Pano3D: Unified 3D Reconstruction and Panoptic Segmentation

Victor Barberteguy[1,2], Ahmet Iscen[1], Mathilde Caron[1], Alireza Fathi[1], Gül Varol[2], and Cordelia Schmid[1]

[1] Google DeepMind
[2] LIGM, École des Ponts, IP Paris, Univ Gustave Eiffel, CNRS
victorbbt@google.com
https://victorbbt.github.io/Pano3D

Abstract. Recent advances in 3D feedforward reconstruction neural networks have achieved remarkable success in dense reconstruction from images without any camera parameters. Yet, equipping these models with robust semantic understanding remains an open problem. Here we introduce an approach that performs 3D reconstruction and 3D panoptic segmentation in a unified framework. We build on existing 3D reconstruction models and augment them with a set-based mask decoder. The approach is jointly trained with a geometric and semantic loss, which are shown to be mutually beneficial. More precisely, the features are initialized from the geometric information and then finetuned to capture jointly geometry and semantics. We demonstrate the generality of our approach by successfully applying our framework both to online and all-to-all attention reconstruction backbones. Our method achieves state-of-the-art performance in 3D panoptic segmentation across ScanNet, ScanNet200, and ScanNet++ datasets. Ablation studies show that such joint training of a unified model equips 3D feedforward reconstruction neural networks with panoptic segmentation and yields mutually beneficial improvements.

Keywords: 3D Panoptic Segmentation · Scene Reconstruction

1 Introduction

Holistic 3D scene understanding requires jointly reasoning about "where" things are (geometry) and "what" they are (semantics). Recent feedforward reconstruction models (FRM), such as DUSt3R [38], MASt3R [21], MUSt3R [1], and VGGT [37], have demonstrated that transformers can reconstruct dense 3D pointmaps from unposed RGB images, opening new opportunities for unified 3D scene understanding. Several approaches [9,14,20,36,41] have started adding semantic capabilities to FRMs, either by freezing the backbone and relying on external 2D features [28,41,45], or by adding dense contrastive heads that require unsupervised clustering at inference [20,23,26,31].

In this paper, we introduce a unified reconstruction and segmentation approach, **Pano3D**, that eliminates the heuristic clustering algorithms in favor of a natively differentiable, query-based panoptic head appended to a multi-view reconstruction

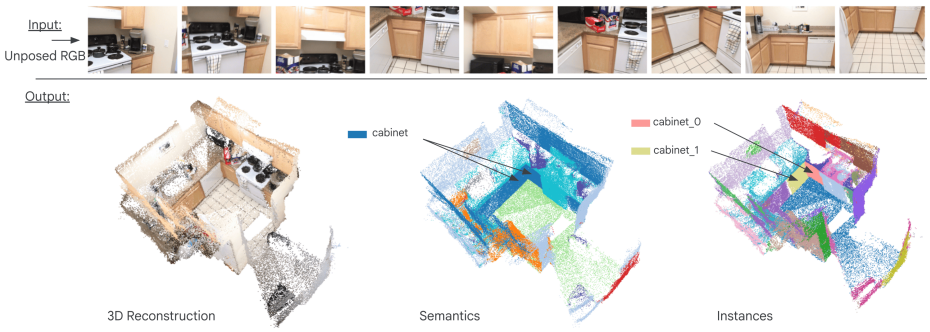

Fig. 1: Pano3D inputs and outputs: Our model reconstructs a 3D point cloud (left) from unposed RGB frames, and predicts semantic classes (middle) as well as semantic instance masks (right). Pano3D can jointly reconstruct a scene as well as detect instances and their classes.

backbone. Our approach is built on the simple insight that recognizing objects is implicitly encoded within feedforward reconstruction models. Indeed, these models contain 2D, monocular information in the encoder stage, and iteratively refine the representation to detect spatial boundaries and 3D structural coherence in the cross-view decoder, which, once combined, form a strong representation of the objects. This makes Pano3D alignment-free: we directly predict instance-aware masks and class logits from learned queries in a feedforward manner, without relying on external 2D models or postprocessing.

Our approach appends a set-based mask decoder directly to the output of the FRM. The decoder is similar to Mask2Former [5] in the 2D image segmentation field. By simply fusing the features from the encoder with the final cross-view decoder features we build features that are both precise and scene-aware, allowing the queries of the mask decoder to predict consistent masks across the entire input sequence. Rather than freezing the network or crafting explicit consistency losses, we finetune the multi-view decoder on both 3D reconstruction and panoptic segmentation jointly. We demonstrate that this setting gives increased instance and semantic segmentation results while maintaining the reconstruction capability. Fig. 1 illustrates the outputs from our model on an example input sequence: 3D reconstruction on the left, 3D semantic segmentation in the middle, and 3D semantic instance segmentation on the right. Our core contributions are the following:

- **Unified model:** We tackle 3D reconstruction and panoptic segmentation from unposed RGB views relying only on pretrained FRMs (see Fig. 2). Without relying on auxiliary 2D models and heuristic post-processing, Pano3D achieves state-of-the-art results on ScanNet [6], ScanNet200 [34] and ScanNet++ [43] purely through feature adaptation.
- **Backbone generality:** We apply our method to two different types of FRMs, namely MUSt3R [1] and Pi3 [39], demonstrating the adaptability of our method across online reconstruction models [1] and all-to-all attention models [39].
- **Joint training:** We demonstrate that finetuning the geometry decoder, rather than freezing it, improves segmentation without compromising geometric fidelity. Furthermore, we highlight the importance of gradient scaling in balancing these two objectives.

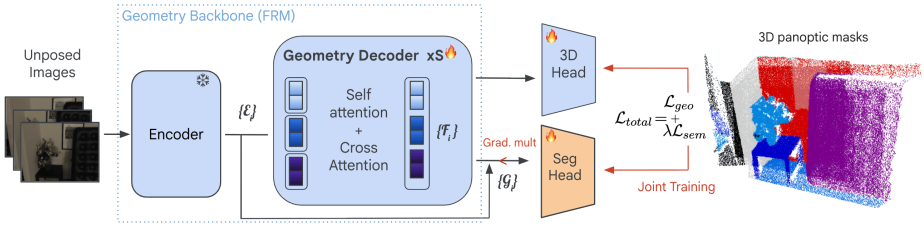

Fig. 2: Framework overview: Our method appends a segmentation head inspired from Mask2Former [5] on top of a feedforward reconstruction model. We extract features from the monocular encoder and the multi-view decoder of the FRM to predict 3D points as well as consistent masks across the input sequence. We jointly optimize the geometry decoder to bake in object understanding.

2 Related Work

2D Panoptic Segmentation. Panoptic segmentation [18] consists in predicting a dense label map with instances and semantic labels for countable objects ('thing' classes: chair, water bottle, ...), and semantic labels only for 'stuff' classes (sky, floor, ...). Early approaches [4,12] extended CNN architectures by adding specialized heads. More recent transformer-based [7] set-prediction methods, notably Mask2Former [5], use bipartite matching over learnable object queries [2] to jointly predict thing and stuff classes. While successful in 2D, these methods lack spatial consistency across multiple views.

Feedforward Reconstruction Models. Recent models such as DUSt3R [38] and MASt3R [21] have redefined 3D reconstruction as a dense pointmap regression problem for pairs of unposed views. This paradigm naturally led to multi-view extensions: VGGT [37] and Pi3 [39] use all-to-all cross-attention for accurate but computationally expensive reconstruction, while MUSt3R [1] introduces a memory mechanism for online, causal prediction. Our work builds upon these geometric backbones, building on the insight that their progressive refinement from monocular to multi-view features naturally encodes spatial discontinuities suitable for learning object instances.

Unified 3D Semantic Transformers.

Prior 3D panoptic methods operate on explicit representations such as raw point clouds from RGBD sensors as in ODIN [15] and DeepViewAgg [33], or postprocessed meshes as in OpenScene [29], while others rely on implicit representations (NeRF [27], Gaussians [16]) where semantic priors are learnt or distilled [9,19,44]. Building on FRMs, a new generation of feedforward models unifies reconstruction and semantics, often relying on CLIP features [30] lifted in 3D [11,14] or augmented with Gaussian heads [8,36]. UNITE [20] adds dense DPT heads [31] on VGGT [37] but depends on knowledge distillation and explicit consistency losses. Both IGGT [23] (which uses SAM2 [32] features) and UNITE require unsupervised clustering (HDBSCAN [26]) at inference, often yielding coarse masks. PanSt3R [45] places a Mask2Former-style [4] decoder on a frozen MUSt3R backbone with DINOv2 [28] features and QUBO post-processing, but lacks an integrated framework for joint reconstruction and segmentation. SIU3R [41] branches geometry and semantics early, requiring hand-crafted regularizations to enable inter-task benefits [3]. **Pano3D** builds on these works to design a uni-

fied framework trained jointly with geometric and semantic losses. By employing a set-based query decoder on top of the geometry network, we avoid contrastive clustering and demonstrate that explicit consistency losses and heuristic post-processing are unnecessary. Our approach is generic and can be applied to various geometry backbones.

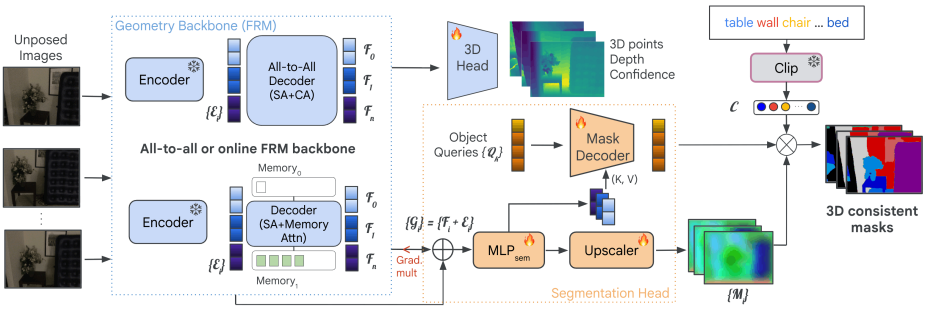

Fig. 3: Model architecture: For a given input sequence $\mathcal{I} = \{I_i\}$, we extract features from the geometry encoder $\mathcal{E}_i$ and decoder $\mathcal{F}_i$ to create frame features $\mathcal{G}_i$ and mask features $\mathcal{M}_i$. The blue components (encoder, geometry blocks, 3D head) are from the geometry backbone (either it uses a memory-based architecture [1] with causal memory attention or all-to-all attention [39]). Learnable object queries $\mathcal{Q}_k$ are refined in the mask decoder by attending to $\mathcal{G}_i$. Then, they perform dot product against mask features $\mathcal{M}_i$ and a matrix of frozen text embeddings $\mathcal{C}$ to predict semantic instance masks. Crucially, one object query predicts a set of 3D-consistent masks on the full input sequence. $\mathcal{F}_i$ are at the same time used to predict 3D information (pointmaps in reference frame's coordinate system, local pointmaps and confidence) with a linear head. As masks and pointmaps are pixel-aligned, we lift these masks to obtain 3D instance segmentation. We train the cross-view decoder on both the geometry and segmentation tasks.

3 Methodology

We introduce Pano3D, a unified transformer framework for 3D reconstruction and segmentation, see Fig. 3. Given an unordered collection of N unposed RGB images $\mathcal{I} = \{I_1,...,I_N\}$, our goal is to jointly recover the dense scene geometry (Sec. 3.1) and a set of multi-view consistent 3D semantic instance masks (Sec. 3.2) with joint finetuning of the geometry decoder (Sec. 3.3). We instantiate our framework by building on MUSt3R [1] (memory-based online FRM in Fig. 3) or on Pi3 [39] (all-to-all FRM in Fig. 3) to extract features at the encoder and decoder stages (Sec. 3.4). The features from this encoder-decoder setup are then trained to both regress 3D information (3D head in Fig. 3) and to learn consistent object queries with a mask decoder similar to [15]. The predicted masks and pointmaps are pixel-aligned (i.e., direct correspondence between a 3D predicted point and the masks), which makes the lifting to 3D straightforward.

3.1 3D Reconstruction

Our unified framework is built on top of a feedforward reconstruction model (FRM), which we briefly introduce in this section. Given a collection of N unposed RGB

images $\mathcal{I}=\{I_1,...,I_N\}$, the objective of the geometric backbone is to predict a dense pointmap $X_i \in \mathbb{R}^{H\times W\times 3}$ and an associated confidence map $C_i \in [0,\infty)$ for each frame I_i in a globally aligned coordinate system. For an input image, patch features E_i are first extracted using a pretrained monocular 2D encoder (e.g., CrocoV2 [40] or DINOv2 [28]). Following the training strategy detailed in Sec. 4.1, we keep this encoder frozen as we observe in our ablations (Sec. 4.4) that finetuning end-to-end has a minor impact while increasing the training cost. These tokens are subsequently fed into a multi-view transformer decoder Φ_{geo}, which aggregates information across viewpoints to output cross-view consistent geometric latent features F_i:

$$F_i = \Phi_{\text{geo}}(E_i, \mathcal{C}_{\text{context}}), \tag{1}$$

where $\mathcal{C}_{\text{context}}$ represents the multi-view scene context. This setting was first introduced by DUSt3R [38] for pairs of images. This context can be established either via an all-to-all cross-attention mechanism across all frames simultaneously (e.g., Pi3 [39]), or through a sequential memory tracking mechanism for online, streaming prediction (e.g., MUSt3R [1]).

A 3D prediction head then maps these refined latents to the spatial outputs: $\{X_i, C_i\} = \text{Head}_{\text{3D}}(F_i)$. The geometric backbone is optimized using a dense regression objective $\mathcal{L}_{\text{geo}}$:

$$\mathcal{L}_{\text{geo}} = \sum_{i=1}^{N} \sum_{p\in\Omega} C_{i,p} ||X_{i,p} - \hat{X}_{i,p}||_2 - \alpha \log(C_{i,p}), \tag{2}$$

where Ω is the pixel grid and $\hat{X}_{i,p}$ denotes the ground-truth coordinates. The exact form of $\mathcal{L}_{\text{geo}}$ depends on the FRM (e.g., relative cameras and focal depth [39], or pointmaps in a global reference frame [1]); we detail each instantiation in Sec. A of the supplementary. As opposed to most methods integrating semantic understanding into FRMs [20, 45], our framework keeps $\mathcal{L}_{\text{geo}}$ activated and uses it as a geometric objective within our joint finetuning formulation (Eq. 3) to prevent corrupting the reconstruction accuracy while optimizing for segmentation.

3.2 3D Panoptic Segmentation

We hypothesize that the high-dimensional feature space of a geometric backbone trained to align unposed images across viewpoint changes already contains rich structural descriptors. Starting from monocular context, the cross-view decoder progressively recovers geometry and scene appearance [35]. Our approach thus extracts features at different stages of the underlying FRM, resulting in a unified architecture that does not need additional external models.

Geometric Feature Bridge. We introduce a lightweight adaptation module (cf. Fig. 3, MLP_{sem}). For each frame i, we concatenate the features from the frozen encoder E_i with the multi-view consistent features F_i from the finetuned decoder. These are projected through an MLP adapter to produce patch features $\mathcal{G}_i$ for masked cross-attention with the object queries. The encoder, often trained in a self-supervised manner, contains local monocular context whereas the decoder brings scene-level consistency and geometry appearance.

Set Transformers for Panoptic Segmentation. We adopt the set-prediction framework of Mask2Former [5] and extend it to sequences as proposed in ODIN [15]. We initialize K learnable object queries $\mathcal{Q} = \{q_k\}_{k=1}^{K}$ that act as sequence-level detectors; a single q_k tracks a unique 3D instance across all N images. For class predictions, we compute the similarity between query embeddings and a matrix of frozen CLIP [30] text embeddings $\mathcal{C}$ (cf. Fig. 3, right) as in [10, 15, 45]. The semantic loss $\mathcal{L}_{sem}$ follows standard bipartite matching [5]. This allows Pano3D to be queried on any text at inference time, simply by substituting the text matrix with new embeddings. We provide the loss specifics in Sec. A of the supplementary material.
High-Resolution Mask Features. To recover sharp boundaries, we upscale the adapted patch features by a factor of 4 (for patch size 16, it results in one-quarter of the input resolution) using sequential pixel-shuffle operations in the upscaler U_{sem}, resulting in the dense mask features $\mathcal{M}_i$. Because our queries $\mathcal{Q}$ attend to the combination of these features, they are provided with both local appearance and cross-view (scene level) context, enabling consistent 3D segmentation.

Crucially, we do not stop the gradients on the decoder features F_i in the concatenation, providing a direct gradient path from $\mathcal{L}_{sem}$ into the geometry decoder through both the patch features $\mathcal{G}_i$ and the mask features $\mathcal{M}_i$ (Sec. 3.3). Following the Mask2Former method [5], we supervise the object queries at each layer of the mask decoder to have more stable and faster convergence.

3.3 Joint Training

Existing work either freeze the FRM [45] or add a pixel-wise consistency loss to the multi-task learning [20] to prevent corrupting the geometric quality. While FRMs learn precise local, low-level matching, they are explicitly trained to discard any ambiguous prediction in textureless areas or reflective surfaces [38]. We identify that semantic instance segmentation can act as an implicit object prior that mitigates this limitation.

By unfreezing the geometric decoder Φ_{geo} and allowing the semantic gradients from $\mathcal{L}_{sem}$ to flow back from the bipartite matching loss into the geometric features, the network learns to group pixels into discrete entities. Our combined objective is:

$$\mathcal{L}_{total} = \mathcal{L}_{geo} + \lambda \mathcal{L}_{sem}, \tag{3}$$

where λ balances the two tasks. As we demonstrate in our experiments (Sec. 4), this joint optimization yields multi-view consistent masks without the need for heuristic post-processing algorithms such as Quadratic Unconstrained Binary Optimization (QUBO) [45] and without enforcing any explicit prior with hand-crafted consistency losses. We refer to Sec. 4.4 for a study on loss balancing to show that geometry performance is maintained when training on both tasks.
Segmentation Gradient Scaling. Besides the loss balancing, we apply a gradient multiplier during backpropagation to part of the network. Specifically, when training, we first log the per-task gradients and observe that the semantic gradients are typically an order of magnitude higher than the geometric gradients, which could perturb the weights of the multi-view decoder excessively. We therefore measure the scale between the norm of the two gradients when trained with no gradient multiplier, and apply this as a constant scaling factor $\gamma = 0.1$ to the gradients originating

from the segmentation head before they enter the geometric decoder Φ_{geo} during backpropagation (as denoted in Fig. 3 with 'Grad. mult.'). As demonstrated by our ablation studies in Sec. 4.4, while this gradient scaling has a negligible impact on segmentation metrics, it serves as a safeguard for geometry, ensuring improved 3D consistency when jointly finetuning the geometry decoder.

3.4 Adaptation to Different FRMs

Our framework is flexible and can be plugged with memory-based reconstruction models such as MUSt3R [1], as well as all-to-all attention models such as Pi3 [39].
Memory Models. To handle large image collections beyond device memory limits, we leverage the recurrent memory mechanism of MUSt3R [1], but extend it to maintain semantic instance consistency in the queries. Our model is thus equipped with two distinct memories. The first is an object-centric memory of the scene (object queries), and the other one is the original frame memory of MUSt3R (containing the patch tokens). As we finetune the model, the frame tokens learn some object priors on top of the original geometric cues inherited from the pretrained model. The processing happens in two phases. In Phase 1 (Update), the model processes keyframes to build the global geometric memory $C_{context} = \mathcal{M}em_{geo}$. These frame features $\mathcal{G}_i^{update}$ are used to initialize the semantic queries $\mathcal{Q}_k^{update}$ in the mask decoder $\Phi_{panoptic}$. In Phase 2 (Render), the same or new frames attend to $\mathcal{M}em_{geo}$ to grasp the scene context. Then, the queries are updated against the new frames features $\mathcal{G}_i^{render}$ to produce $\mathcal{Q}_k^{render}$. A final dot product between $\mathcal{Q}_k^{render}$ and $\mathcal{M}_i^{render}$ gives the mask predictions:

$$\mathcal{Q}_k^{render} = \Phi_{panoptic}(\mathcal{Q}_k^{update}, \mathcal{G}_i^{render}). \tag{4}$$

As in MUSt3R [1], we supervise both phases during training, but the object queries need particular care. The queries are initialized in the update phase, and the render phase reuses the updated query state. Importantly, we concatenate the mask predictions from both phases along the view axis, and perform a single bipartite matching against the scene-level ground truth. This enforces that a given query $\mathcal{Q}_k$ must detect the same object instance in both the update and render frames, naturally yielding consistent assignments when filling the memory, or when rendering new frames. We provide a detailed figure and the specifics on the training of the memory in Sec. A of the supplementary.

4 Experiments

We first provide implementation details (Sec. 4.1), followed by benchmark descriptions (Sec. 4.2). We then compare against the state of the art (Sec. 4.3) and ablate model components (Sec. 4.4). Finally, we present qualitative results (Sec. 4.5).

4.1 Implementation Details

Architecture. We use the MUSt3R [1] architecture as the online geometry decoder initialized with pretrained weights, and the Pi3 [39] pretrained model for all-to-all geometry model. The mask decoder is a 9-layer decoder initialized from Mask2Former [5].

We use the decoder head trained on top of ResNet50 features [13] on COCO [24] dataset. We use 100 queries for ScanNet [6] and ScanNet200 [34], and 200 queries for ScanNet++ [43].
Training. All models are trained on TPU v6 accelerators using the AdamW optimizer [17]. We use a base image resolution of 384×512 for MUSt3R [1] and 378×518 for Pi3 [39]. For training we use $N = 15$ views. Following standard practice, we apply random color jitter and random stride between 1 and 4 for frame selection.
Loss Weights. We use the Mask2Former [5] loss formulation with weights λ_{cls}, λ_{dice} and λ_{bce} set to 2, 2 and 5 following the original paper. The geometric reconstruction loss is weighted by a factor of 10 relative to the segmentation loss. We scale the segmentation gradients before backpropagation in the geometry decoder by a factor 0.1.
Inference. Unless stated otherwise, we evaluate our online model in its *render phase*. We select up to 20 frames via uniform linear spacing (linspace) to build the geometric memory and initialize object queries. We then perform a second render pass on all evaluation frames to generate dense, consistent predictions. For the all-to-all model, we pass all the frames into the model.

We refer to Sec. A of the supplementary material for more details on the architecture and the training.

4.2 Datasets and Benchmarks

Datasets. We evaluate our method across three indoor datasets, ScanNet [6], ScanNet200 [34] and ScanNet++ [43]. **ScanNet** [6] is a standard dataset for 3D scene understanding. It consists of 1,513 scans with 20 semantic categories. We subsample every 20th frame of the original sequences. **ScanNet200** [34] uses the same physical scenes as ScanNet but expands the label set to 200 categories. This introduces a challenging long-tail distribution that requires more fine-grained understanding. **ScanNet++** [43] was captured with laser scanners and DSLR cameras that yield an improved depth resolution compared to the depth sensors used previously. It also contains dense annotations of more than 1000 different object types which are remapped and truncated to the top 100 classes for the standard semantic benchmark.
Benchmark Protocols. To assess Pano3D, we adopt two 3D-centric evaluation protocols concurrently established by UNITE [20] and IGGT [23].
(1) *Mesh-based Evaluation (UNITE):* The semantic performance is evaluated independent from geometric imprecision. To do so, UNITE maps its semantic predictions on the GT clean mesh used in official benchmarks. They lift their predicted features to 3D, downsample and aggregate them in a local neighborhood before transferring them on the mesh with nearest neighbor. They report mean 3D intersection over union (3D mIoU) and mean accuracy (mAcc) for semantic segmentation, and class-agnostic average precision AP, AP25, and AP50 for 3D instance detection.
(2) *Voxel-based Evaluation (IGGT):* IGGT evaluates the semantic predictions jointly with the predicted geometry. To do so they discretize the GT and predicted 3D point cloud into a voxel grid (voxel size 0.1m). A prediction is a true positive only if the model correctly predicts both the spatial occupancy (reconstruction) and the semantic category (segmentation). We report the 3D mIoU on voxelized predictions alongside standard per view and Multi-View (MVD) with median alignment geometric metrics, including Absolute Relative Error (Abs. Rel) and Inlier Ratio (τ) at a 1.03

Table 1: 3D semantic segmentation. Comparison on ScanNet, ScanNet200 (open-vocabulary), and ScanNet++ with other RGB-only methods using the UNITE metrics. We report 3D mIoU and mAcc. † denotes evaluation by [20].

	ScanNet		ScanNet200		ScanNet++	
Method	$\text{mIoU}_{3D}\uparrow$	$\text{mAcc}_{3D}\uparrow$	$\text{mIoU}_{3D}\uparrow$	$\text{mAcc}_{3D}\uparrow$	$\text{mIoU}_{3D}\uparrow$	$\text{mAcc}_{3D}\uparrow$
Pe3R† [14]	10.7	19.8	2.5	6.5	8.3	16.0
Uni3R† [36]	29.3	39.4	4.1	6.8	5.2	10.8
LSM† [8]	32.2	41.1	6.3	9.7	14.3	26.5
PanSt3R† [45]	42.6	49.7	13.3	19.9	21.6	31.4
UNITE [20]	48.7	68.3	14.5	26.3	17.2	37.0
Pano3D (MUSt3R)	**65.3**	**76.6**	**24.6**	**32.9**	**29.7**	**42.0**

Table 2: Class-agnostic 3D instance segmentation. Similar to Tab. 1, we report the UNITE metrics. † denotes evaluation by [20].

	ScanNet			ScanNet200			ScanNet++		
Method	AP	AP_{50}	AP_{25}	AP	AP_{50}	AP_{25}	AP	AP_{50}	AP_{25}
SAM3D [42]	6.3	17.9	47.3	12.1	28.6	54.1	3.0	7.9	22.3
PanSt3R† [45]	11.4	29.3	53.4	10.6	27.3	50.8	6.5	15.9	33.9
UNITE [20]	13.2	29.6	57.2	12.3	28.8	58.2	6.6	16.1	**40.4**
Pano3D (MUSt3R)	**15.7**	**38.9**	**66.9**	**14.6**	**36.8**	**59.7**	**6.9**	**18.2**	39.7

threshold. These two complementary protocols enable benchmarking 3D semantic understanding and depth accuracy separately. This evaluation is more challenging than mesh aggregation, as nearest-neighbor interpolation on a mesh can smooth over small geometric drifts, whereas voxelization strictly penalizes spatial misalignment.

4.3 Comparison with the State of the Art

To compare to the most closely related state-of-the-art methods, the concurrent work UNITE [20] and the very recent work IGGT [23], we use their respective evaluation protocols and report these comparisons in separate sections.

Comparison to the State of the Art with the UNITE Metrics [20]. For both tasks in the UNITE benchmark, we first detail how we transfer our predictions on the mesh, and then compare to other RGB only methods.

• **Semantic Segmentation (Tab. 1).** To evaluate on the mesh, we aggregate the class logits from all the queries (marginalization) to get logits on the frames. We then lift the logits map using GT depth information and aggregate them on the mesh via k-nearest neighbors. Finally, we interpolate the class logits on the mesh and then take the most probable class at each 3D point.

Tab. 1 reports results for 3D semantic segmentation on ScanNet, ScanNet200 and ScanNet++. We observe that Pano3D achieves state-of-the-art results by a margin. It outperforms UNITE by 16.6 mIoU on ScanNet, 10.1 mIoU on ScanNet200 and 12.5 mIoU on ScanNet++. While UNITE attempts to distill a large amount of teacher

information (instances, semantics, articulations) via contrastive clustering, Pano3D directly optimizes for the target task using a set-based decoder, allowing it to detect instances more accurately. We note that the mIoU performance gap is larger than that of mAcc. This can be explained by the fact that UNITE clusters high-dimensional features, which can cause the mask to bleed onto the background. Hence it covers the ground truth mask but results in more false positives, which reduce the mIoU but not the mAcc. In contrast, our mask decoder's Dice loss directly optimizes for IoU, resulting in crisper boundaries and fewer false positives.

Furthermore, Pano3D outperforms PanSt3R [45] by an even larger margin. PanSt3R relies on frozen geometric features combined with 2D DINOv2 features and selects a subset of "memory frames" for feature extraction. In contrast, Pano3D's global object queries attend to the entire image sequence. This ensures full scene coverage and better agreement from multi-view information, whereas static frame selection can miss objects that appear in views which were not selected.

• **Instance Segmentation (Tab. 2)**. In Pano3D, each query predicts mask logits for each pixel of the entire sequence. Thus, it's likely that some queries predict masks logits for the same pixels (e.g., a mouse pad might get a mask proposal, but the query that detects the desk on which it sits might want to integrate the mouse pad into its mask). To resolve this, we aggregate the mask proposals on the mesh (weighted by k nearest neighbors with $k=5$), and use a winner-takes-all approach (the most confident query at each 3D point is kept) without relying on expensive postprocessing.

Tab. 2 evaluates class-agnostic 3D instance segmentation. It shows consistent performance gain of Pano3D compared to other works on ScanNet and ScanNet200, for example +9.3 in AP50 on ScanNet over the second best method UNITE. Because our object queries are optimized concurrently with the geometric backbone, they natively learn to produce 3D-consistent masks. The superior AP scores on ScanNet, ScanNet200 and ScanNet++ demonstrate that our queries are more effective at delineating instances than clustering-based alternatives. On ScanNet++ that contains more instances per scene on average, and for which we train 200 queries, Pano3D struggles to detect accurately instances, leading to only slight improvement over UNITE in AP and AP50. While AP and AP50 rewards detection with accurate boundaries, AP25 rewards detecting additional small objects with less precision. This is possible with the clustering approach used by UNITE, and is the reason why our scores are higher in AP, but lower in AP25. If we detect an instance, we detect it more precisely than instance feature learning frameworks that need clustering. However, as the number of instances gets larger in the scene, Pano3D can miss some objects that UNITE might detect by finding small clusters in 3D. We also observe that the model sometimes groups instances that are close into the same mask as observed in Fig. 6. This harms object detection, but has less impact on semantics where we outperform other methods by a larger margin (Table 1). It is also worth highlighting that even though UNITE distills CLIP features whereas we train on the labels, this evaluation is **class-agnostic** and shows our masks are of superior quality, set aside the classes.

Comparison to the State of the Art with the IGGT Voxel-based Evaluation [23]. Following IGGT [23], we evaluate the 3D consistency of our masks using a voxel-based metric on their subset of the ScanNet++ benchmark. It consists of 9 scenes

Table 3: Main results on the IGGT benchmark (ScanNet++). We evaluate Pano3D on joint reconstruction and semantic understanding following the protocol of IGGT [23]. We report geometric fidelity (AbsRel, τ) per frame and multi-view (MVD), sequential tracking metrics, and semantic segmentation in both 2D and 3D. All 3D mIoU scores are evaluated on voxelized predictions. ‡ denotes our reproduction of the model. * denotes our evaluation with the official inference script. The rest of IGGT scores and previous works are evaluated by IGGT [23].

	Reconstruction				Semantics	
Method	AbsRel ↓	τ ↑	AbsRel (MVD) ↓	τ (MVD) ↑	2D mIoU ↑	3D mIoU ↑
ScanNet++						
LSeg [22]	n/a	n/a	n/a	n/a	22.61	n/a
OpenSeg [10]	n/a	n/a	n/a	n/a	13.92	n/a
Feature-3DGS [44]	5.92	41.64	–	–	22.47	10.59
LSM [8] (Multi-View)	2.96	83.28	–	–	17.88	15.17
VGGT [37]	2.75	85.41	–	–	–	–
IGGT [23]	2.61	85.66	2.73*	83.90*	31.31	20.14
MUSt3R‡ [1]	2.95	81.87	2.88	80.62	n/a	n/a
Pi3‡ [39]	**2.44**	**87.50**	2.49	83.40	n/a	n/a
Pano3D (MUSt3R)	3.10	78.21	2.59	74.81	46.61	24.88
Pano3D (Pi3)	**2.44**	87.30	**2.42**	**84.93**	**52.54**	**29.46**

from ScanNet++ containing each around 10 frames. We align our predictions with Procrustes alignment [25] and apply density and confidence filter to remove outliers.

As observed in Tab. 3, our MUSt3R model is geometrically less precise than the VGGT model [37] used by IGGT. Indeed, VGGT is a significantly larger model, i.e., VGGT contains more than 1B parameters, whereas MUSt3R only has 468M parameters. This translates to a slightly low 3D reconstruction accuracy of our Pano3D model compared to IGGT. However, the semantic performance including the 3D mIoU is higher for our Pano3D model. We can also observe that learning instances and semantics in the model slightly degrades the per-view reconstruction performance. When using Pi3 [39] which is closer in size and quality to VGGT [37], we notice an improvement in multi-view depth while maintaining the per-view reconstruction performance, and even better downstream segmentation than with MUSt3R [1]. It suggests that better FRMs naturally produce stronger features for segmentation. However, we acknowledge that IGGT is a fully open-vocabulary method since it's trained on a very diverse set of instance masks, and then queries frozen 2D models. Hence, our in-domain training naturally yields better segmentation, and the alignment and filtering pipeline reduces the impact of geometric errors. We also highlight that being an open-vocabulary method, their pipeline is more cumbersome and not feedforward only: they use a heavier model, use clustering to produce masks, reproject the masks in 2D and call external specialized models for semantics [10].

4.4 Ablation Studies

We validate our core architectural and training choices through a series of ablation studies on the ScanNet validation set. We evaluate 3D mIoU and class agnostic 3D AP following UNITE [20] as detailed in Sec. 4.2. For 2D mIoU and 2D mAcc, as well as multi-view depth estimation (MVD), we take 20 frames from the validation set (we

Table 4: Ablation studies on ScanNet validation. We investigate training regimes, architectural components, external priors, and backbone generality. (FT: Finetuned, G: Geometry, S: Semantics). Joint optimization (**G+S**) maintains geometric fidelity while improving instance localization.

	3D Mesh		2D Metrics		Multi View Depth	
Variant	**mIoU3D ↑**	**AP3D ↑**	**mIoU2D ↑**	**mAcc2D ↑**	**AbsRel ↓**	**τ<1.03 ↑**
(1) Training Regime (MUSt3R)						
Frozen Geo + Sem Head	61.9	10.8	72.1	83.1	5.2	44.8%
FT Geometry Only (G)	n/a	n/a	n/a	n/a	3.4	58.1%
FT Semantics Only (S)	n/a	n/a	72.0	82.8	156.3	2.0%
End to end (training encoder)	63.4	**16.6**	72.8	82.9	**3.3**	**60.7%**
Pano3D (MUSt3R) (G + S)	**65.3**	15.7	**73.5**	**83.7**	3.5	57.2%
(2) Feature Bridge						
Encoder Context (E_i) only	60.4	7.8	70.5	82.0	**3.4**	**58.1%**
Final Geo features (F_i) only	63.8	13.9	73.0	**83.9**	3.7	53.2%
Pano3D (MUSt3R) (F_i + E_i)	**65.3**	**15.7**	**73.5**	83.7	3.5	57.2%
(3) Loss coefficient						
λ_{geo}=λ_{sem}	61.8	12.1	72.9	82.1	5.1	39.6%
λ_{geo}=20×λ_{sem}	63.3	**16.2**	71.7	81.8	**3.4**	**57.0%**
Pano3D (MUSt3R) (λ_{geo}=10×λ_{sem})	**65.3**	15.7	**73.5**	**83.3**	3.5	56.7%

first uniformly sample 200 frames, and select the first 20). In MVD, we apply standard median alignment on the entire sequence whereas the per view depth evaluated in Tab. 3 scales each frame independently. Unless specified otherwise, we use the online version of our method for ablations, building on the MUSt3R reconstruction model [1].

Joint Finetuning. A central claim of our methodology (Sec. 3.3) is the introduction of a unified model trained jointly on both 3D reconstruction and panoptic segmentation. Tab. 4 (1) shows how joint finetuning benefits both tasks: while finetuning the model only on semantics (*FT Semantics Only*) causes the geometric latent space to be completely corrupted—evidenced by the catastrophic increase in AbsRel—our joint optimization ($G + S$) preserves the geometric fidelity of the original FRM. A critical insight is that training on geometry and semantics increases the MVD performance compared to the frozen model, as also evidenced in Tab. 3. Even though the per frame depth metrics are slightly degraded (meaning the predictions are a little blurred), the geometric layout at the scene level is better (less drift or unconfident predictions). However, the MVD final performance is still below MUSt3R [1] finetuned only on geometry.

It is also worth noting that joint training provides a more important boost in instance localization rather than semantics, increasing AP by 4.9 compared to the frozen baseline (15.7 vs 10.8). This suggests that the joint optimization does not merely inject semantic labels, but actively refines the model's ability to delineate 3D object boundaries and maintain multi-view consistency. This confirms that semantic gradients act as an implicit structural regularizer. Finally, unfreezing the encoder and finetuning the full architecture end-to-end yields improvement in geometry, but loses semantic capacity over Pano3D, while increasing the training cost.

Combining Encoder and Decoder Features. We investigate the necessity of our multi-level feature adaptation. To do so, we provide the mask decoder head only with encoder or decoder features. As shown in Tab. 4 (2), using only the initial encoder features yields poorer performance. The mask decoder can infer multi-view masks by cross attending to the full sequence, but has **-3** mIoU as it lacks multi-view

Table 5: Ablation studies on model components (ScanNet).

Variant	mIoU2D ↑	mAcc2D ↑
Pano3D	73.5	83.7
Pano3D + DinoV2	71.7	82.5
Pano3D + Dense Head	**73.9**	**84.2**
Pano3D + Dense Head and QUBO [45]	72.6	83.6

Table 6: Impact of the gradient scaling.

	Segmentation		Per-Frame Depth		MVD	
Gradient Factor	mAcc	mIoU	AbsRel	$\tau < 1.03$	AbsRel	$\tau < 1.03$
Frozen Pi3 [39]	n/a	n/a	2.58	81.25	3.01	73.11
1.0 (no scaling)	61.59	47.41	2.44	82.99	2.95	73.20
2.0	60.50	44.86	2.49	82.32	3.18	68.93
0.5	**61.79**	**47.53**	2.42	83.99	3.29	66.28
0.1	60.50	44.91	**2.32**	**84.56**	**2.77**	**76.26**

consistency of the cross-view decoder. Using only the decoder features yields a closer performance in segmentation, yet the depth metrics are lower compared to the final version, as the decoder loses some capacity to refine geometry. This confirms that object representations are distributed across the hierarchy of the geometric backbone, from local monocular context in the encoder to cross-view scene aware features, and that a simple skip connection in Pano3D provides both capacity and efficiency to reconstruct and segment a scene.

Loss Balancing. As in Tab. 4 (3), training with a weak geometry weight degrades the frozen geometric performance while giving lower semantic performance, similarly to the FT semantics only ablations in Tab. 4 (1). This demonstrates that finetuning the pretrained representations on both tasks not only preserves the geometry better, but is also beneficial for segmentation. On the contrary, increasing the geometry weight too much gives marginal gains in MVD, but loses semantic understanding.

Model Components. We ablate the design choices comparing to the closest method PanSt3R [45]. First, as we train in-domain, we replace the frozen text embeddings with a (learnable) dense layer. Tab. 5 shows that learning the class embeddings yields diminishing returns in segmentation scores. In our method, the performance bottleneck is more the instance detection and refining boundaries rather than classifying a given mask. To investigate the impact of the image encoder, we also experiment with adding DINOv2 [28] to our method. Simply inputting DINOv2 features to the mask decoder makes our approach heavier while degrading the segmentation results. Finally, we apply QUBO [45] on our predictions with the default parameters of PanSt3R [45], which gives small regression in segmentation scores, therefore adding processing time without benefit.

Gradient Scaling. We take the Pi3 [39] model that we train and evaluate in ScanNet++, following the same frame selection protocol as defined at the beginning of this section for ScanNet. We train the model and experiment with different scaling factors ranging from 0.1 (downscaling segmentation gradients) to 2.0 (multiplying the segmentation gradients) that we apply before the gradient propagation in the multi-view decoder. Results on segmentation, per-view depth and MVD in Tab. 6

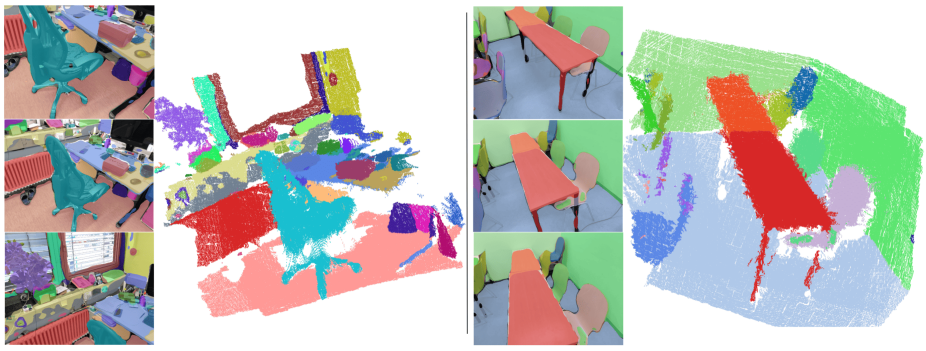

Fig. 4: 3D reconstruction and instance segmentation on ScanNet++: Our model detects 3D-consistent instances across a given sequence (the same query highlights the same object in 3D in all the views). Instances are overlaid on the RGB images, where one color corresponds to one mask (as there can be many instances, some colors may be close). Our model is able to detect small objects, and to maintain good reconstruction capabilities in a shared reference frame in a single feedforward pass.

indicate that reducing the semantic gradients in the geometry decoder yields slightly lower segmentation, but improves geometry the most. Even though not scaling (factor 1.0) improves both geometry and semantics contrary to 2.0 and 0.5, we retain for our final model the 0.1 factor that produces good segmentation (-1 point in mIoU) and a bigger improvement in geometry (+3% in per view and multiview inliers).

4.5 Qualitative Observations

3D Semantic Instance Segmentation. Fig. 4 demonstrates the capabilities of our Pano3D (MUSt3R [1] backbone) on ScanNet++ [43]. In challenging scenes with a lot of small objects (Fig. 4 left), the queries successfully detect the same object throughout the input sequence. In environments with similar objects (like the chairs or the two nearby tables on the right of Fig. 4 right), the model correctly separates the objects. Predicting a pointmap that is pixel-aligned with the masks enables the direct extraction of a segmented 3D point cloud in a single feedforward pass. We provide additional qualitative results with different backbones in Sec. C of the supplementary.
Benefits of Joint Training. Optimizing the geometry decoder modifies the feature space by injecting 'objectness' priors. We look at the modifications of this joint training on geometry by visualising the confidence associated to the 3D predictions of the geometry backbone. As shown in Fig. 5, the frozen MUSt3R backbone hallucinates confidence on ambiguous geometries (e.g., treating a glass door as a wall on the left example) and is unconfident on reflective, textureless areas (right example). On the contrary, the finetuned model detects ambiguous areas like windows and is legitimately confident on objects like tables and boxes. More largely, we observe that the uncertainty in the finetuned model is mostly at the edges of objects, and on ambiguous areas (windows, mirrors).
Cluttered Environments. Our method also inherits the drawbacks of query-based detection. In environments with a lot of objects as in Fig. 6, our model struggles to

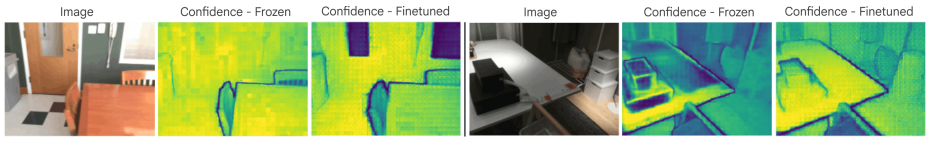

Fig. 5: Benefits of joint training: The middle column shows the confidence maps from the frozen MUSt3R [1] model. The model doesn't detect any anomaly on the wall with windows (on the left example) and is unconfident on reflective areas (like the big table on the right). Our model, finetuned with geometry and segmentation supervision, shows a different behavior that demonstrates how joint training modifies the geometric understanding. The finetuned model does not amplify the confidence of the frozen model, but demonstrates object understanding by highlighting glass windows as ambiguous areas separately from the door or wall (left example), and reinforcing the confidence on the center of objects despite shadows (on the boxes on the right example) or reflections (on the table).

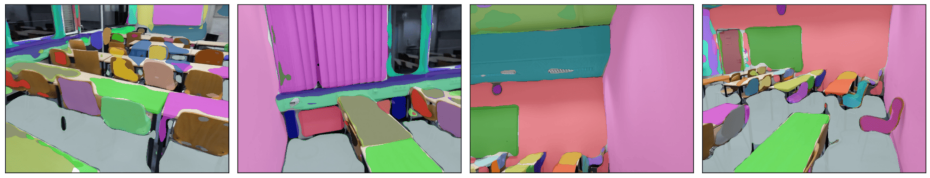

Fig. 6: Cluttered environments: In scenes where there are a lot of objects, our model struggles to separate instances and groups different objects into one mask. Here, all the frames are part of the same sequence. We overlay instance predictions in different colors (due to the high number of instances, some might share similar colors). Some masks in the first image (left) group two different chairs or two different tables.

detect all the instances and detections 'bleed' on multiple objects. For instance, one query might detect two chairs that are close.

5 Conclusion

We presented Pano3D, a streamlined integrated framework for joint 3D reconstruction and panoptic segmentation from unposed image collections. By appending a set-based panoptic head to the final representations of feedforward reconstruction models and jointly finetuning the entire system, Pano3D manages to combine information at different stages and achieves state-of-the-art results. This work opens a path towards treating 3D reconstruction and scene understanding from unposed views as one task. While our current method relies on finetuning an existing geometry-first backbone, our findings suggest that potential future foundation models should be trained jointly on structure and semantics from inception. We envision fully unified representations where scene-level geometric information interacts with object centric representations.

Acknowledgements. We acknowledge Walid Bousselham, Mohammad Fahes and Théodore Fougereux for the feedback and support throughout the project. We also thank Anurag Arnab and Guillaume Le Moing for the insightful discussions.

References

1. Cabon, Y., Stoffl, L., Antsfeld, L., Csurka, G., Chidlovskii, B., Revaud, J., Leroy, V.: MUSt3R: Multi-view network for stereo 3D reconstruction. In: CVPR (2025)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
3. Chen, X., Xia, T., Xu, S., Yang, J., Chai, J., Cheng, Z.: Sab3r: Semantic-augmented backbone in 3d reconstruction. arXiv preprint arXiv:2506.02112 (2025)
4. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: CVPR (2020)
5. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)
6. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
8. Fan, Z., Zhang, J., Cong, W., Wang, P., Li, R., Wen, K., Zhou, S., Kadambi, A., Wang, Z., Xu, D., et al.: Large spatial model: End-to-end unposed images to semantic 3d. NeurIPS (2024)
9. Fu, X., Zhang, S., Chen, T., Lu, Y., Zhu, L., Zhou, X., Geiger, A., Liao, Y.: Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In: 3DV (2022)
10. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: ECCV (2022)
11. Gong, Z., Li, X., Tosi, F., Han, J., Mattoccia, S., Cai, J., Poggi, M.: Ov3r: Open-vocabulary semantic 3d reconstruction from rgb videos. arXiv preprint arXiv:2507.22052 (2025)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: CVPR (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
14. Hu, J., Wang, S., Wang, X.: PE3R: Perception-efficient 3D reconstruction. arXiv:2503.07507 (2025)
15. Jain, A., Katara, P., Gkanatsios, N., Harley, A.W., Sarch, G., Aggarwal, K., Chaudhary, V., Fragkiadaki, K.: ODIN: A single model for 2D and 3D segmentation. CVPR (2024)
16. Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (TOG) (2023)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
18. Kirillov, A., He, K., Girshick, R.B., Rother, C., Dollár, P.: Panoptic segmentation. CVPR (2018)
19. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. ArXiv (2022)
20. Koch, S., Wald, J., Matsuki, H., Hermosilla, P., Ropinski, T., Tombari, F.: Unified semantic transformer for 3d scene understanding. arXiv preprint arXiv:2512.14364 (2025)
21. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3D with MASt3R. In: ECCV (2024)
22. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. arXiv preprint arXiv:2201.03546 (2022)
23. Li, H., Zou, Z., Liu, F., Zhang, X., Hong, F., Cao, Y., Lan, Y., Zhang, M., Yu, G., Zhang, D., Liu, Z.: IGGT: Instance-grounded geometry transformer for semantic 3D reconstruction. In: ICLR (2025)

24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014)
25. Luo, B., Hancock, E.: Iterative procrustes alignment with the em algorithm. Image and Vision Computing (2002)
26. McInnes, L., Healy, J.: Accelerated hierarchical density based clustering. In: International Conference on Data Mining Workshops (ICDMW) (2017)
27. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM (2021)
28. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
29. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: CVPR (2023)
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
31. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: CVPR (2021)
32. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C.K., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C., Girshick, R.B., Doll'ar, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. ArXiv (2024)
33. Robert, D., Vallet, B., Landrieu, L.: Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In: CVPR (2022)
34. Rozenberszki, D., Litany, O., Dai, A.: Language-grounded indoor 3d semantic segmentation in the wild. In: ECCV (2022)
35. Stary, M., Gaubil, J., Tewari, A.K., Sitzmann, V.: Understanding multi-view transformers. ArXiv (2025)
36. Sun, X., Jiang, H., Liu, L., Nam, S., Kang, G., Wang, X., Sui, W., Su, Z., Liu, W., Wang, X., Park, E.: Uni3R: Unified 3D reconstruction and semantic understanding via generalizable gaussian splatting from unposed multi-view images. ArXiv (2025)
37. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: CVPR (2025)
38. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3D vision made easy. In: CVPR (2024)
39. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. In: ICLR (2026)
40. Weinzaepfel, P., Lucas, T., Leroy, V., Cabon, Y., Arora, V., Brégier, R., Csurka, G., Antsfeld, L., Chidlovskii, B., Revaud, J.: Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In: CVPR (2023)
41. Xu, Q., Wei, D., Zhao, L., Li, W., Huang, Z., Ji, S., Liu, P.: Siu3r: Simultaneous scene understanding and 3d reconstruction beyond feature alignment. ArXiv (2025)
42. Yang, Y., Wu, X., He, T., Zhao, H., Liu, X.: Sam3d: Segment anything in 3d scenes. ArXiv (2023)
43. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: CVPR (2023)
44. Zhou, S., Chang, H., Jiang, S., Fan, Z., Zhu, Z., Xu, D., Chari, P., You, S., Wang, Z., Kadambi, A.: Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In: CVPR (2024)
45. Zust, L., Cabon, Y., Marrie, J., Antsfeld, L., Chidlovskii, B., Revaud, J., Csurka, G.: Panst3r: Multi-view consistent panoptic segmentation. In: CVPR (2025)