# UniC-Lift: Unified 3D Instance Segmentation via Contrastive Learning

**Ankit Dhiman**[*1,2], **Srinath R**[*1], **Jaswanth Reddy**[*1],
**Lokesh R Boregowda**[2], **Venkatesh Babu Radhakrishnan**[1]

[1]Indian Institute of Science, Bangalore
[2]Samsung R&D Institute India - Bangalore

## Abstract

3D Gaussian Splatting (3DGS) and Neural Radiance Fields (NeRF) have advanced novel-view synthesis. Recent methods extend multi-view 2D segmentation to 3D, enabling instance/semantic segmentation for better scene understanding. A key challenge is the inconsistency of 2D instance labels across views, leading to poor 3D predictions. Existing methods use a two-stage approach in which some rely on contrastive learning with hyperparameter-sensitive clustering, while others preprocess labels for consistency. We propose a unified framework that merges these steps, reducing training time and improving performance by introducing a learnable feature embedding for segmentation in Gaussian primitives. This embedding is then efficiently decoded into instance labels through a novel "Embedding-to-Label" process, effectively integrating the optimization. While this unified framework offers substantial benefits, we observed artifacts at the object boundaries. To address the object boundary issues, we propose hard-mining samples along these boundaries. However, directly applying hard mining to the feature embeddings proved unstable. Therefore, we apply a linear layer to the rasterized feature embeddings before calculating the triplet loss, which stabilizes training and significantly improves performance. Our method outperforms baselines qualitatively and quantitatively on the ScanNet, Replica3D, and Messy-Rooms datasets.

**Project Page** — https://unic-lift.github.io/
**Code** — https://github.com/val-iisc/UniC-Lift

## Introduction

Extensive research has been undertaken on various 2D segmentation variants, including semantic (Xu, Xiong, and Bhattacharyya 2023; Zhou et al. 2023b), instance (He et al. 2023; Cheng et al. 2022), and panoptic (Kirillov et al. 2019a; Hu et al. 2023) segmentation. Recently, advancements in open-set segmentation (Kirillov et al. 2023b; Zou et al. 2024) have enabled these techniques to predict classes absent in the training dataset. Building upon these 2D advancements, researchers have focused on understanding 3D scenes, which is crucial for applications in AR/VR (Choy, Gwak, and Savarese 2019), autonomous driving (Feng et al. 2019), and path planning (Bartolomei, Teixeira, and Chli 2020). A major challenge in 3D scene understanding is the lack of large-scale, richly annotated datasets, unlike 2D image datasets (e.g., Cityscapes (Cordts et al. 2016), MS COCO (Lin et al. 2015), CamVid (Brostow, Fauqueur, and Cipolla 2009)). To overcome this, many works "lift" labels from 2D segmentation methods to 3D representations such as point clouds (Liu et al. 2022), Neural Radiance Fields (NeRF) (Kundu et al. 2022; Siddiqui et al. 2023), and Gaussian Splatting (3DGS) (Zhou et al. 2023a). This approach uses efficient 2D segmentation techniques for 3D tasks but faces a key challenge: *inconsistent multi-view semantic masks from 2D models*.

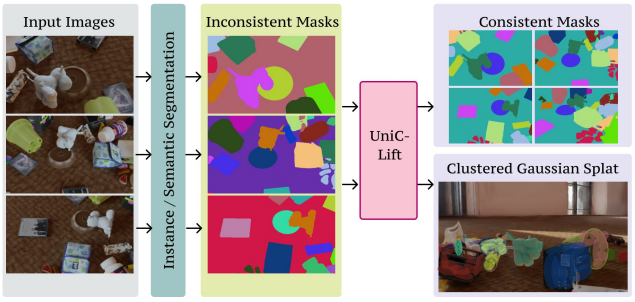


Figure 1: Our method takes a set of multi-view RGB images as input, and passes these images through a pre-trained instance/semantic segmentation method. Although accurate, segmentation methods for 2D images do not generate multi-view consistent segmentation masks. Observe how labels vary across the images. Our method formulates this as a clustering problem in 3D, utilizing feature-vectors in 3D space which are trained using contrastive losses. Our method generates consistent segmentation masks. We further motivate the necessity of this work in Fig. 2

In this work, we tackle the problem of 3D instance segmentation under inconsistent 2D instance masks. (Fig. 1). One plausible solution for the task is to use an off-the-shelf video instance segmentation method (Zhang et al. 2023; Heo et al. 2023) to obtain consistent masks for multi-view image sequences. However, these methods have limitations when there are significant occlusions and large changes in the observed scene, thus requiring joint optimization for re-

[*]These authors contributed equally.

identification task (Li and Loy 2018), which is challenging for real-world in-the-wild scenes. Recent 3D representations (NeRF, 3DGS) learn the underlying geometry of the scene from 2D multi-view images. Thus, these representations can be utilized to aggregate semantic/instance labels from 2D segmentation methods and generate 3D consistent masks. Recent works (Fu et al. 2022; Kundu et al. 2022; Bing et al. 2023; Zhi et al. 2021; Siddiqui et al. 2023) in this direction lift the 2D segmentation labels to 3D representations to achieve multi-view consistent 3D segmentation. However, a significant limitation of these methods is that they are extremely slow as they have additional expensive steps of label assignment (Siddiqui et al. 2023) in the loss function.

Another promising line of work (Kobayashi, Matsumoto, and Sitzmann 2022; Zhou et al. 2023a) is to distill features from rich 2D image feature extractors such as CLIP-LSeg (Li et al. 2022) or DINO (Caron et al. 2021) instead of lifting the 2D labels. However, these methods suffer from slow speed as they distill high-dimensional features into 3D representations. For example, on average, DFF (Kobayashi, Matsumoto, and Sitzmann 2022) takes approximately two days of training. Recently, Contrastive-Lift (Bhalgat et al. 2023) solved this problem by connecting 2D segmentation to 3D using a learnable 3D vector embedding that aligns with 2D label predictions. Contrastive-Lift achieves this by integrating slow-fast contrastive loss, eliminating the need to solve the linear assignment problem. However, this method requires an additional expensive step of HDBSCAN (McInnes, Healy, and Astels 2017) as post-processing to find the centroids to label the optimized embedding. In this work, we propose an efficient *single-stage* 3D segmentation method achieving superior performance in a fraction of training time compared to baseline.

We propose *UniC-Lift*, a novel 3D segmentation method built on 3DGS (Kerbl et al. 2023). *UniC-Lift* addresses the challenge of generating consistent 3D segmentation masks from inconsistent 2D segmentation maps. To effectively encode 3D features, we add a d-dimensional vector embedding $v \in \mathbb{R}^d$ as a property for each 3D Gaussian primitive and then rasterize this embedding to the desired camera view. Further, to address the issues at the boundary, we propose a strategy for hard-mining samples to apply the triplet loss. We propose a simple "Embedding-to-label" process to decode the rasterized embeddings to class label. This straightforward approach outperforms existing state-of-the-art 3D segmentation methods. We demonstrate the utility of *UniC-Lift* in two downstream tasks: object manipulation and object extraction. Our main contributions are as follows:

- An effective single-stage method for 3D segmentation that directly decodes the learned 3D embedding to a consistent segmentation label, given inconsistent 2D segmentation labels.
- A novel contrastive loss based on the triplets to reduce the inter-class variance which aids in accurate 3D segmentation.
- Demonstrating high-quality and accurate 3D segmentation by showing effective downstream object manipulation and extraction applications.

## Related Work

**3D Representations.** Neural Radiance Fields (Mildenhall et al. 2021) and its variants (Barron et al. 2022; Müller et al. 2022; Barron et al. 2021; Fridovich-Keil et al. 2022; Chen et al. 2022; Barron et al. 2023) utilize volumetric rendering equation which is based on ray-tracing to make a differentiable renderer. These representations perform well on the task of novel-view synthesis from input multi-view images and their camera pose. Recently, Gaussian Splatting (Kerbl et al. 2023) has revolutionized this field. It is based on rasterization instead of ray-tracing and can achieve very high FPS to render high-resolution novel-views (Lu et al. 2024). Further, these representations are extended to other tasks such as dynamic scenes (Yang et al. 2023), stylization (Huang et al. 2022; Dhiman et al. 2025), sparse-views (Li et al. 2024; Wu et al. 2023), hierarchical scenes (Dhiman et al. 2023), etc.

**Scene Understanding in 3D Representations.** Semantic-NeRF (Zhi et al. 2021) lifts 2D semantic labels to 3D by using a separate radiance field network to predict the class labels utilizing a cross-entropy based loss. Whereas NeSF (Vora et al. 2021) samples a pre-trained NeRF model to obtain the volumetric grid and convert this to a semantic grid by using a convolutional volume-to-volume network. This semantic grid is converted into class probabilities by rendering. These methods did not take into account the inconsistency in the 2D ground-truth semantic maps. Afterwards, a lot of works (Mirzaei et al. 2022; Yu, Guibas, and Wu 2021) explored the similar approach and improved the performance in scene understanding using NeRF. Similar techniques (Lan et al. 2023; Li, Liu, and Zhou 2024) have also been explored in the domain of Gaussian splatting.

Methods such as Panoptic NeRF (Fu et al. 2022) and Instance-NeRF (Liu et al. 2023) use information from 3D instances by extracting volume density from the pre-trained NeRF network. The major disadvantage of these works was the use of 3D masks or tracked object masks. Panoptic Lifting (Siddiqui et al. 2023) and DM-NeRF (Bing et al. 2023) use linear assignment problem during optimization to solve the multi-view inconsistency problem in the ground-truth 2D mask. Contrastive-Lift (Bhalgat et al. 2023) used learnable permutation-invariant embedding vectors to solve the multi-view inconsistency problem. But rely on algorithms like HDBSCAN to find the cluster centroids.

**Open-Set segmentation in 3D.** Recently, methods such as SAM (Kirillov et al. 2023a), SEEM (Zou et al. 2024) which has capability of segmenting objects in a scene by using texts or interactive scribbles. DFF (Kobayashi, Matsumoto, and Sitzmann 2022) distills the knowledge of off-the-shelf 2D image feature extractors such as CLIP-LSeg (Li et al. 2022) or DINO (Caron et al. 2021) into a 3D feature field optimized in parallel to the radiance field. Then, these distilled features are used to decompose the 3D scene and perform various editing tasks. Based on the similar strategy, other works (Tschernezki et al. 2022; Kerr et al. 2023) distill open-set semantics into 3D space which can then be used for zero-shot segmentation. Similar to NeRF. a lot of work (Shi et al. 2023; Qin et al. 2023; Zhou et al. 2023a) have also emerged for Gaussian splitting.

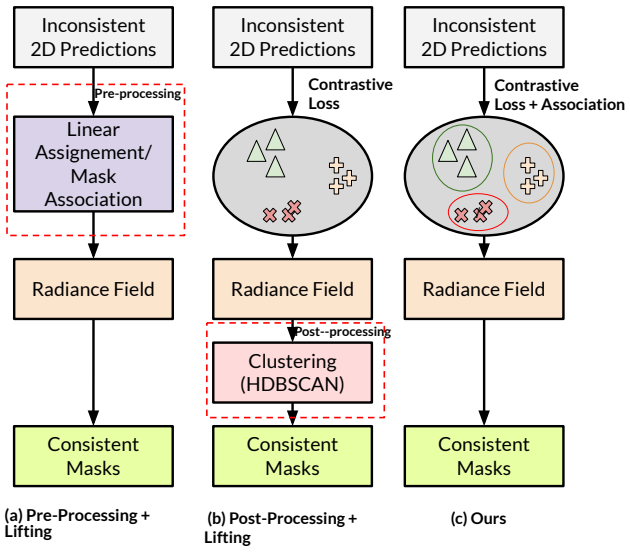


Figure 2: Previous methods are multi-stage: (a) some pre-process 2D masks before lifting them to 3D (e.g., DM-NeRF (Bing et al. 2023), Gaussian Grouping (Ye et al. 2023), and Panoptic-Lift (Kundu et al. 2022)), while others use (b) a separate clustering algorithms on learned embeddings (e.g., Contrastive-Lift (Bhalgat et al. 2023)). (c) In contrast, our method uses a single, unified representation to perform segmentation directly.

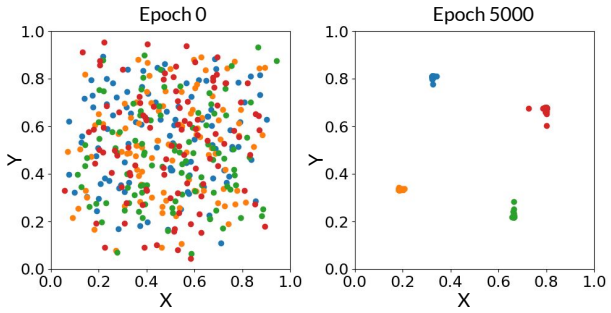


Figure 3: **Intuition of Embedding-To-Label Process.** A toy experiment demonstrating how contrastive learning enables direct label prediction. Embeddings, initially distributed in a constrained space (e.g., $[0, 1]^2$), converge to distinct corners during training. Each corner effectively becomes a unique binary code that can be thresholded and mapped to a label, removing the need for post-processing.

# Proposed Method

## Preliminaries

3D Gaussian Splatting (3DGS) models a scene using set of 3D Gaussians parameterized by position ($\boldsymbol{\mu}$), covariance ($\boldsymbol{\Sigma}$), opacity ($\alpha$), and spherical harmonics for color ($c$). Given input RGB images and camera poses, 3DGS jointly optimizes these parameters to render novel views. Gaussians are initialized from SfM point cloud and refined via "Adaptive Density Control" algorithm which splits, clones, and prunes primitives based on training gradients. During rendering, Gaussians are projected to $T$ 2D splats, rasterized, and $\alpha$-blended to get the final pixel color $C$.

$$C = \sum_{i \in T} c_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j). \tag{1}$$

The final opacity $\alpha'_i$ is determined by multiplying the learned opacity $\alpha_i$ with the 2D projected Gaussian density evaluated at the pixel location. The rendering loss function is a combination of $\mathcal{L}_1$ and D-SSIM loss:

$$\mathcal{L}_{rendering} = (1 - \lambda)\mathcal{L}_1 + \lambda \mathcal{L}_{ssim} \tag{2}$$

## Motivation

Two-stage methods like Contrastive-Lift require a computationally expensive post-processing step, using algorithms such as HDBSCAN to derive labels from optimized embeddings. This approach introduces significant computational overhead during training (Tab. 5). Further, it also results in a complexity for novel views of $O(n \log c)$, where $n$ is the number of pixels and $c$ is the number of clusters. In contrast, our unified, single-stage framework eliminates this bottleneck by directly associating labels with the optimized vectors. This design achieves a superior $O(n)$ time complexity for novel-view prediction, reducing overall inference time, and represents a fundamental architectural improvement over prior methods, as illustrated in Fig. 2.

**Embedding-To-Label Process.** A key challenge in developing a unified contrastive learning framework is the decoding of optimized embeddings into discrete labels. We illustrate the intuition behind our approach by presenting a simplified toy experiment that demonstrates how our learned embeddings are transformed into discrete labels in a unified, single-stage process. **i. Experimental Setup:** Consider a scenario where we train a model using contrastive learning on embeddings constrained to the $[0, 1]^2$ space via a sigmoid activation. The goal is to cluster these embeddings into at least four distinct classes. **ii. Observation**: Initially, the embedded vectors are randomly distributed within the square as shown in Fig. 3. Embeddings of similar instances are pulled together and different instances are pulled apart due to the applied contrastive clustering loss.As training proceeds, the embeddings align and converge towards the corners of the square, forming well-separated clusters. **iii. Conversion to Label:** The conversion to discrete labels is straightforward. For e.g, an embedding clustered near the top-right corner (e.g., $[0.9, 0.8]$), after thresholding by 0.5, yields a binary vector $[1, 1]$ and decodes to 3. Each binary vector directly decodes to a specific label (e.g., [1,1] decodes to 3, with other corners decoding to 0, 1, and 2).

## Methodology

As explained in Preliminaries, a 3D Gaussian is parametrized by $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\alpha$, and color $c$ in 3DGS. We add an extra parameter, vector $\boldsymbol{v}$, where $\boldsymbol{v} \in \mathbb{R}^d$ is a d-dimensional vector to model instance segmentation of a scene. Given set of images $\mathcal{I} = \{I_1, I_2, ..., I_N\}$, their corre-

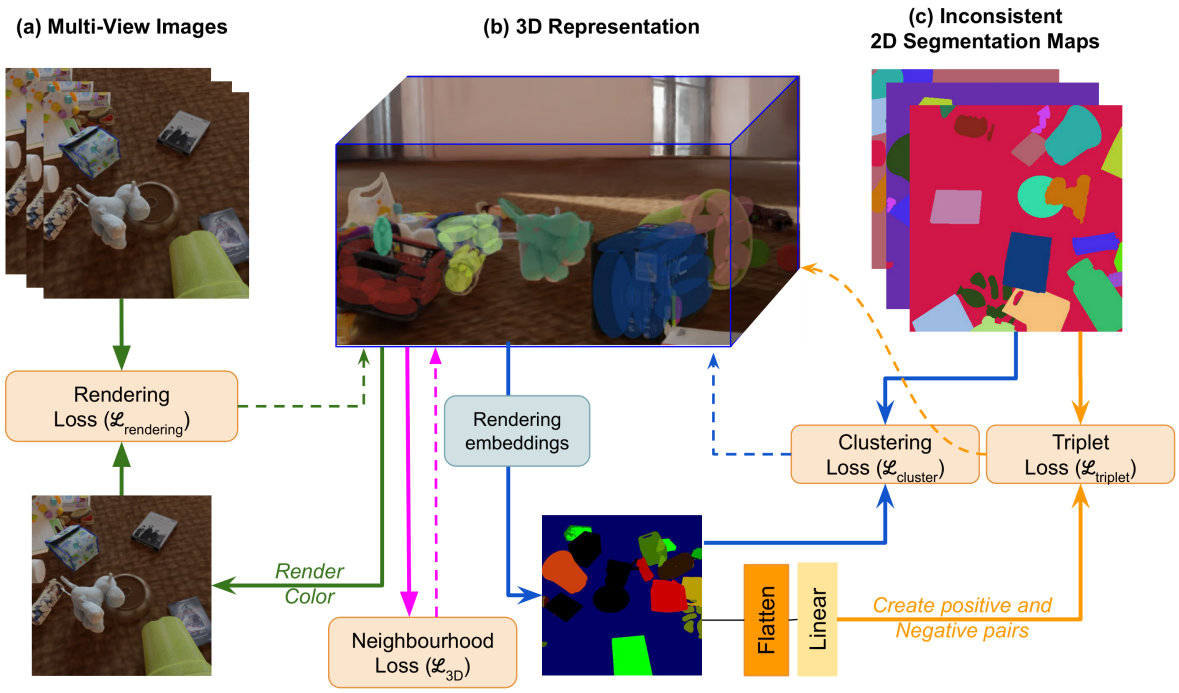


Figure 4: Overview of our pipeline. Given input (a) multi-view RGB images and (c) inconsistent segmentation maps, we optimize (b) 3D representation to lift 2D segmentation labels to 3D. To learn the view-dependent changes associated with the novel-view synthesis task, we take a rendering loss: $\mathcal{L}_{rendering}$ with GT RGB image. For optimizing the learnable 3D vector embedding we apply $\mathcal{L}_{cluster}$ to the rasterized embedding and further apply $\mathcal{L}_{triplet}$ by passing it through a linear layer. Further, we apply a 3D loss $\mathcal{L}_{3D}$ to the primitives of 3D representation. More details are in Method section.

sponding poses $\pi = \{\pi_1, \pi_2, ..., \pi_N\}$ and the semantics or instance segmentation labels $\mathcal{M} = \{M_1, M_2, ..., M_N\}$, our method also optimizes for the vector $v$ along with other parameters of 3DGS. Note that $M_i \in \mathbb{R}^{H \times W}$, $I_i \in \mathbb{R}^{H \times W \times 3}$ where $H$ and $W$ are the height and width of an input image. We render $\boldsymbol{v}$ to get vector-embedding $\boldsymbol{\mathcal{V}}$, on which we apply contrastive loss and then pass it through a linear layer. Then, we apply triplet loss between positive and negative pairs along the boundaries of the 2D segmentation map. Further, we regularize 3D Gaussians with a smoothness loss such that neighbouring Gaussians have a similar vector $\boldsymbol{v}$. During inference, we apply a simple "Embedding-to-Label" process to the rendered vector $\boldsymbol{\mathcal{V}}$ to get the class label.

**Rendering of learnable vector-embeddings.** Each 3D Gaussian has a vector embedding $\boldsymbol{v}$ for instance or semantic segmentation. Unlike color, $\boldsymbol{v}$ is view-independent. During training, we initialize $\boldsymbol{v}$ using a normal distribution and render these vector-embeddings in the same way as color.

$$\boldsymbol{\mathcal{V}} = \sum_{i \in T} \boldsymbol{v}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \tag{3}$$

**Contrastive Loss on rendered vector-embeddings ($\boldsymbol{\mathcal{V}}$).** Building upon the rendered vector-embeddings $\boldsymbol{\mathcal{V}}$, we generate a vector map $\mathbb{V} \in \mathbb{R}^{H \times W \times d}$ for a given camera pose $\pi_i$. Corresponding to the camera pose, we also have a ground truth segmentation mask $M_i$. Using $M_i$, we divide the pixel space into $K$ distinct sets, each representing a segment. This creates a partition $\mathcal{O} = \{\Omega_1, \Omega_2, ..., \Omega_K\}$, where each $\Omega_i \cap \Omega_j = \emptyset$ is disjoint from the others. Next, we compute the centroid for each segment as $\boldsymbol{m}_{\boldsymbol{\Omega_i}} = \frac{1}{|\Omega_i|} \sum_{u \in \Omega_i} \mathbb{V}(u)$. The core idea is to apply a contrastive loss that maximizes the similarity between vector-embeddings within the same segment and minimizes the similarity between vector-embeddings from different segments. This clustering loss is then calculated as follows:

$$\mathcal{L}_{\text{cluster}} = \sum_{\Omega_i \in \mathcal{O}} \sum_{u \in \Omega_i} \|\mathbb{V}(u) - \boldsymbol{m}_{\boldsymbol{\Omega_i}}\|_2^2 - \sum_{\substack{\Omega_i, \Omega_j \in \mathcal{O} \\ i \neq j}} \left\|\boldsymbol{m}_{\boldsymbol{\Omega_i}} - \boldsymbol{m}_{\boldsymbol{\Omega_j}}\right\|_2^2 \tag{4}$$

**Triplet Loss.** To cluster the vector embeddings $\boldsymbol{\mathcal{V}}$, we use the clustering loss that measures the similarity between two vector-embeddings. However, this distance metric does not guarantee a consistent penalty, as detailed in the supplementary material. To address this, we transform $\mathbb{V}$ by passing it through a linear layer, parametrized by $\mathbb{W}^{d \times d}$, resulting in a transformed vector $\boldsymbol{z} = \mathbb{W}\boldsymbol{\mathcal{V}}$. Before passing through linear layer, we pass $\boldsymbol{\mathcal{V}}$ through a sigmoid layer to restrict the range to $[0, 1]$. Instead of clustering $\boldsymbol{z}$ directly, we apply a triplet loss. This loss further reduces variance within clusters and maximizes the margin between different clusters. Our triplet sampling strategy is straightforward. For each pixel


Figure 5: **Qualitative comparison of our method with Contrastive lift on scenes from ScanNet dataset.** Regions, where Contrastive lift performs poorly, are highlighted with red boxes.

$u_1, u_2 \in \Omega_i$, we select vector-embedding $a = \mathbb{W}\sigma(\mathbb{V}(u_1))$ as an anchor. We then randomly choose a positive sample $p = \mathbb{W}\sigma(\mathbb{V}(u_2))$ and a negative sample $n = \mathbb{W}\sigma(\mathbb{V}(u_3))$ from a different segment $u_3 \in \Omega_j$, where $i \neq j$ and $\sigma(.)$ is sigmoid operator. To enhance cluster separation, we select positive and negative samples from segment boundaries. After iterating through all pixels, we obtain a set of triplets $\Delta$, which we use to enforce the triplet loss as defined in Eq. 5 where $\delta$ is the margin. Note that the triplets are obtained in the projected space through linear layer mentioned earlier.

$$\mathcal{L}_{triplet} = \sum_{(a,p,n)\in\Delta} max(0, ||a-p||_2^2 - ||a-n||_2^2 + \delta) \quad (5)$$

**3D neighborhood Regularization.** To encourage nearby Gaussians to share similar embeddings, we apply a spatial smoothness prior in the 3D space. For each Gaussian primitive $i$ with center $\mu_i$, we construct a 3D neighborhood set $\mathcal{N}(i) = \{\|\mu_i - \mu_j\|_2^2 \leq \tau; j \neq i;\ i, j \in \{1, ..., |\mathcal{G}|\}\}$, where $\tau$ is a threshold which selects only those Gaussians that are physically close and $|\mathcal{G}|$ are the number of Gaussian primitives in 3DGS model $\mathcal{G}$. We then enforce embedding consistency only within these neighborhoods by penalizing the difference between their embedding vectors:

$$\mathcal{L}_{3D} = \sum_{i=1}^{|\mathcal{G}|} \sum_{j\in\mathcal{N}(i)} \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2 \quad (6)$$

We apply this loss after 15000 iterations, once the adaptive density control mechanism has stabilized.
Total loss function for this optimization problem is:

$$\begin{aligned}\mathcal{L}_{total} = \mathcal{L}_{rendering} + \lambda_{cluster}\mathcal{L}_{cluster} \\ +\lambda_{triplet}\mathcal{L}_{triplet} + \lambda_{3D}\mathcal{L}_{3D}\end{aligned} \quad (7)$$

For better stability, we apply $\mathcal{L}_{triplet}$ and $\mathcal{L}_{3D}$ after the adaptive density control step. We use $\lambda_{cluster}$, $\lambda_{triplet}$, $\lambda_{3D}$, $\delta$ and $\tau$ as $1e^{-1}$, $1e^{-1}$, $1e^{-1}$, 1, and $1e^{-2}$, respectively.
**Embedding-to-label.** We first convert the rendered embedding $\boldsymbol{\mathcal{V}}$ by applying a sigmoid function $\hat{\boldsymbol{\mathcal{V}}} = \sigma(\boldsymbol{\mathcal{V}})$. We then threshold it to convert into a binary vector $\tilde{\boldsymbol{\mathcal{V}}} = 1[\hat{\boldsymbol{\mathcal{V}}} > \tau]$, where $\tau = 0.5$. We obtain discrete label by mapping $\tilde{\boldsymbol{\mathcal{V}}}$ into a label $l = \sum_k \tilde{\boldsymbol{\mathcal{V}}}_k\, 2^{k-1}$. Further details and rationale are provided in the supplementary material.

## Experiments

**Implementation Details** We build our method on the 3DGS (Kerbl et al. 2023) and train it for $30k$ iterations on a single RTX A6000 GPU. The model is optimized using the ADAM optimizer (Kingma and Ba 2014) with a learning rate of $1e^{-4}$ and the loss defined in Eq. 7. We exclude gradients from the segmentation losses (Eqs. 4, 5, 6) from the 3DGS "Adaptive Density Control" step. For our experiments, we set the embedding dimension ($d$) to 12 and use a maximum of $3,000$ triplets.
**Dataset and Baselines** We benchmark our results on ScanNet (Dai et al. 2017) and Replica3D (Straub et al. 2019) datasets using the same scenes as Contrastive-Lift. We also evaluate on Messy-Rooms dataset, which contains scenes with 25 to 500 objects. We compare our method with Contrastive-Lift (Bhalgat et al. 2023), Panoptic-Lifting (Fu et al. 2022), Panoptic-NeRF (Kundu et al. 2022), Gaussian-Grouping (Ye et al. 2023) and Unified-Lift (Zhu et al. 2025). We follow the Contrastive-Lift evaluation protocol for ScanNet and Replica.
**Metrics.** We evaluate label accuracy using mean intersection over Union (mIoU) and label consistency using scene-level Panoptic Quality ($PQ^{scene}$). $PQ^{scene}$ (Siddiqui et al. 2023) extends Panoptic Quality (PQ) (Kirillov et al. 2019b) by performing matching at the scene level: predicted and ground-truth segments of the same class are merged across views, and pairs with IoU $>$ 0.5 are treated as valid matches. We additionally learn a semantic embedding per primitive alongside the instance embedding, and rasterize both to obtain instance and semantic labels to compute $PQ^{scene}$.
**Quantitative Results** Tab. 2 shows quantitative comparison on ScanNet and Replica3D dataset. We follow the setup described in Contrastive-Lift. We observe that our method outperforms other methods in both datasets. Our approach shows a significant gain of nearly 10 points for the Replica3D datasets. We evaluate our method on the Messy-Rooms dataset (Tab. 1). This dataset consists of a large number of objects, allowing us to assess the scalability of our approach. Our method outperforms Contrastive-Lift in 6 out of 8 scenes within this dataset, demonstrating its effectiveness. Further, our method also outperforms the recent method Gaussian-Grouping (Ye et al. 2023) on both Scan-

| Method | Old Room environment | | | | Large Corridor environment | | | | Mean (%) |
|---|---|---|---|---|---|---|---|---|---|
| | 25 Objects | 50 Objects | 100 Objects | 500 Objects | 25 Objects | 50 Objects | 100 Objects | 500 Objects | |
| Panoptic-Lifting (Siddiqui et al. 2023) | 73.2 | 69.9 | 64.3 | 51.0 | 65.5 | 71.0 | 61.8 | 49.0 | 63.2 |
| Contrastive-Lift (Bhalgat et al. 2023) | 78.9 | 75.8 | 69.1 | 55.0 | 76.5 | 75.5 | 68.7 | 52.5 | 69.0 |
| OmniSeg3D-GS (Ying et al. 2024) | 80.1 | 72.4 | 61.4 | 46.8 | 74.9 | **79.6** | 63.9 | 48.5 | 66.0 |
| Unified-Lift (Zhu et al. 2025) | 79.1 | 72.2 | 65.9 | 53.9 | **77.0** | 78.9 | 70.7 | 54.1 | 69.0 |
| **Ours** | **86.0** | **79.1** | **70.8** | **57.4** | 76.5 | 72.3 | **73.0** | **56.7** | **71.5** |

Table 1: **Messy-Rooms dataset.** $PQ^{scene}$ metric is reported and best results are marked in **bold**. Results for the baseline methods are sourced from (Zhu et al. 2025). We observe that our method outperforms the baseline methods in 6 out of 8 scenes.

| **Method** | **PNF +GT Boxes** | **Panoptic-Lifting** | **Contrastive-Lift** | **Gaussian-Grouping** | **Ours** |
|---|---|---|---|---|---|
| **ScanNet** | 54.3 | 58.9 | 62.3 | 61.83 | **63.0** |
| **Replica3D** | 52.5 | 57.9 | 59.1 | 66.52 | **88.7** |

Table 2: Results on ScanNet and Replica datasets. We report the $PQ^{scene}$ metric and source these values from (Bhalgat et al. 2023). We observe that our method outperforms baselines in both the datasets. Our method achieves a $1.3\times$ higher $PQ^{scene}$ (88.7 vs. 66.52) than Gaussian Grouping.

| **Method** | Panoptic Lifting | Contrastive Lift | Ours |
|---|---|---|---|
| **Training** | $> 20\,hrs$ | $> 15\,hrs$ | $< 40\,mins$ |

Table 3: Training time comparison on NVIDIA A6000 for a scene in Replica dataset. Our method exhibits a significant training time advantage for lifting 2D segmentation masks to 3D over other methods.

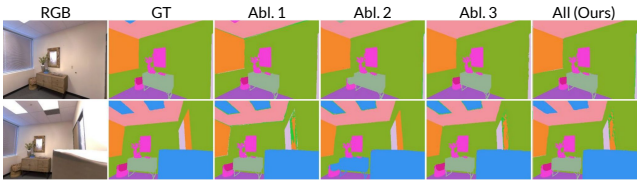


Figure 6: Qualitative results for different design choices discussed in ablation studies in Tab. 4.

Net and Replica3D dataset.

**Qualitative Results.** We compare our method with Contrastive-Lift in Fig. 5 and 7. We can observe that our method generates accurate masks compared to the baseline. For example, in Fig. 5 Contrastive-Lift consistently fails to predict the leg of the highlighted chair whereas our method consistently predicts the leg of the chair. Further, we provide more qualitative results in the supplementary material.

**Time-performance comparison.** Current methods for lifting 2D segmentation masks to 3D often suffer from slow training speeds due to complex, multi-stage pipelines. For instance, Panoptic-Lifting (Siddiqui et al. 2023) relies on time-consuming linear assignment to resolve mask inconsistencies, while Contrastive-Lift (Bhalgat et al. 2023) employs a computationally intensive clustering algorithm for post-processing. In contrast, our unified, end-to-end approach eliminates these costly intermediate steps, leading to significantly faster training, as shown in Tab. 3. To ensure a fair evaluation, we also compare against a stronger baseline, Contrastive-Lift-3DGS, which uses the same 3DGS representation as our method and $\mathcal{L}_{cluster}$ loss followed by HDBSCAN clustering. We observe that our approach is faster than this enhanced baseline as well (Tab. 5, with further analysis provided in the supplementary material.

| **Contrastive-Loss (CL)** | ✓ | ✓ | ✓ | ✓ | ✓ |
|---|---|---|---|---|---|
| **Triplet Loss** | × | × | ✓ | ✓ | ✓ |
| **3D Regularization** | ✓ | × | × | ✓ | ✓ |
| **MLP** | × | × | ✓ | × | ✓ |
| | **Abl. 1** | **Abl. 2** | **Abl. 3** | **w/o MLP** | **with MLP** |
| $PQ^{scene}$ | 88.0 | 83.7 | 89.0 | 88.0 | 89.0 |
| **mIoU** | 94.4 | 91.8 | 95.2 | 94.0 | 95.4 |

Table 4: Effect of different loss functions. Abl.1 uses "CL + 3D Regularization" loss, Abl.2 uses only Contrastive loss and Abl.3 uses "CL+Triplet loss with MLP". We observe that a combination of Contrastive loss, Triplet loss and 3D neighborhood loss yields the best results for our method.

| **Method** | **3DGS Time** ↓ | **Cluster Time** ↓ | **Total Time** ↓ | **$PQ^{scene}$** ↑ | **IoU** ↑ |
|---|---|---|---|---|---|
| CL+3DGS | 42.2 m | 43.21 m | 85.41 m | 94.6 | 97.4 |
| Ours | **42 m** | **0 m** | **42 m** | 94 | 96.2 |

Table 5: Quantitative comparison of our method with Contrastive-Lift (CL) + 3DGS

## Ablations Study

**Contrastive Losses, 3D Loss** Our method uses a combination of 3D neighborhood loss, contrFastive loss and triplet loss to optimize the feature field for instance segmentation purpose. We evaluate the impact of these losses. Tab. 4 and Fig. 6 shows results when these losses are applied separately. We also observe that applying the triplet loss on MLP-projected embeddings yields better separation and improves the overall metrics. We observe that when we use a combination of the three losses, we obtain the best results.

**Number of segmentation masks during training** We evaluated our method's robustness using a pre-trained 3DGS model on the Messy-Rooms dataset with only $5\%$, $10\%$ and $20\%$ of segmentation masks. Fig. 9 shows that the results achieve accuracy comparable to training with all masks,

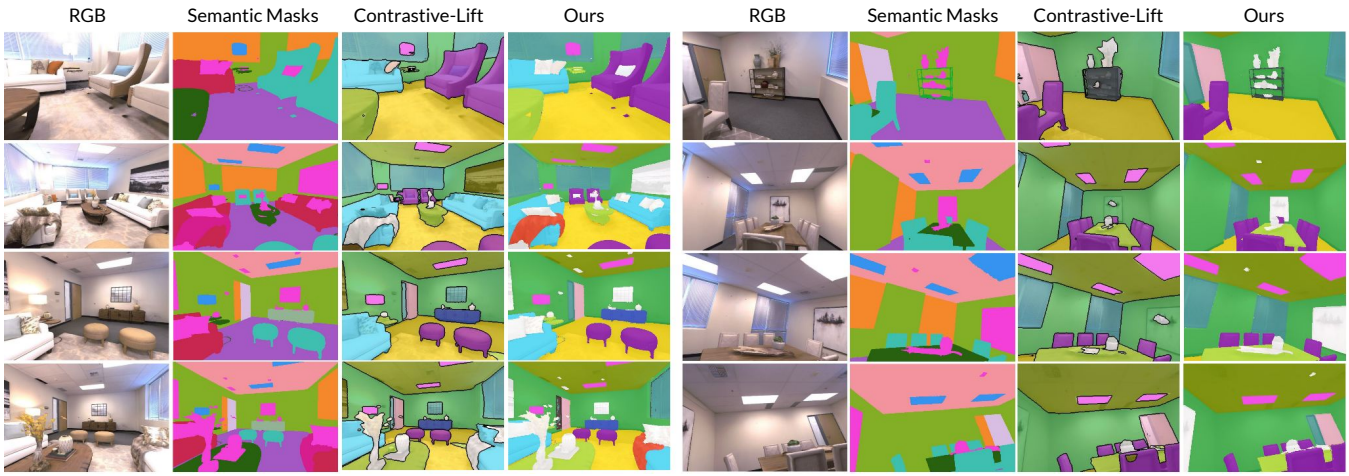


Figure 7: **Comparison with Contrastive lift on Replica3D dataset.** Observe that Contrastive-Lift fails to capture some objects such as the pillows on the sofa set and the set of objects placed inside the mini-cupboard

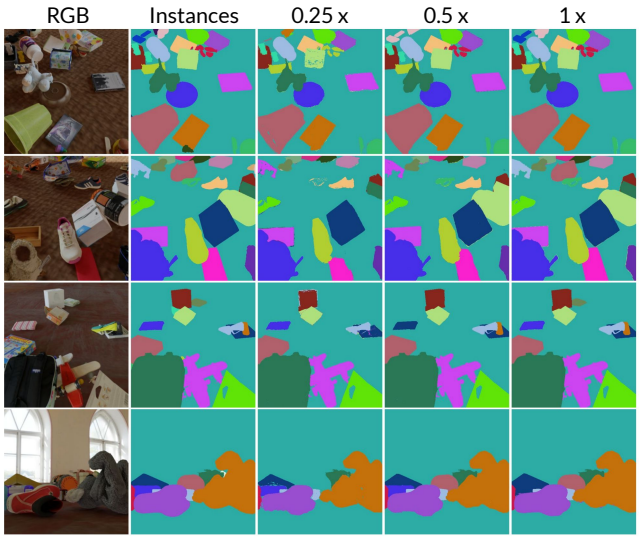


Figure 8: **Qualitative results on Messy-Rooms dataset by varying resolution of the segmentation masks.** We show results with $0.25\times$, $0.5\times$ and $1\times$(full) resolution

demonstrating the method's effectiveness with limited data. We provide implementation details for this ablation in the supplementary material.

**Input resolution of the segmentation masks** This experiment investigates the impact of input segmentation mask resolution. We evaluate our method's ability to handle downscaled masks ($0.25\times$ and $0.5\times$) on four scenes from the Messy-Rooms dataset (Fig. 8). Visual inspection reveals no significant difference between results obtained with $0.5\times$and full-resolution masks. Training at a lower resolution will accelerate our method without compromising accuracy. We provide implementation details for this ablation in the supplementary material.

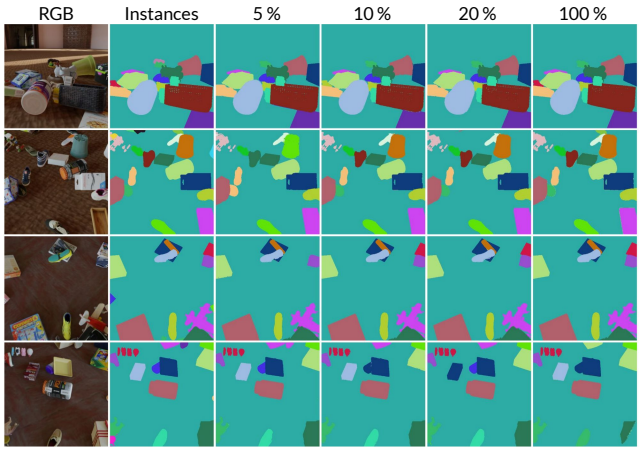


Figure 9: **Qualitative results on Messy-Rooms dataset by varying percentage of training data.** We show results with 5, 10, 20 and 100(full) % training samples.

## Conclusion and Future Work

In this work, we introduce UniC-Lift, a single-stage framework for lifting inconsistent 2D multi-view instance masks into a 3D representation. Unlike previous two-stage methods that rely on computationally expensive post-processing such as clustering, our approach directly decodes class labels from learned vector embeddings. These embeddings are optimized with a triplet loss function to reduce intra-class variance. Experiments demonstrate that UniC-Lift outperforms state-of-the-art methods while significantly reducing training time. This efficiency and accuracy enable practical applications such as interactive scene editing and mesh extraction, while its robust performance in complex, multi-object environments confirms its scalability. Future work focuses on extending UniC-Lift to dynamic scenes, large-scale unbounded scenes and hierarchical segmentation for a 3D representation.

## Acknowledgments

This work was partly supported by KIAC, IISc. We thank Rishubh Parihar for reviewing the manuscript and providing insightful feedback.

## References

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV 2021*, 5855–5864.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR 2022*, 5470–5479.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV 2023*, 19697–19705.

Bartolomei, L.; Teixeira, L.; and Chli, M. 2020. Perception-aware Path Planning for UAVs using Semantic Segmentation. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5808–5815.

Bhalgat, Y.; Laina, I.; Henriques, J. F.; Zisserman, A.; and Vedaldi, A. 2023. Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion. In *NIPS 2023*.

Bing, W.; Chen, L.; Yang, B.; and Yang, B. 2023. DM-NeRF: 3D Scene Geometry Decomposition and Manipulation from 2D Images. In *The Eleventh International Conference on Learning Representations*.

Brostow, G. J.; Fauqueur, J.; and Cipolla, R. 2009. Semantic object classes in video: A high-definition ground truth database. *Pattern recognition letters*, 30(2): 88–97.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *ICCV 2021*, 9650–9660.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *ECCV 2022*, 333–350. Springer.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *CVPR 2022*, 1290–1299.

Choy, C. B.; Gwak, J.; and Savarese, S. 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. *CVPR 2019*, 3070–3079.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR 2016*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR 2017*, 5828–5839.

Dhiman, A.; Srinath, R.; Rangwani, H.; Parihar, R.; Boregowda, L. R.; Sridhar, S.; and Babu, R. V. 2023. Strata-NeRF: Neural Radiance Fields for Stratified Scenes. In *ICCV 2023*, 17603–17614.

Dhiman, A.; Srinath, R.; Sarkar, S.; Boregowda, L. R.; and Babu, R. V. 2025. ChromaDistill: Colorizing Monochrome Radiance Fields with Knowledge Distillation. In *WACV 2025*, 2400–2410. IEEE.

Feng, D.; Haase-Schuetz, C.; Rosenbaum, L.; Hertlein, H.; Duffhauss, F.; Gläser, C.; Wiesbeck, W.; and Dietmayer, K. C. J. 2019. Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22: 1341–1360.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *CVPR 2022*, 5501–5510.

Fu, X.; Zhang, S.; Chen, T.; Lu, Y.; Zhu, L.; Zhou, X.; Geiger, A.; and Liao, Y. 2022. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *3DV 2022*, 1–11. IEEE.

He, J.; Li, P.; Geng, Y.; and Xie, X. 2023. FastInst: A simple query-based model for real-time instance segmentation. In *CVPR 2023*, 23663–23672.

Heo, M.; Hwang, S.; Hyun, J.; Kim, H.; Oh, S. W.; Lee, J.-Y.; and Kim, S. J. 2023. A generalized framework for video instance segmentation. In *CVPR 2023*, 14623–14632.

Hu, J.; Huang, L.; Ren, T.; Zhang, S.; Ji, R.; and Cao, L. 2023. You Only Segment Once: Towards Real-Time Panoptic Segmentation. *CVPR 2023*, 17819–17829.

Huang, Y.-H.; He, Y.; Yuan, Y.-J.; Lai, Y.-K.; and Gao, L. 2022. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *CVPR 2022*, 18342–18352.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4): 1–14.

Kerr, J.; Kim, C. M.; Goldberg, K.; Kanazawa, A.; and Tancik, M. 2023. Lerf: Language embedded radiance fields. In *ICCV 2023*, 19729–19739.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirillov, A.; Girshick, R. B.; He, K.; and Dollár, P. 2019a. Panoptic Feature Pyramid Networks. *CVPR 2019*, 6392–6401.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P. 2019b. Panoptic segmentation. In *CVPR 2019*, 9404–9413.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023a. Segment Anything. *arXiv:2304.02643*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023b. Segment anything. In *ICCV*, 4015–4026.

Kobayashi, S.; Matsumoto, E.; and Sitzmann, V. 2022. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35: 23311–23330.

Kundu, A.; Genova, K.; Yin, X.; Fathi, A.; Pantofaru, C.; Guibas, L. J.; Tagliasacchi, A.; Dellaert, F.; and Funkhouser, T. 2022. Panoptic neural fields: A semantic object-aware neural scene representation. In *CVPR 2022*, 12871–12881.

Lan, K.; Li, H.; Shi, H.; Wu, W.; Liao, Y.; Wang, L.; and Zhou, P. 2023. 2d-guided 3d gaussian segmentation. *arXiv preprint arXiv:2312.16047*.

Li, B.; Weinberger, K. Q.; Belongie, S.; Koltun, V.; and Ranftl, R. 2022. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*.

Li, J.; Zhang, J.; Bai, X.; Zheng, J.; Ning, X.; Zhou, J.; and Gu, L. 2024. DNGaussian: Optimizing Sparse-View 3D Gaussian Radiance Fields with Global-Local Depth Normalization. *arXiv preprint arXiv:2403.06912*.

Li, M.; Liu, S.; and Zhou, H. 2024. SGS-SLAM: Semantic Gaussian Splatting For Neural Dense SLAM. *arXiv preprint arXiv:2402.03246*.

Li, X.; and Loy, C. C. 2018. Video Object Segmentation with Joint Re-identification and Attention-Aware Mask Propagation. In *ECCV 2018*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; and Dollár, P. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312.

Liu, M.; Zhu, Y.; Cai, H.; Han, S.; Ling, Z.; Porikli, F. M.; and Su, H. 2022. PartSLIP: Low-Shot Part Segmentation for 3D Point Clouds via Pretrained Image-Language Models. *CVPR 2023*, 21736–21746.

Liu, Y.; Hu, B.; Huang, J.; Tai, Y.-W.; and Tang, C.-K. 2023. Instance neural radiance field. In *ICCV 2023*, 787–796.

Lu, T.; Dhiman, A.; Srinath, R.; Arslan, E.; Xing, A.; Xiangli, Y.; Babu, R. V.; and Sridhar, S. 2024. Turbo-GS: Accelerating 3D Gaussian Fitting for High-Quality Radiance Fields. arXiv:2412.13547.

McInnes, L.; Healy, J.; and Astels, S. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11): 205.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Mirzaei, A.; Kant, Y.; Kelly, J.; and Gilitschenski, I. 2022. Laterf: Label and text driven object radiance fields. In *ECCV 2022*, 20–36. Springer.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Qin, M.; Li, W.; Zhou, J.; Wang, H.; and Pfister, H. 2023. LangSplat: 3D Language Gaussian Splatting. *arXiv preprint arXiv:2312.16084*.

Shi, J.-C.; Wang, M.; Duan, H.-B.; and Guan, S.-H. 2023. Language embedded 3d gaussians for open-vocabulary scene understanding. *arXiv preprint arXiv:2311.18482*.

Siddiqui, Y.; Porzi, L.; Buló, S. R.; Müller, N.; Nießner, M.; Dai, A.; and Kontschieder, P. 2023. Panoptic lifting for 3d scene understanding with neural fields. In *CVPR 2023*, 9043–9052.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*.

Tschernezki, V.; Laina, I.; Larlus, D.; and Vedaldi, A. 2022. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *3DV 2022*, 443–453. IEEE.

Vora, S.; Radwan, N.; Greff, K.; Meyer, H.; Genova, K.; Sajjadi, M. S.; Pot, E.; Tagliasacchi, A.; and Duckworth, D. 2021. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. *arXiv preprint arXiv:2111.13260*.

Wu, R.; Mildenhall, B.; Henzler, P.; Park, K.; Gao, R.; Watson, D.; Srinivasan, P. P.; Verbin, D.; Barron, J. T.; Poole, B.; and Holynski, A. 2023. ReconFusion: 3D Reconstruction with Diffusion Priors. *arXiv*.

Xu, J.; Xiong, Z.; and Bhattacharyya, S. P. 2023. PIDNet: A real-time semantic segmentation network inspired by PID controllers. In *CVPR 2023*, 19529–19539.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2023. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. *arXiv preprint arXiv:2309.13101*.

Ye, M.; Danelljan, M.; Yu, F.; and Ke, L. 2023. Gaussian grouping: Segment and edit anything in 3d scenes. *arXiv preprint arXiv:2312.00732*.

Ying, H.; Yin, Y.; Zhang, J.; Wang, F.; Yu, T.; Huang, R.; and Fang, L. 2024. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *CVPR 2024*, 20612–20622.

Yu, H.-X.; Guibas, L. J.; and Wu, J. 2021. Unsupervised discovery of object radiance fields. *arXiv preprint arXiv:2107.07905*.

Zhang, Y.; Li, L.; Wang, W.; Xie, R.; Song, L.; and Zhang, W. 2023. Boosting Video Object Segmentation via Space-Time Correspondence Learning. *CVPR 2023*, 2246–2256.

Zhi, S.; Laidlow, T.; Leutenegger, S.; and Davison, A. J. 2021. In-place scene labelling and understanding with implicit scene representation. In *ICCV 2021*, 15838–15847.

Zhou, S.; Chang, H.; Jiang, S.; Fan, Z.; Zhu, Z.; Xu, D.; Chari, P.; You, S.; Wang, Z.; and Kadambi, A. 2023a. Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields. *CVPR 2024*, 21676–21685.

Zhou, Z.; Lei, Y.; Zhang, B.; Liu, L.; and Liu, Y. 2023b. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *CVPR 2023*, 11175–11185.

Zhu, R.; Qiu, S.; Liu, Z.; Hui, K.-H.; Wu, Q.; Heng, P.-A.; and Fu, C.-W. 2025. Rethinking end-to-end 2d to 3d scene segmentation in gaussian splatting. In *CVPR 2025*, 3656–3665.

Zou, X.; Yang, J.; Zhang, H.; Li, F.; Li, L.; Wang, J.; Wang, L.; Gao, J.; and Lee, Y. J. 2024. Segment everything everywhere all at once. *NIPS 2024*, 36.