than sequential training. This is because applying full segmentation supervision before the reconstruction branch has converged introduces conflicting gradients early in training, destabilizing both objectives simultaneously. The warm-up resolves this by allowing initial geometry to emerge before grouping supervision takes full effect, after which both objectives reinforce rather than compete with each other.

Decomposed feature lifting. Table 4 compares three variants: anchor residuals only (Anchor), group-level features only (Group), and our full decomposition (Group + Anchor). The anchor-only variant performs worst, as low-dimensional residuals lack the capacity to represent full semantic content without the shared group embedding to anchor them. The group-only variant already achieves strong results, which is itself an informative finding: it implies that within-group semantic variation is relatively small, validating our design choice of a compact shared embedding as the primary semantic carrier. The full model gains a further improvement by adding anchor residuals to capture the sub-instance variation that the shared embedding cannot represent. Together these results confirm the division of role built into our representation – group features handle instance-level semantics, anchor residuals handle local specificity – and show that both levels contribute meaningfully.

5 Discussion

This work suggests a new direction for 3D scene representation: rather than reconstructing dense primitives and recovering structure post-hoc, we treat object instances as a native interface. Below we discuss future extensions that this token group could enable, along with its current limitations.

Toward compositional reasoning and generation. Since our representation encodes a scene as a compact set of object-aligned token groups, it offers a promising starting point for connecting 3D scenes to large models. For reasoning, a large language model could treat a scene as a small

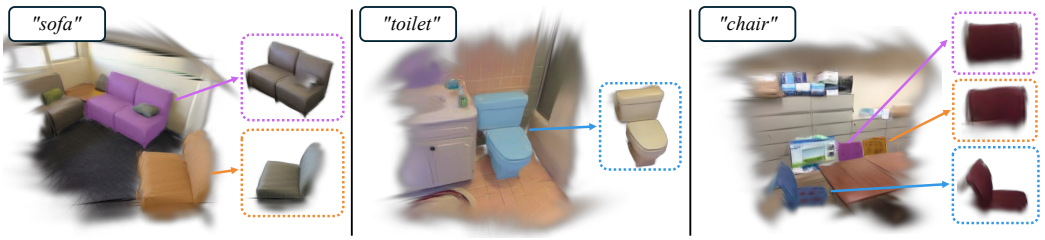


Figure 7: **Open-vocabulary 3D instance retrieval.** We present our results with lifted LSeg features. Our token groups naturally offer efficient instance-level retrieval operations without post-processing.

Table 3: **Ablations on the joint reconstruction-segmentation scheme.** 'Sequential' denotes sequentially training reconstruction and then segmentation. 'w/o warm-up' denotes the variant without λ_{seg} warm-up in early training. We report results on target viewpoints with 2 context views.

	Reconstruction			Instance Segmentation		
Training	PSNR↑	SSIM↑	LPIPS↓	AP↑	AP_{50} ↑	AP_{25} ↑
Sequential	23.65	0.737	0.348	0.032	0.097	0.315
w/o warm-up	23.09	0.732	0.329	0.081	0.186	0.415
Ours	**25.11**	**0.769**	**0.240**	**0.193**	**0.377**	**0.529**

Table 4: **Ablations on the decomposed feature lifting.** 'Anchor' denotes low-dimensional anchor residuals, and 'Group' denotes the group-level shared feature. We report feature lifting results on target views.

Feat.	mIoU↑	Acc↑
Anchor	0.527	0.714
Group	0.638	0.769
Group + Anchor	**0.661**	**0.791**

set of entities, operating on group-level tokens and drawing on anchor tokens when finer detail is needed. For generation, since groups are organized around instances rather than spatial primitives, a generative model could synthesize each group independently, compose multiple groups into a scene, and transfer or mix groups across scenes. These directions suggest the token group as an entity-level 3D representation for compositional reasoning and generation, bridging 3D scenes and large models.

Toward object-centric world models for robotics. A robotic agent interacting with the physical world must reason about objects — which ones are present, where they are, and how they can be manipulated — yet dominant scene representations expose no such structure natively. Our token groups offer a natural bridge: given a handful of unposed images, the framework produces a compact set of instance-level handles that map directly onto the entities a robot needs to reason about. For perception, the group-level semantic embeddings support grounding natural language instructions ("pick up the chair") to specific token groups without additional modules. For planning, the instance-level manipulation interface provides exactly the kind of object-level mental simulation a robotic world model needs to evaluate candidate actions before executing them. For efficiency, operating over fewer than 100 instance tokens rather than tens of thousands of primitives makes forward prediction far more tractable at the timescales robotics demands. Together these properties suggest that instance-structured token groups could serve as the perceptual front-end of an object-centric world model, connecting raw multi-view observations to the entity-level representations that planning and manipulation policies can most naturally operate over.

Limitations. Our evaluation focuses primarily on bounded indoor scenes, and scaling to outdoor environments and large-scale scenes remains an open challenge — the fixed upper bound of $L = 100$ groups and the model training both likely to require revisiting at larger scale. The single shared group-level token may also have insufficient expressivity to fully capture the semantics of complex or highly varied instances; employing multiple shared semantic tokens as basis features within each group is a natural extension. Finally, the current framework assumes static scenes; extending to dynamic settings with moving objects would be a prerequisite for the robotic applications.

6 Conclusion

We presented a feed-forward framework that reconstructs 3D scenes as instance-structured token groups, making object instances a first-class element of the representation. Within each group, an instance token summarizes entity-level identity while anchor tokens encode local geometry and appearance. The resulting token groups support accurate reconstruction and scene understanding, while directly enabling instance-level manipulations and retrieval without post-hoc processing.

Acknowledgments and Disclosure of Funding

This work was supported in part by the Institute of Information Communications Technology Planning Evaluation (IITP) grants funded by the Korea government (MSIT) (Nos. RS-2020-II201361 and RS-2024-00457882), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-00554790), and the AI Seoul Tech Research Support Program of the Seoul Future Foundation.

References

[1] Honggyu An, Jaewoo Jung, Mungyeom Kim, Sunghwan Hong, Chaehyun Kim, Kazumi Fukuda, Minkyeong Jeon, Jisang Han, Takuya Narihira, Hyuna Ko, Junsu Kim, Yuki Mitsufuji, and Seungryong Kim. C3G: Learning compact 3D representations with 2K gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[4] Rohan Chacko, Nicolai Häni, Eldar Khaliullin, Lin Sun, and Douglas Lee. Lifting by gaussians: A simple, fast and flexible method for 3D instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[5] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19457–19467, 2024.

[6] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 370–386. Springer, 2024.

[7] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[9] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[10] Tsuheng Hsu, Guiyu Liu, Juho Kannala, and Janne Heikkilä. Scene-agnostic object-centric representation learning for 3D gaussian splatting. *arXiv preprint arXiv:2604.09045*, 2026.

[11] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[13] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–91. Springer, 2024.

[14] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations (ICLR)*, 2022.

[15] Hao Li, Zhengyu Zou, Fangfu Liu, Xuanyang Zhang, Fangzhou Hong, Yukang Cao, Yushi Lan, Manyuan Zhang, Gang Yu, Dingwen Zhang, and Ziwei Liu. IGGT: Instance-grounded geometry transformer for semantic 3D reconstruction. In *International Conference on Learning Representations (ICLR)*, 2026.

[16] Yu Liu, Baoxiong Jia, Yixin Chen, and Siyuan Huang. SlotLifter: Slot-guided feature lifting for learning object-centric radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[17] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[18] Juliette Marrie, Romain Ménégaux, Michael Arbel, Diane Larlus, and Julien Mairal. LUDVIG: Learning-free uplifting of 2D visual features to gaussian splatting scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[19] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.

[20] Phuc Nguyen, Minh Luu, Anh Tran, Cuong Pham, and Khoi Nguyen. Any3DIS: Class-agnostic 3D instance segmentation by 2D mask tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[21] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[22] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] Hongyu Shen, Junfeng Ni, Yixin Chen, Weishuo Li, Mingtao Pei, and Siyuan Huang. Trace3D: Consistent segmentation lifting via gaussian instance tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[24] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3R: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024.

[25] Xiangyu Sun, Haoyi Jiang, Liu Liu, Seungtae Nam, Gyeongjin Kang, Xinjie Wang, Wei Sui, Zhizhong Su, Wenyu Liu, Xinggang Wang, and Eunbyung Park. Uni3R: Unified 3D reconstruction and semantic understanding via generalizable gaussian splatting from unposed multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[26] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-vocabulary 3D instance segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[27] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5294–5306, 2025.

[28] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709, 2024.

[29] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. SAM3D: Segment anything in 3D scenes. *arXiv preprint arXiv:2306.03908*, 2023.

[30] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *International Conference on Learning Representations (ICLR)*, 2025.

[31] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 162–179. Springer, 2024.

[32] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

[33] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. kMaX-DeepLab: k-means mask transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[34] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (SIGGRAPH)*, 2018.

[35] Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. ObjectGS: Object-aware scene reconstruction and scene understanding via gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[36] Lojze Zust, Yohann Cabon, Juliette Marrie, Leonid Antsfeld, Boris Chidlovskii, Jerome Revaud, and Gabriela Csurka. PanSt3R: Multi-view consistent panoptic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

A Societal impacts

Instance-structured 3D representations benefit robotics, AR/VR, and content creation by making scenes editable at the object level. However, reconstructing and recomposing real environments from casual captures raises privacy concerns, and token-level editing lowers the barrier to fabricated 3D scenes. We encourage provenance signals, consented capture, and disclosure of edits, and view detection of manipulated 3D content as an important complement to the capabilities introduced here.

B Additional implementation details

All feature dimensions are set to 1024, and our model takes 256×256 images as input and renders at the same resolution. The Gaussian head predicts, for each anchor, the scale, rotation, opacity, and per-Gaussian local offsets relative to the anchor center, together with spherical harmonics coefficients of degree 2. For the VGGT backbone, we follow Uni3R and change the initial DINO patch size from 14 to 16 to accommodate 256×256 inputs, and add a linear layer that takes the ground-truth camera intrinsics as an additional input. We implement our framework in PyTorch with `bf16` mixed precision and rasterize the Gaussians with `gsplat` [32]. All models are optimized with AdamW (learning rate 1×10^{-4}, weight decay 0.05) using a linear warm-up followed by cosine decay to 1×10^{-6}, with gradient clipping at 0.5. For semantic feature lifting, we train an additional group-token decoder of 4 cross-attention transformer layers on top of the frozen tokenizer. Training the feature-lifting model takes less than 3 hours on 4 RTX A6000 GPUs under the 2-view setup.

C Experiment setup details

C.1 ScanNet class-agnostic novel-view instance segmentation

Common setup. We follow the 40-scene ScanNet test subset introduced by LSM [9]. For each scene, we sample frames from the full sequence with stride 10 and reconstruct scene-level COLMAP [22] cameras from the sampled frames. Among the sampled frames, we use an interleaved split with 8 training views and 7 test views. All compared methods share the same COLMAP initialization and the same train/test view split, and evaluation is conducted on the test views.

Per-scene Gaussian labeling baselines. Gaussian Grouping [31] and ObjectGS [35] are trained in a per-scene manner using the shared COLMAP initialization described above. Their predictions are rendered on the same test views for evaluation. For both Gaussian Grouping and ObjectGS, AP confidence is computed from rendered per-pixel scores.

IGGT + LUDVIG with mask-regularized 3DGS. For IGGT [15] + LUDVIG [18], we use a mask-regularized 3DGS backbone instead of vanilla 3DGS for label uplifting. This follows the panoptic regularization strategy adopted in PanSt3R [36], which helps reduce the tendency of Gaussians optimized only with RGB reconstruction to spread across object boundaries. We use the IGGT instance masks as input labels and convert them into one-hot instance labels. We then uplift these labels to the mask-regularized 3DGS scene using LUDVIG. The uplifted labels are projected to the test views to obtain the final novel-view instance predictions. For this method, the confidence used for AP computation is derived from the raw scores produced during lifting and reprojection.

D Direct 3D Evaluation

While the main experiments evaluate reconstruction and instance segmentation through rendered views, we additionally assess the predicted representation directly in 3D. We use the same 40-scene ScanNet test subset and align the predicted geometry to the ground-truth coordinate frame using a similarity transform estimated from camera poses. To focus on the observed scene regions, evaluation is restricted to ground-truth vertices visible from at least one of the eight input views, with visibility determined using ground-truth depth. All compared baselines use COLMAP poses, whereas our model reconstructs the scene directly from unposed images.

Table 5: Direct 3D geometry evaluation on the 40-scene ScanNet test set. All predicted point clouds are voxelized to 2 cm resolution. Chamfer distance (CD), accuracy (Acc.), and completeness (Comp.) are in cm; F-scores are in percent. The evaluation is restricted to ground-truth vertices visible from at least one of the eight input views.

Method	CD ↓	$\mathrm{CD}_{\mathrm{med}}$ ↓	Acc. ↓	Comp. ↓	F@2 cm ↑	F@5 cm ↑	F@10 cm ↑
Gaussian Grouping	26.1	20.6	20.1	21.1	10.4	25.7	41.4
ObjectGS	41.2	21.6	25.6	17.6	9.9	25.6	41.9
IGGT + LUDVIG	27.2	21.9	21.2	22.6	10.1	25.0	40.7
Ours	**9.9**	**6.8**	**8.6**	**5.0**	**13.4**	**42.3**	**69.0**

Table 6: Direct 3D instance evaluation on the 40-scene ScanNet test set. We report Hungarian-matched instance mIoU, instance-level F1 at IoU thresholds of 0.25 and 0.50, AP averaged over IoU thresholds from 0.50 to 0.95 in increments of 0.05, and AP at IoU thresholds of 0.50 and 0.25.

Method	mIoU ↑	F1@25 ↑	F1@50 ↑	AP ↑	AP50 ↑	AP25 ↑
Gaussian Grouping	0.169	0.229	0.089	0.007	0.030	0.100
ObjectGS	0.176	0.283	0.107	0.005	0.018	0.115
IGGT + LUDVIG	0.140	0.205	0.075	0.005	0.018	0.100
Ours	**0.487**	**0.496**	**0.352**	**0.112**	**0.343**	**0.583**

D.1 3D geometry

Evaluation protocol. We voxelize the predicted point clouds at a common resolution of 2 cm to account for differences in the number of points across methods. We then compare them against the ScanNet ground-truth mesh using Chamfer distance, accuracy, completeness, and F-scores at distance thresholds of 2, 5, and 10 cm.

Results. As shown in Table 5, our method achieves substantially lower geometric errors and consistently higher F-scores than the compared baselines. The improvement is particularly pronounced in completeness and at larger F-score thresholds, suggesting that the anchor-based representation captures the overall scene geometry more faithfully despite using a compact set of predicted Gaussians.

D.2 3D instance segmentation

Evaluation protocol. We evaluate the predicted instance assignments directly on the visible ground-truth mesh vertices. Each vertex inherits the instance label of its nearest Gaussian mean within a distance threshold. We report Hungarian-matched mIoU, treating unmatched ground-truth instances as zero, along with instance-level F1 at IoU thresholds of 0.25 and 0.5 and average precision (AP) across standard IoU thresholds.

Results. Table 6 shows that our token groups consistently outperform the baselines across all 3D instance metrics. Unlike methods that recover instance identities from individual Gaussians or lifted 2D predictions, our representation organizes the scene into instance-level groups directly. The strong 3D results, together with the novel-view segmentation performance reported in the main paper, indicate that these groups capture coherent object instances in 3D rather than merely producing consistent masks across rendered views. The advantage also holds when evaluation is restricted to the 18 official ScanNet instance classes, where our method achieves the highest mIoU.

Comparison with Uni3R + SAM2. We further compare against Uni3R [25] combined with SAM2 [21] instance predictions to examine whether instance-level structure can be recovered from a primitive-based feed-forward representation. This comparison complements the per-scene optimization and feature-lifting baselines above by considering a feed-forward reconstruction backbone equipped with off-the-shelf instance segmentation.

Table 7: Direct 3D instance segmentation comparison with Uni3R + SAM2 on the 40-scene ScanNet test set. Both methods are evaluated using the same 3D instance metrics.

Method	mIoU ↑	F1@25 ↑	F1@50 ↑	AP ↑	AP50 ↑	AP25 ↑
Uni3R + SAM2	0.360	0.330	0.187	0.047	0.159	0.412
Ours	**0.487**	**0.496**	**0.352**	**0.112**	**0.343**	**0.583**

Table 8: Effect of the number of group tokens L on reconstruction and instance segmentation with two context views on ScanNet.

L	Reconstruction			Instance Segmentation		
	PSNR ↑	SSIM ↑	LPIPS ↓	AP ↑	AP50 ↑	AP25 ↑
30	**25.41**	**0.773**	**0.237**	0.170	0.350	0.472
50	25.20	0.771	0.240	0.184	0.358	0.465
100 (default)	25.28	0.771	0.238	0.194	0.377	0.525
200	25.26	0.772	0.238	**0.207**	**0.424**	**0.579**

As shown in Table 7, our method consistently outperforms Uni3R + SAM2 across all 3D instance metrics. While combining feed-forward reconstruction with 2D instance predictions provides a strong baseline, the resulting 3D instance assignments remain less accurate than those of our token groups. This further supports learning instance structure as part of the 3D representation itself, rather than recovering it from independently predicted primitives.

E Additional ablation studies

We further examine the effects of two design parameters: the number of instance groups and the dimensionality of the anchor-level semantic residuals. All experiments are conducted on ScanNet with two context views, following the setting of the main-paper ablations.

E.1 Number of groups

The number of group tokens L determines the maximum number of instances represented in a scene. We vary L while keeping the remaining configuration fixed.

As shown in Table 8, reconstruction quality remains largely unchanged across different group counts, while instance segmentation generally improves as more groups become available. This suggests that increasing the number of groups primarily affects the granularity of instance decomposition rather than the underlying reconstruction. We use $L = 100$ as the default, balancing segmentation quality with a compact instance-level representation.

E.2 Anchor residual dimensions

We also study the dimensionality d of the anchor-level residuals used in decomposed semantic feature lifting. The group-level semantic features are kept fixed, while d controls the capacity for representing local variations within each instance.

Table 9 shows that increasing the residual dimension generally improves feature lifting, although the gains become marginal beyond $d = 8$. A larger residual dimension provides more capacity for local semantic detail but also increases feature storage, reducing the efficiency of the decomposed representation. We therefore adopt $d = 8$, which retains most of the benefit of higher-dimensional residuals at a lower representation cost.

F Additional generalization experiments

We complement our main ScanNet experiments with two studies that test how our tokenization framework behaves beyond the in-domain setting: (i) training on RealEstate10K with SAM2 pseudo-

Table 9: Effect of the anchor residual dimension d on semantic feature lifting with two context views on ScanNet. We report open-vocabulary semantic segmentation performance on source and target views.

d	Source View		Target View	
	mIoU ↑	Acc. ↑	mIoU ↑	Acc. ↑
4	0.642	0.772	0.646	0.776
8 (default)	0.661	0.786	0.661	0.791
16	**0.677**	0.789	**0.682**	0.793
32	0.666	**0.793**	0.666	**0.797**

Table 10: Reconstruction quality on RealEstate10K (2-view). Our model is trained with SAM2 pseudo-labels as instance supervision, while the baseline follows its original training protocol. **Bold** indicates the best result.

Method	PSNR↑	SSIM↑	LPIPS↓
C3G [1]	22.39	0.713	0.259
Ours	**22.85**	**0.746**	**0.230**

labels in place of human-annotated masks, and (ii) zero-shot transfer of a ScanNet-trained model to MipNeRF360. Both experiments use the 2-view input setting.

F.1 RealEstate10K with SAM2 pseudo-labels

Dataset and view sampling. RealEstate10K [34] consists of casually captured real-estate video clips with camera poses but no instance annotations. We follow the standard train/test scene split and use a 2-view setup throughout: for each clip, we sample two source views as input and evaluate reconstruction on held-out target views from the same clip.

SAM2 pseudo-label generation. Since RE10K provides no ground-truth instance masks, we generate pseudo-labels with SAM2 [21]. The resulting masks are used as the targets for our 2D instance segmentation loss; no human-annotated 3D or 2D labels are used at any stage of training. While SAM2 maintains object identities across frames within a clip, ID tracks are occasionally broken—e.g., when an object is briefly occluded or leaves and re-enters the view, it may be assigned a new ID upon reappearance, splitting a single physical instance across multiple supervisory IDs.

Training and baselines. We train our tokenizer on RE10K from scratch under the 2-view setup, keeping the loss formulation and hyperparameters identical to the ScanNet experiments unless noted otherwise. For reconstruction comparison, we evaluate C3G [1] on the same 2-view splits.

Results. Table 10 reports reconstruction performance on RealEstate10K under the 2-view setting. Despite relying solely on SAM2-generated pseudo-labels for instance supervision, our tokenizer outperforms C3G across all three metrics, indicating that the anchor–group tokenization remains effective even when supervised with noisy, automatically generated masks. Figure 8 shows our rendered RGB and predicted instance masks on held-out test views, along with SAM2 masks on the same views shown for reference—to give a sense of the kind of supervisory signal the model was trained from, not as a direct baseline. Our token groups produce coherent instance decompositions on scenes that lack ground-truth annotations altogether, and the decompositions remain consistent even where the reference SAM2 masks fragment a single physical instance into multiple IDs. Together, these results indicate that the instance structure emerging in our representation does not depend on dataset-specific clean labels, and can be bootstrapped from off-the-shelf 2D segmentation models.

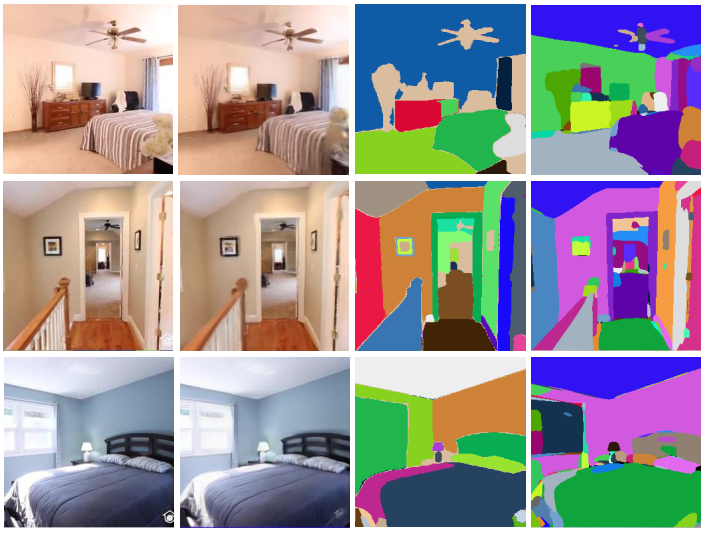


Figure 8: Qualitative results on RealEstate10K. From left to right: ground-truth RGB, our rendered RGB, our predicted instance masks, and—for reference—SAM2 masks obtained on the test view under the same protocol used to generate our training supervision. The SAM2 column is shown only to give a sense of the kind of supervisory signal our model was trained from, and is not part of the evaluation.

Table 11: Zero-shot reconstruction on MipNeRF360 (2 context views, 1 target view). All models are trained on ScanNet and evaluated on MipNeRF360 without any fine-tuning. **Bold** indicates the best result.

Method	PSNR↑	SSIM↑	LPIPS↓
Uni3R [25]	14.58	0.317	0.472
Ours	**16.52**	**0.408**	**0.439**

F.2 Zero-shot transfer to MipNeRF360

Setup. We evaluate zero-shot generalization to MipNeRF360 [2], applying our ScanNet-trained model directly without any fine-tuning. We use a 2-context, 1-target setup: for each scene, two source views are provided as input and reconstruction is evaluated on a single held-out target view.

Baseline. We compare against Uni3R [25] under the same zero-shot protocol, also applying its ScanNet-trained checkpoint directly to MipNeRF360 without fine-tuning. Both methods share the same 2-context, 1-target splits and evaluation views.

Results. Table 11 reports PSNR, SSIM, and LPIPS on the target views. Our model outperforms Uni3R across all three metrics despite the marked distribution shift from ScanNet's bounded indoor layouts to MipNeRF360's unbounded outdoor and object-centric scenes, indicating that the anchor–group tokenization captures a representation prior that holds beyond the training distribution.

G Additional qualitative results

We provide additional qualitative comparisons on three more ScanNet scenes each for novel-view reconstruction, open-vocabulary feature lifting, and class-agnostic instance segmentation, complementing the main-paper figures.

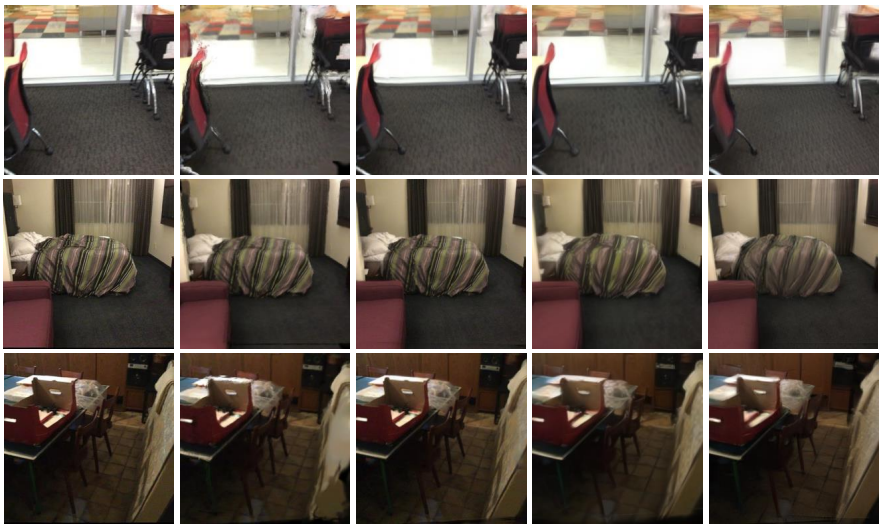


Figure 9: Additional qualitative reconstruction results on ScanNet, complementing Figure 3 of the main paper. Three additional scenes are shown; each row presents a single test view rendered by LSM [9], Uni3R [25], C3G [1], and ours, alongside the ground truth.

G.1 Additional reconstruction results

Figure 9 provides additional novel-view reconstruction comparisons on three more ScanNet scenes, extending the qualitative comparison in Figure 3 of the main paper. Across diverse layouts, our anchor–group tokenization reproduces the overall scene structure and major object appearances faithfully, while the compact token-based representation occasionally smooths over fine high-frequency details compared to pixel-aligned baselines—consistent with the small reconstruction gap reported in Table 1.

G.2 Additional feature lifting results

Figure 10 provides additional open-vocabulary feature lifting comparisons on three more ScanNet scenes, extending the qualitative comparison in Figure 4 of the main paper. The decomposed group-level and anchor-level features predicted by our model produce semantic maps that respect object boundaries and remain consistent across regions belonging to the same entity, supporting the quantitative gains in source- and target-view mIoU reported in Table 1.

G.3 Additional class-agnostic instance segmentation results

Figure 11 provides additional class-agnostic instance segmentation comparisons on three more ScanNet scenes, extending the qualitative comparison in Figure 5 of the main paper. Our token-group-based segmentation continues to produce coherent instance boundaries and avoids the fragmented regions characteristic of per-Gaussian identity baselines, particularly on large, contiguous surfaces such as walls, floors, and beds.

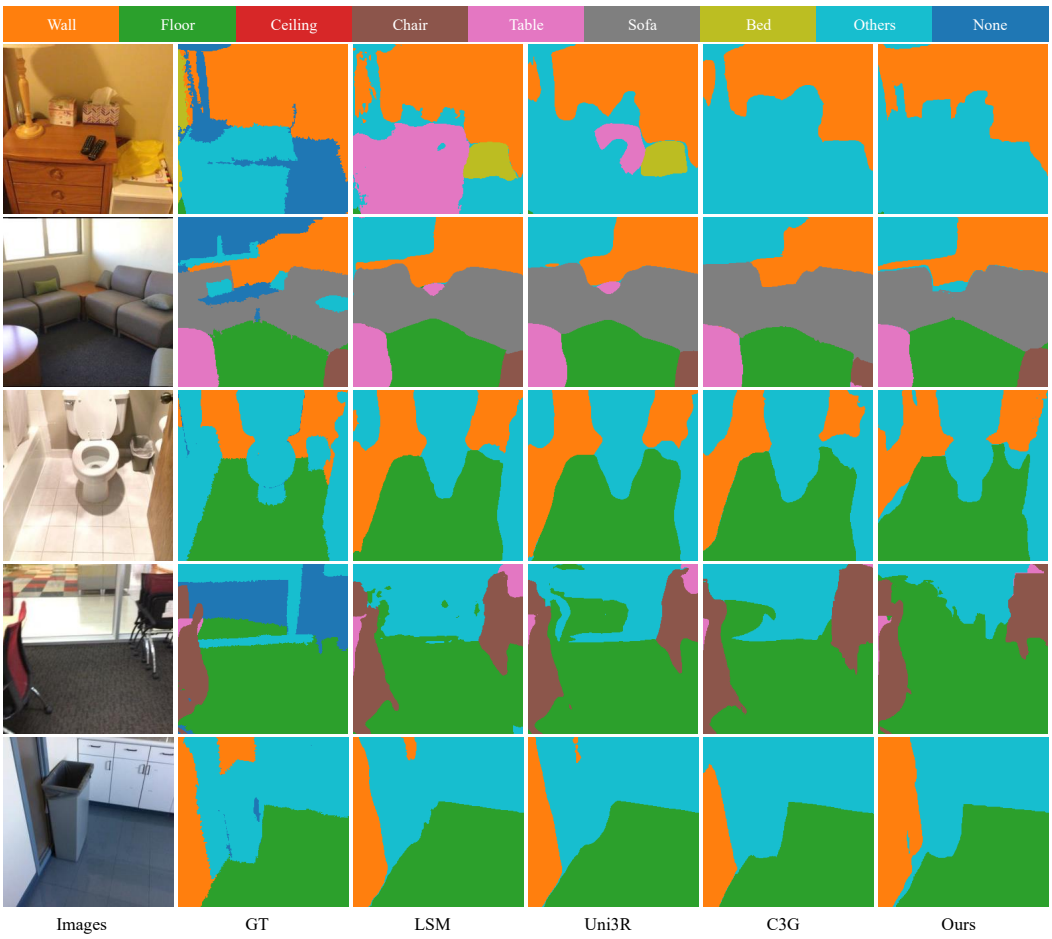


Figure 10: Additional qualitative LSeg [14] feature distillation results on ScanNet, complementing Figure 4 of the main paper. Three additional scenes are shown; each row presents the lifted semantic feature map from LSM [9], Uni3R [25], C3G [1], and ours, alongside the ground-truth segmentation.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

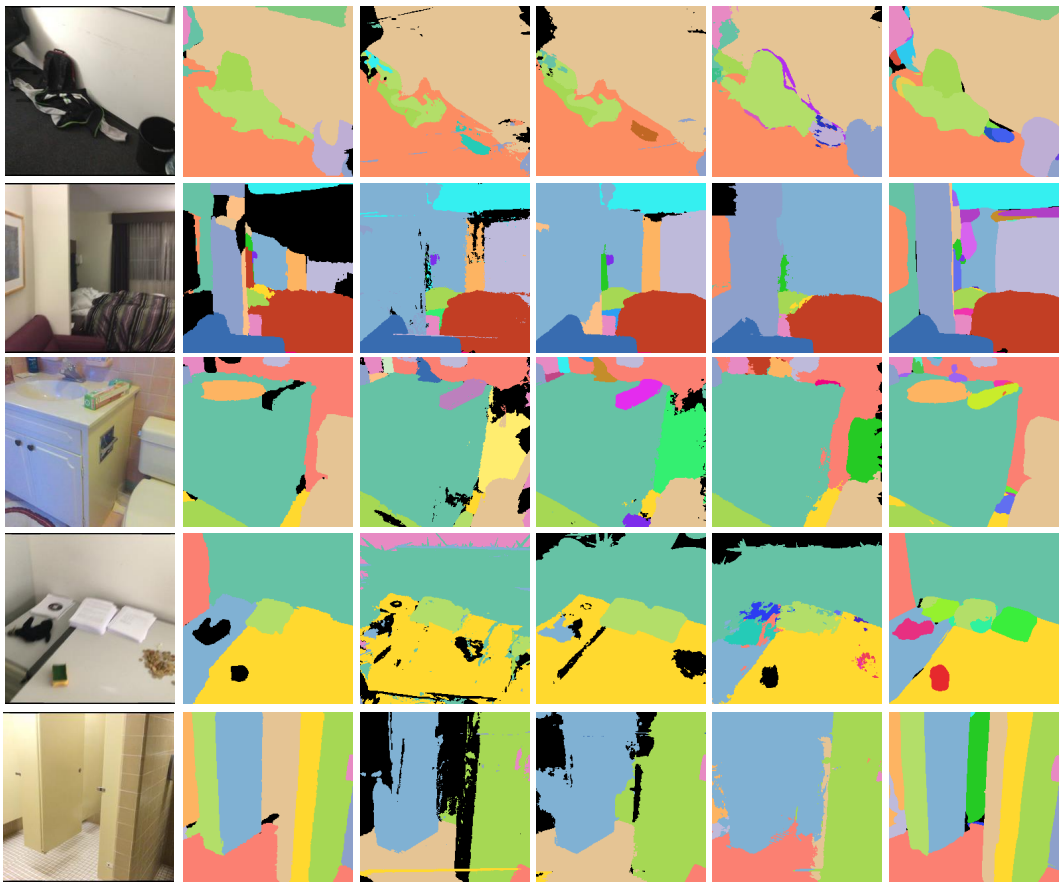


Figure 11: Additional qualitative class-agnostic novel-view instance segmentation results on ScanNet, complementing Figure 5 of the main paper. Three additional scenes are shown; each row presents the predicted instance masks from Gaussian Grouping [31], ObjectGS [35], IGGT [15]+LUDVIG [18], and ours, alongside the ground-truth RGB and instance mask.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our main contributions.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have limitations in the Discussion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification:

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The implementation details are listed in the manuscript, and the code will be made publicly available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code will be made publicly available if accepted.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We follow the training recipe of Uni3R.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We did not report error bars due to the high computational cost of repeated training runs. However, we ensure fair comparison by using consistent random seeds and fixed data splits across all methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research complies with the NeurIPS Code of Ethics. We use publicly available datasets and do not handle personally identifiable information.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive and negative societal impacts of our work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Not relevant for high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use the ScanNet dataset to evalute the performance.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Our paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: This study does not involve human subjects and does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not involved in the methodology. LLMs were only used for minor grammar editing during writing.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.