Panoptic Lifting for 3D Scene Understanding with Neural Fields

Yawar Siddiqui[1,2] Lorenzo Porzi[2] Samuel Rota Bulò[2]
Norman Müller[1,2] Matthias Nießner[1] Angela Dai[1] Peter Kontschieder[2]
Technical University of Munich[1] Meta Reality Labs Zurich[2]

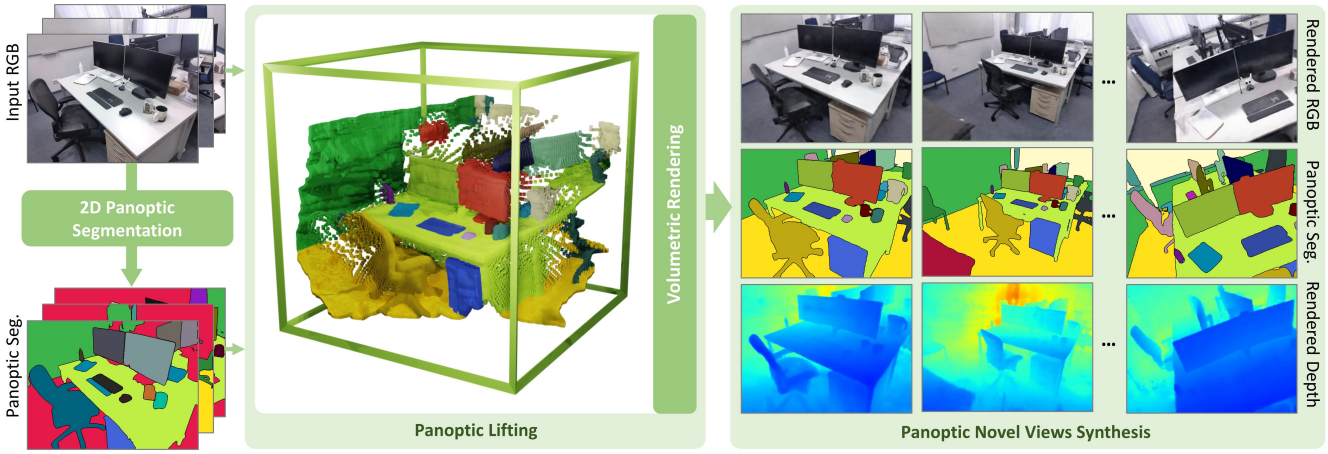

Figure 1. Given only RGB images of an *in-the-wild* scene as input, our method optimizes a panoptic radiance field which can be queried for color, depth, semantics, and instances for any point in space. We obtain poses for input images with COLMAP [34], and 2D panoptic segmentation masks using a pretrained off-the-shelf network [6]. During training, our method lifts these 2D segmentation masks, which are often noisy and view-inconsistent, into a consistent 3D panoptic radiance field. Once trained, our model is able to render images and their corresponding panoptic segmentation masks from both existing and novel viewpoints.

Abstract

We propose Panoptic Lifting, a novel approach for learning panoptic 3D volumetric representations from images of in-the-wild scenes. Once trained, our model can render color images together with 3D-consistent panoptic segmentation from novel viewpoints. Unlike existing approaches which use 3D input directly or indirectly, our method requires only machine-generated 2D panoptic segmentation masks inferred from a pre-trained network. Our core contribution is a panoptic lifting scheme based on a neural field representation that generates a unified and multi-view consistent, 3D panoptic representation of the scene. To account for inconsistencies of 2D instance identifiers across views, we solve a linear assignment with a cost based on the model's current predictions and the machine-generated segmentation masks, thus enabling us to lift 2D instances to 3D in a consistent way. We further propose and ablate contributions that make our method more robust to noisy, machine-generated labels, including test-time augmentations for confidence estimates, segment consistency loss, bounded segmentation fields, and gradient stopping. Experimental results validate our approach on the challenging Hypersim, Replica, and ScanNet datasets, improving by 8.4, 13.8, and 10.6% in scene-level PQ over state of the art.

1. Introduction

Robust panoptic 3D scene understanding models are key to enabling applications such as VR, robot navigation, or self-driving, and more. Panoptic image understanding – the task of segmenting a 2D image into categorical "stuff" areas and individual "thing" instances – has experienced tremendous progress over the past years. These advances can be attributed to continuously improved model architectures and the availability of large-scale labeled 2D datasets, leading to state-of-the-art 2D panoptic segmentation models [6,21,45] that generalize well to unseen images captured in the wild.

Single-image panoptic segmentation, unfortunately, is still insufficient for tasks requiring coherency and consistency across multiple views. In fact, panoptic masks often contain view-specific imperfections and inconsistent classifications, and single-image 2D models naturally lack the ability to track unique object identities across views (see Fig. 2). Ideally, such consistency would stem from a full, 3D understanding of the environment, but lifting machine-generated 2D panoptic segmentations into a coherent 3D panoptic scene representation remains a challenging task.

Project page: nihalsid.github.io/panoptic-lifting/

Recent works [11,19,42,47] have addressed panoptic 3D scene understanding from 2D images by leveraging Neural Radiance Fields (NeRFs) [24], gathering semantic scene data from multiple sources. Some works [11, 42] rely on ground truth 2D and 3D labels, which are expensive and time-consuming to acquire. The work of Kundu *et al*. [19] instead exploits machine-generated 3D bounding box detection and tracking together with 2D semantic segmentation, both computed using off-the-shelf models. However, this approach is limited by the fact that 3D detection models, when compared to 2D panoptic segmentation ones, struggle to generalize beyond the data they were trained on. This is in large part due to the large difference in scale between 2D and 3D training datasets. Furthermore, dependence on multiple pre-trained models increases complexity and introduces potentially compounding sources of error.

In this work we introduce Panoptic Lifting, a novel formulation which represents a static 3D scene as a panoptic radiance field (see Sec. 3.2). Panoptic Lifting supports applications like novel panoptic view synthesis and scene editing, while maintaining robustness to a variety of diverse input data. Our model is trained from only 2D posed images and corresponding, machine-generated panoptic segmentation masks, and can render color, depth, semantics, and 3D-consistent instance information for novel views of the scene.

Starting from a TensoRF [4] architecture that encodes density and view-dependent color information, we introduce lightweight output heads for learning semantic and instance fields. The semantic field, represented as a small MLP, is directly supervised with the machine-generated 2D labels. An additional segment consistency loss guides this supervision to avoid optima that fragment objects in the presence of label inconsistencies. The 3D instance field is modelled by a separate MLP, holding a fixed number of class-agnostic, 3D-consistent surrogate object identifiers. To supervise this field, the 2D machine-generated instances are mapped to the surrogate identifiers by solving an assignment problem. Losses for both the fields are weighted by confidence estimates obtained by test-time augmentation on the 2D panoptic segmentation model. Finally, we discuss specific techniques, e.g. bounded segmentation fields and stopping semantics-to-geometry gradients (see Sec. 3.3), to further limit inconsistent segmentations.
In summary, our contributions are:

- A novel approach to panoptic radiance field representation that models the radiance, semantic class and instance id for each point in the space for a scene by directly lifting machine-generated 2D panoptic labels.
- A robust formulation to handle inherent noise and inconsistencies in machine-generated labels, resulting in a clean, coherent and view-consistent panoptic segmentations from novel views, across diverse data.

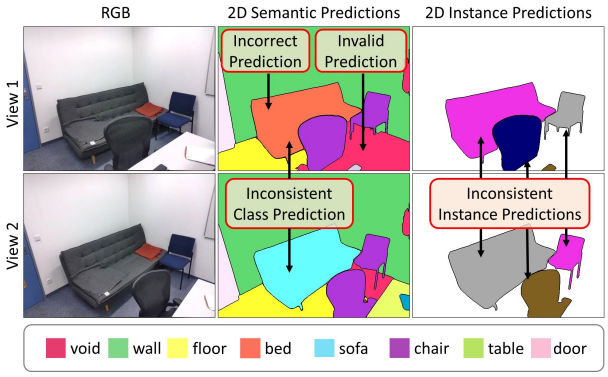

Figure 2. Predictions from state-of-the-art 2D panoptic segmentation methods such as Mask2Former [6] are typically noisy and inconsistent when compared across views of the same scene. Typical failure modes include conflicting labels (e.g. sofa predicted as a bed above) and segmentations (e.g. labeled void above). Furthermore, instance identities are not preserved across frames (represented as different colors).

2. Related Work

Neural Radiance Fields (NeRFs) offer a unified representation to model a scene's photo-realistic appearance [2,4,10, 24,26,40], geometry [1,29,43], and other spatially-varying properties [12,19,39,41,47] (e.g., semantics). NeRF methods can be broadly divided into two macro-categories: i) those which encode the entire scene into a coordinate-based neural network [2,24,40] and ii) those which attach parameters to an explicit 3D structure, such as a voxel grid [10,37], point cloud [44] or spatial hash [27]. A speed-memory trade off exists between the two, as models in (i) are generally more compact, while models in (ii) are often faster for training and inference. In our work we adopt a hybrid approach, modeling appearance and geometry with an explicit neural field derived from TensoRF [4], and semantics and instances with a pair of small implicit MLPs.

NeRFs for semantic 3D scene modeling. The work of Zhi *et al*. [47] first explored encoding semantics into a NeRF, demonstrating how noisy 2D semantic segmentations can be fused into a consistent volumetric model, improving their accuracy and enabling novel view synthesis of semantic masks. Since then, several works have extended this idea, e.g., by adding instance modeling [11, 19, 42] or by encoding abstract visual features [18, 39] from which a semantic segmentation can be derived a-posteriori. Panoptic NeRF [11] and DM-NeRF [42] describe panoptic radiance fields, focusing on the tasks of label transfer and scene editing, respectively. In contrast to our work, they both require some form of manual ground truth for the target scene: panoptically segmented coarse meshes for the former, and per-image 2D panoptic segmentations for the latter. On the other hand, Panoptic Neural Fields (PNF) [19] relies purely

on RGB images, the same setting as ours. PNF, however, exploits a much larger set of predictions from pre-trained networks, including per-image semantic segmentation, 3D object bounding boxes, and object tracking across frames. While 3D tracking enables PNF to handle moving objects, the use of 3D detectors induces strong sensitivity to errors in the predicted boxes, especially beyond the datasets on which these networks were trained on.

2D and 3D panoptic segmentation. The task of 2D panoptic segmentation and its associated metrics were first defined by Kirillov *et al*. [17]. A first wave of works in this field [5, 17, 30] proposed solutions based on combining a semantic segmentation network with an instance segmentation network, often sharing a common backbone. Recent works [6, 7, 46] instead adopt a more unified approach, inspired by the DETR object detector [3]. These works use a single transformer-based network to directly produce panoptic output as a set of image segments for both "things" and "stuff." In our work, we adopt Mask2Former [6] with a Swin-L [23] backbone as our pre-trained 2D panoptic segmenter, due to its state-of-the-art performance on the general-purpose COCO Panoptic [22] dataset.

Panoptic segmentation has also been explored in a 3D context, both as segmentation of pre-computed 3D structures [13, 25, 35, 48] (e.g., voxel grids or point clouds), and as simultaneous 3D segmentation and reconstruction from 2D images [8, 28, 32]. However, these methods leveraging meshes, voxel grids or point clouds, do not allow for photo-realistic novel view synthesis as NeRF-based methods do.

3. Method

Given posed RGB images $\{I\}$ of a scene with corresponding machine-generated 2D panoptic segmentations, our goal is to build an implicit, volumetric representation of the scene that models appearance and density, together with 3D object instances and their semantics. Our approach, called Panoptic Lifting, enables generating 2D panoptic segmentations from novel views in a 3D-consistent fashion, such that the rendered 2D instance ID of the same 3D object is preserved across views.

3.1. Input Data

Input posed images $\{I\}$ are abstracted as sets of viewing rays expressed in world coordinates. A ray r can be parameterized in 3D space as $r = p_0 + td_r$, with p_0 the ray origin, d_r its unit direction and a scalar t.

Rays r belonging to a training image I have an associated RGB color $\hat{c}_r \in \mathbb{R}^3$, a semantic class $\hat{k}_r \in \mathcal{K}$ and a 2D instance ID $\hat{h}_r \in \mathcal{H}_I$. The 2D instance ID $\hat{h}_r$ is defined only for *thing* classes, *i.e*. only if $\hat{k}_r \in \mathcal{K}_\mathtt{T}$, where $\mathcal{K}_\mathtt{T}$ and $\mathcal{K}_\mathtt{S}$ partition $\mathcal{K}$ into *thing* and *stuff* classes, respectively.

These semantic and instance labels, generated using a pre-trained 2D segmentation network [6], are often noisy and inconsistent, as shown in Fig. 2. While 2D networks provide a probabilistic distribution over the predicted classes and a confidence per pixel (ray), these are often very peaked, even for incorrectly predicted segments (Fig. 4). To better estimate class probabilities and confidences, we run test-time augmentations on the input images (e.g., horizontal flip, scale, brightness, contrast, etc.) and fuse the resulting segmentations by segment clustering (see supplementary for details). This gives us for each pixel (ray r) in the training images, a semantic class distribution $\hat{\kappa}_r$ over the classes $\mathcal{K}$ and a confidence estimate w_r of the prediction.

3.2. Scene Representation and Rendering

Panoptic radiance field. A *panoptic radiance field* is an implicit, volumetric scene representation modeled as a function $\Phi(x, d)$, which assigns to each 3D point $x \in \mathbb{R}^3$ and viewing direction $d \in \mathbb{S}^2$ a density $\sigma \in \mathbb{R}_{\geq 0}$, a semantic class distribution κ over $\mathcal{K}$, a distribution π over surrogate identifiers $\mathcal{J}$ and an RGB color $c \in \mathbb{R}^3$. From this representation, we can derive a per-point distribution over 3D object IDs by considering probability $\kappa(k)\pi(j)$ for each 3D object ID $(k, j) \in \mathcal{H}_{\mathtt{3D}}$, with $\mathcal{H}_{\mathtt{3D}} := \mathcal{K}_\mathtt{T} \times \mathcal{J}$ being the set of all possible 3D instances our model can produce. The function Φ leverages a TensoRF [4] representation for color and density, and we introduce two small MLPs to model the semantic and surrogate identifier fields.

Volumetric rendering. Given the density field σ from Φ as a function of a point x, we can render any vector field f over 3D points along a ray r by the rendering equation [15]:

$$R[f|r, \sigma] := \int_0^\infty \alpha_t(r)\sigma(r_t)f(r_t, d_r)dt\,, \tag{1}$$

where α_t is the transmittance probability at t

$$\alpha_t(r) := \exp\left(-\int_0^t \sigma(r_s)ds\right). \tag{2}$$

For brevity, we use f_r to denote the rendered vector field f along ray r when the density field σ is clear from the context, *i.e*. $f_r := R[f|r, \sigma]$. In particular, this shorthand will be used for rendering all vector-valued fields that are implicitly provided by Φ, namely color c, semantic class distributions κ and surrogate ID distributions π. For example, $c_r := R[c|r, \sigma]$ represents the color field c rendered along ray r.

Rendering panoptic segmentations. One application of Panoptic Lifting is rendering 3D-consistent 2D panoptic segmentations from novel views. Given a pixel from a novel view corresponding to ray r in world coordinates, we can compute a rendered semantic class distribution κ_r using the rendering equation above and obtain a semantic class $k_r^\star$ for the same pixel as the most probable class, *i.e*. $k_r^\star := \arg\max_{k \in \mathcal{K}} \kappa_r(k)$. If $k_r^\star$ is a thing class, we

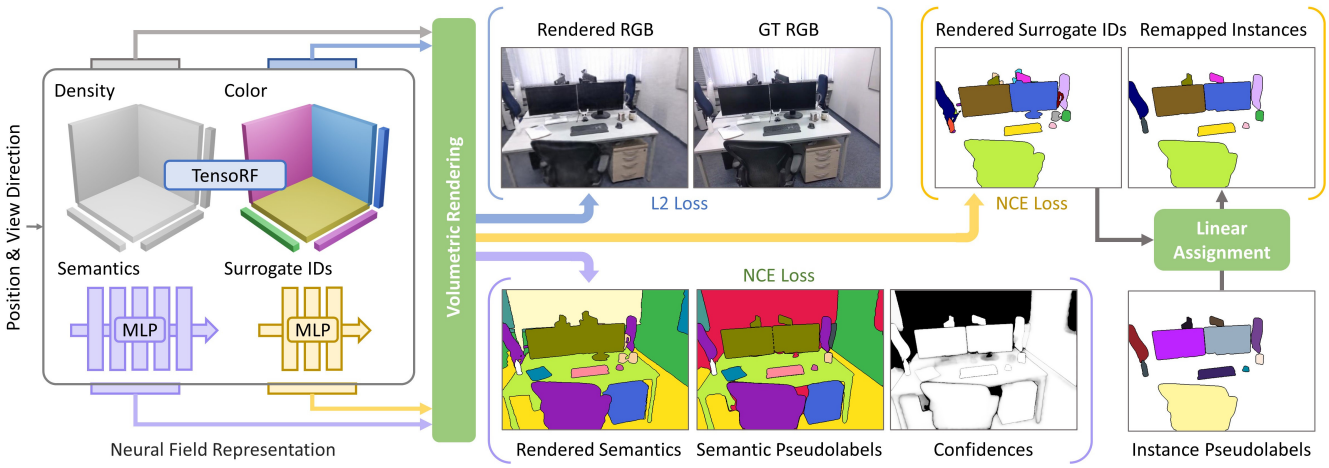

Figure 3. Our neural field representation comprises color, density, semantics, and instances. Color and density leverage TensoRF [4], and we use two small MLPs for semantics and instances. The radiance field is supervised by an L2 photometric loss on the volumetrically rendered radiance along rays. The semantic field is likewise supervised by an NCE loss on the volumetrically rendered class probability distribution and the 2D machine-generated semantic labels. For instances, we first map a machine-generated instance to a 3D surrogate identifier using linear assignment, where the cost depends on the current 3D instance predictions. These mapped 2D instances are used to supervise the rendered instance field with an NCE loss. All segmentation losses are weighted by test-time augmentation confidences.

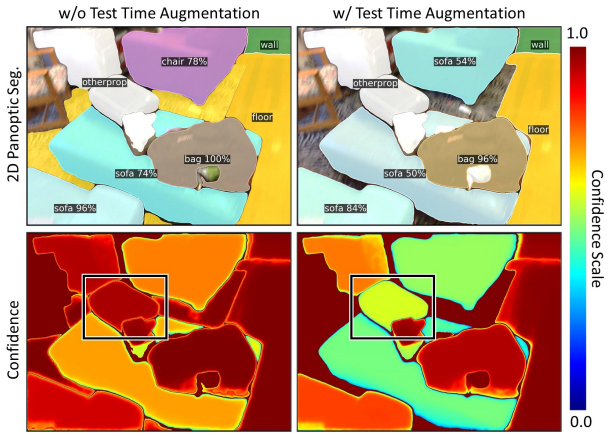

Figure 4. Confidences from a single Mask2Former [6] prediction (left) are often spuriously high, even for incorrectly labeled segments. Above, the *bag* is labeled incorrectly as *otherprop* with high confidence. Merging segmentations from test-time augmentations results in more reliable confidence estimates (right).

can also compute the most probable surrogate identifier $j_r^\star := \arg\max_{j \in \mathcal{J}} \pi_r(j)$ and form a unique 3D instance identifier as the pair $(k_r^\star, j_r^\star) \in \mathcal{H}_{\texttt{3D}}$, which is consistent across the scene. This enables generating multi-view consistent panoptic segmentations for novel views.

3.3. Loss Functions

Appearance loss. Give a batch of rays R, the appearance loss is given as a standard squared Euclidean distance between rendered color field and the ground-truth color:

$$L_{\texttt{RGB}}(R) := \frac{1}{|R|} \sum_{r \in R} \|c_r - \hat{c}_r\|^2 \,. \tag{3}$$

Semantic loss. As described in Sec. 3.1, each ray r has an associated probability distribution $\hat{\kappa}_r$ over the semantic classes $\mathcal{K}$ and a prediction confidence w_r obtained as test-time augmented predictions of a 2D pre-trained panoptic segmentation model. Given the semantic distribution field κ_r rendered along ray r, the semantic loss at r is given by the cross entropy of κ_r relative to $\hat{\kappa}_r$. For a batch of rays R,

$$L_{\texttt{sem}}(R) := -\frac{1}{|R|} \sum_{r \in R} w_r \sum_{k \in \mathcal{K}} \hat{\kappa}_r(k) \log \kappa_r(k) \,. \tag{4}$$

Logits vs. Probabilities. Standard implementations of semantic fields render class logits rather then class distributions [19, 38, 42, 47], converting the rendered logits into probabilities a posteriori via softmax. This approach can potentially introduce semantic inconsistencies across different viewpoints, since it endows the model with the ability of violating geometric constraints induced by the density. Specifically, due to the unbounded nature of logits, a significant signal could in principle be generated even in low density areas, by just providing sufficiently large logits. We resolve this by instead rendering the bounded probability distribution. See Fig. 8 for an example.

Instance loss. Let R denote a batch of rays from an image I, R_h the subset of rays in R that belong to 2D instance $h \in \mathcal{H}_I$, and $H_R \subseteq \mathcal{H}_I$ the subset of 2D instances that are represented in the batch of rays R. Each ray r is assigned the most compatible 3D surrogate identifier via an injective mapping $\Pi_R^\star$, given its 2D machine-generated instance $\hat{h}_r$. The loss minimizes the log-loss of the corresponding predicted probability averaged over all rays in R:

$$L_{\texttt{ins}}(R) := -\frac{1}{|R|} \sum_{r \in R} w_r \log \pi_r(\Pi_R^\star(\hat{h}_r)) \,, \tag{5}$$

with w_r as the prediction confidence. The optimal injective mapping $\Pi_R^\star$ is given by

$$\Pi_R^\star := \underset{\Pi_R}{\operatorname{argmax}} \sum_{h \in H_R} \sum_{r \in R_h} \frac{\pi_r(\Pi_I(h))}{|R_h|}\,, \tag{6}$$

which can be solved as a linear assignment problem.

Segment consistency loss. Let R denote a batch of rays from an image, partitioned into groups of rays $\{R_1, ..., R_m\}$ based on the panoptic segment each ray belongs to. That is, rays are clustered based on their 2D instance or *stuff* class. The segment consistency loss for R is:

$$L_{\texttt{con}}(R) := -\frac{1}{|R|} \sum_{r \in R} w_r \sum_{i=1}^{m} \log \kappa_r(K_i)\,, \tag{7}$$

where w_r is the prediction confidence and K_i is the most probable predicted semantic class within the group R_i, *i.e.* $K_i := \operatorname{argmax}_{k \in \mathcal{K}} \sum_{r \in R_i} w_r \kappa_r(k)$.

Training objective. Let $R = R_S \cup R_I$ be a batch of rays with R_S as rays randomly sampled across the scene and R_I as rays sampled from a single image, then the total loss over the network parameters is $L_{\texttt{tot}}(R) := \lambda_{\texttt{ins}} L_{\texttt{ins}}(R_I) + \lambda_{\texttt{con}} L_{\texttt{con}}(R_I) + \lambda_{\texttt{RGB}} L_{\texttt{RGB}}(R_S) + \lambda_{\texttt{sem}} L_{\texttt{sem}}(R_S)$.

Note that it is important that the supervision for the segmentation with noisy 2D machine labels not influence the geometry. Otherwise the model can still satisfy inconsistent labels by changing the rendering weights, resulting in cloudy geometry to satisfy conflicting segmentation labels (Fig. 8c). To avoid this issue, we stop gradients from the segmentation branches back to the densities.

Implementation Details. Our model is implemented with Pytorch and trained using Adam [16] with learning rates of 5×10^{-4} for the MLPs and 1×10^{-2} for the TensoRF lines and planes. For $L_{\texttt{tot}}$ we use $\lambda_{\texttt{con}} = 1.35$, $\lambda_{\texttt{ins}} = \lambda_{\texttt{sem}} = \lambda_{\texttt{RGB}} = 1$. We train on 1 NVIDIA A6000 for 450k iterations ($\sim$ 10 hours). Further details are in the supplementary.

4. Experiments

We evaluate our method on the tasks of novel view synthesis and novel view panoptic segmentation, and further show scene editing as a possible application.

Data. We show results on scenes from three public datasets: Hypersim [31], Replica [36] and ScanNet [9]. For each of the datasets, we use the available ground-truth poses. The ground-truth semantic and instance labels, however, are only used for evaluation, and are not used for training or refinement of any models. We also show results on in-the-wild scenes, captured with a Pixel3a Android smartphone with COLMAP [34]-computed poses. To obtain the machine-generated 2D panoptic segmentations, we use the same publicly available pre-trained model [6] for all scenes across all datasets. Since this model was trained originally on COCO [22], we map COCO panoptic classes to 21 ScanNet classes (9 *thing* + 12 *stuff*). Hypersim and Replica labels are also mapped to the same class set for evaluation. For the in-the-wild captures we use 31 ScanNet classes (17 *thing* + 14 *stuff*). Further details about the scenes, classes and data splits can be found in the supplementary.

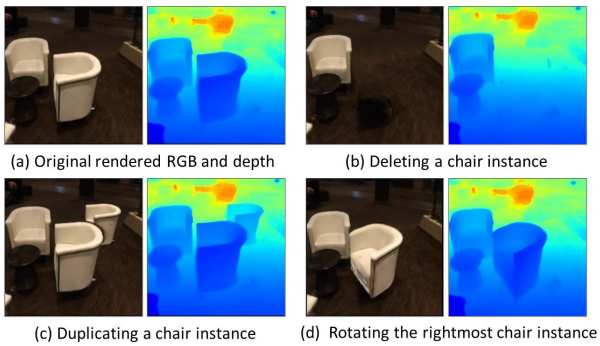

Figure 5. Scene Editing. Once trained, we can generate novel views of a scene with object instances (b) deleted, (c) duplicated or (d) manipulated with affine transforms.

Metrics. We measure the visual fidelity of the synthesized novel views, and the accuracy of their predicted semantic labels using peak signal to noise ratio (PSNR), and mean intersection over union (mIoU), respectively. Recent NeRF-based semantic scene modeling works [11,19] evaluate their panoptic predictions using the Panoptic Quality (PQ) [17] metric. Standard PQ, however, does not measure whether instance identities are consistently preserved across views. To overcome this limitation, we propose a scene-level PQ metric, denoted as PQ^{scene}.

Scene-level Panoptic Quality. Given a set of predicted segments $p \in P$, and ground truth segments $g \in G$, the PQ metric is defined by comparing all possible pairs of segments belonging to an image, marking them as a match if $\text{IoU}(p, g) > 0.5$ (see Sec. 4.1-4.2 of [17]). In PQ^{scene}, we modify the matching process to work on a *scene level*. To achieve this, we first merge the segments into subsets $\mathcal{P} \subset P, \mathcal{G} \subset G$ which contain all segments that, for a certain scene, belong (or are predicted to belong) to the same instance or stuff class. Then, we compare all pairs of *subsets* for each scene, and record a match when $\text{IoU}(\mathcal{P}, \mathcal{G}) > 0.5$.[1]

4.1. Results

We compare with state-of-the-art 2D and NeRF-based 3D semantic and panoptic segmentation methods: Mask2Former [6] predicts 2D panoptic segmentation, SemanticNeRF [47] predicts 3D semantic segmentation, and Panoptic Neural Fields (PNF) [19] and DM-NeRF [42] predict 3D panoptic segmentation. Our method and the neural

[1] Note that PQ^{scene} can be trivially implemented by evaluating standard PQ after tiling together the predictions and ground truths of each scene.

Method	HyperSim [31]			Replica [36]			ScanNet [9]		
	mIoU↑	PQ^{scene} ↑	PSNR↑	mIoU↑	PQ^{scene} ↑	PSNR↑	mIoU↑	PQ^{scene} ↑	PSNR↑
Mask2Former [6]	53.9 (-13.9)	–	–	52.4 (-14.8)	–	–	46.7 (-18.5)	–	–
SemanticNeRF [47]	58.9 (-8.9)	–	26.6	58.5 (-8.7)	–	24.8	59.2 (-6)	–	26.6
DM-NeRF [42]	57.6 (-10.2)	51.6 (-8.5)	28.1	56.0 (-11.2)	44.1 (-13.8)	26.9	49.5 (-15.7)	41.7 (-17.2)	27.5
PNF [19]	50.3 (-17.5)	44.8 (-15.3)	27.4	51.5 (-15.7)	41.1 (-16.8)	29.8	53.9 (-11.3)	48.3 (-10.6)	26.7
PNF + GT Bounding Boxes	58.7 (-9.1)	47.6 (-12.5)	28.1	54.8 (-12.4)	52.5 (-5.4)	**31.6**	58.7 (-6.5)	54.3 (-4.6)	26.8
Panoptic Lifting	**67.8**	**60.1**	**30.1**	**67.2**	**57.9**	29.6	**65.2**	**58.9**	**28.5**

Table 1. Quantitative comparison on novel views from the test set. We outperform both 2D and 3D NeRF methods on semantic and panoptic segmentation tasks. Note that, compared to other methods, *PNF+GT Bounding Boxes* is given the advantage of using ground-truth 3D detections. Mask2Former does not predict scene-level object instances, thus it can't be evaluated for PQ^{scene}.

Segment Consistency	TTA	Bounded Segm. Field	Gradient Blocking	mIoU↑	PQ^{scene} ↑	PSNR↑
✗	✗	✗	✗	57.3	47.9	27.1
✗	✓	✓	✓	60.9	54.3	28.3
✓	✗	✓	✓	63.1	55.2	28.4
✓	✓	✗	✓	62.9	52.5	28.4
✓	✓	✓	✗	61.6	53.7	27.2
✓	✓	✓	✓	**65.2**	**58.9**	**28.5**

Table 2. Ablations of our design choices on ScanNet [9]. The segment consistency loss, test-time augmentations (TTA), probability field, and blocked segmentation gradients all contribute to robustness against real-world noisy labels.

field baselines are all trained using the same set of images, poses and Mask2Former generated 2D labels. For PNF, we additionally provide bounding boxes from a state-of-the-art multi-view 3D object detector [33] pre-trained on ScanNet, or taken from the ground truth (PNF + GT Bounding Boxes). More details are provided in the supplementary.

As shown in Tab. 1 and Fig. 6, we outperform baselines across all datasets on both semantic and panoptic segmentation tasks, without sacrificing view synthesis quality. Fig. 7 additionally shows qualitative results on an in-the-wild capture. We further demonstrate improved consistency and segmentation quality over baselines in the supplemental video.

Our method significantly improves over Mask2Former, harmonizing its inconsistent and noisy outputs by lifting them to 3D. This is reflected by the IoU scores in Tab. 1 (-13.9% w.r.t. Panoptic Lifting), and by evaluating standard PQ on ScanNet, where we register a scores of 43.6% and 60.4% for Mask2Former and our method, respectively. While SemanticNeRF was shown to be robust to synthetic pixel noise, we find that it struggles to handle the error patterns of machine-generated labels. Similarly, DM-NeRF seems to suffer with segment fragmentation when confronted with real-world scenes and machine-generated panoptic labels (Fig. 6). Finally, PNF's sensitivity to errors from the 3D bounding box detector is well highlighted in Tab. 1 (-15.7 to -10.6% PQ^{scene} w.r.t. Panoptic Lifting). When provided with g.t. boxes, PNF is able to partially close the PQ^{scene} gap, but Panoptic Lifting still shows greater robustness to noise in 2D machine-generated labels.

4.1.1 Ablations

In Tab. 2, we show a set of ablations on the ScanNet data. As a first baseline (row 1), we disable all of the robustness-oriented design choices described in Sec. 3.3, obtaining a substantial drop in performance (-8% mIoU, -11% PQ^{scene}).

Segment consistency loss. Fig. 8(d) shows how inconsistent 2D segmentations under different views result in a blend of two labels for the same physical object. This effect is counteracted by our segment consistency loss: disabling it causes the largest drop in mIoU (Tab. 2, row 2).

Test-time augmentation. As shown in Fig. 4, Mask2Former predictions tend to be highly confident, even for incorrectly predicted classes. Test-time augmentation provides smoother confidence estimates and improved masks, boosting both semantic metrics (Tab. 2, row 3).

Bounded vs. unbounded segmentation field. Predicting unbounded logits from the segmentation branch, as in other recent NeRF-based semantic scene modeling works [19, 38, 42, 47], allows the model to "cheat" and predict inconsistent labels even with accurate geometry (see Fig. 8(a,b), Sec. 3.3). Switching to bounded segmentation fields drastically improves consistency, as shown by the PQ^{scene} score in Tab. 2, row 4.

Blocking semantics-to-geometry gradients. Even with a bounded segmentation field, the model can be pushed to learn wrong geometry in an attempt to satisfy inconsistent 2D labels, as shown in Fig. 8(c). This is reflected in a degradation in view synthesis PSNR (Tab. 2, row 5), which we can solve by blocking gradients from semantic and instance branches back to geometry.

4.1.2 Scene Editing

In addition to generating novel views and their panoptic masks, Panoptic Lifting can be used for scene editing. As shown in Fig. 5, once trained, we can generate novel views of a scene with object instances deleted, duplicated or manipulated under affine transformations. Given an instance

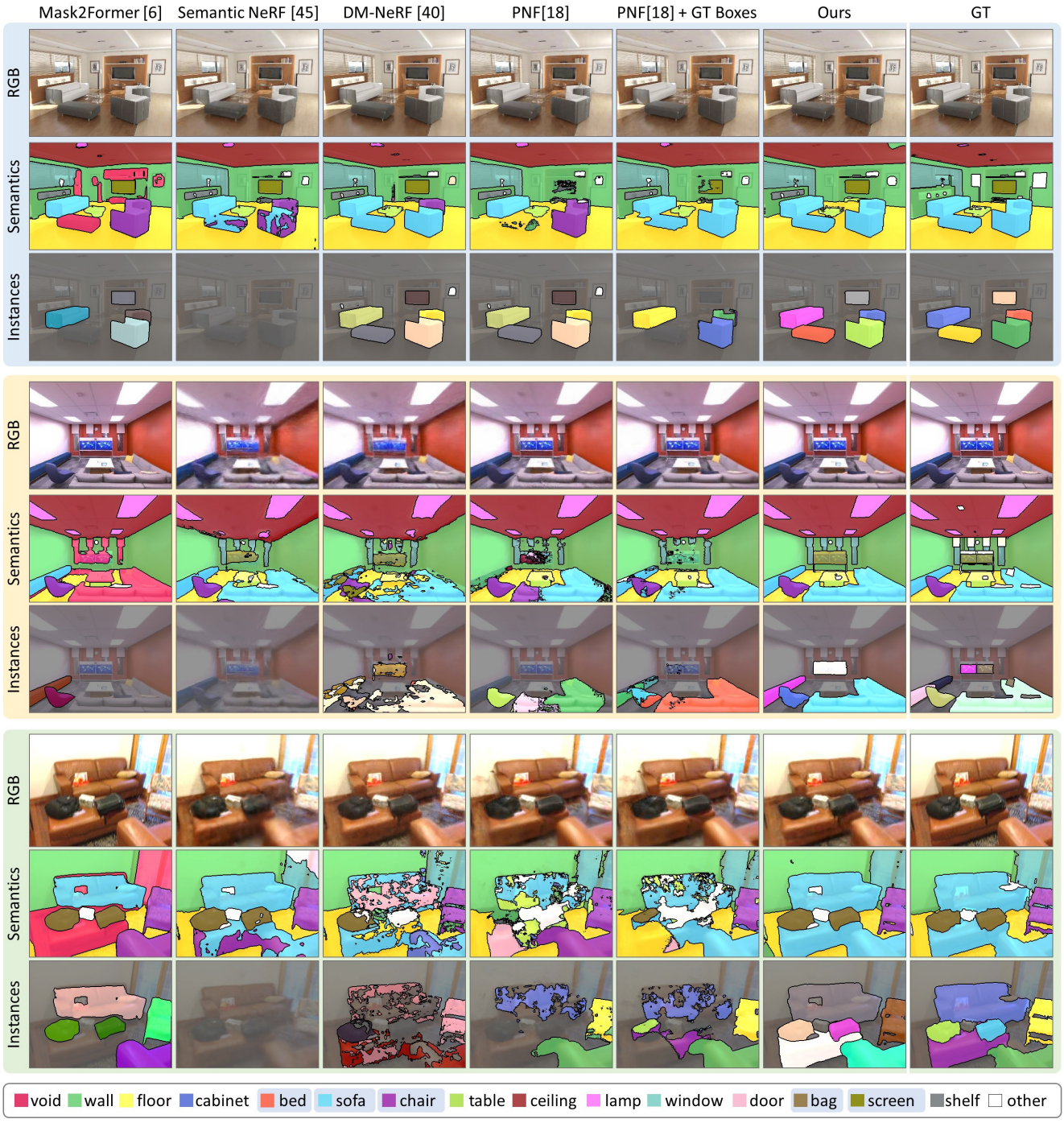

Figure 6. Novel views and their corresponding semantics and instances on Hypersim [31], Replica [36], and ScanNet [9] (top to bottom, respectively). Highlighted legend labels represent instanced (*thing*) classes. All 3D methods use the same posed RGB images and 2D machine-generated labels for training. PNF [19] additionally uses 3D bounding box predictions, while PNF + GT Boxes uses ground truth bounding boxes for instance classes. We outperform the state of the art, producing clean and consistent segmentation masks.

label, deletion is achieved by setting the density of points where a predicted instance is equal to the instance to be deleted to be zero. For duplication, given a new position and rotation for an object to be duplicated, we manipulate the rays passing through the resulting region to query the original instance region instead. Manipulation is achieved by combining deletion and duplication.

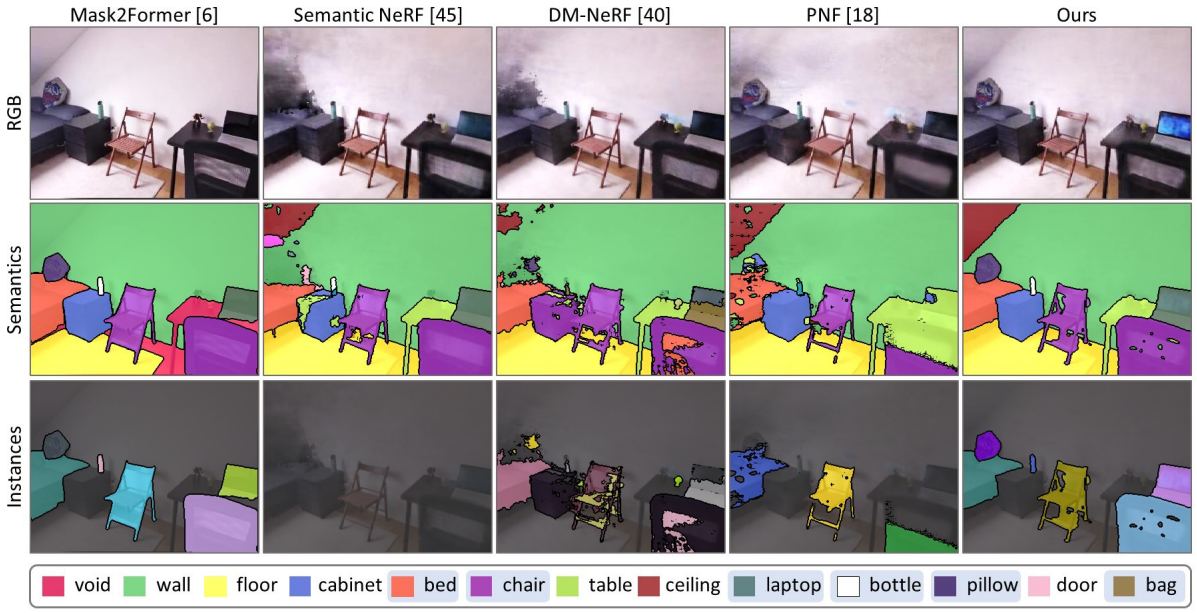

Figure 7. Novel views and their corresponding semantics and instances on *in-the-wild* room capture. Highlighted labels in the legend represent instanced (*thing*) classes. All 3D methods use the same posed RGB images and 2D machine-generated labels for training. PNF [19] additionally uses predictions from a 3D bounding box detector.

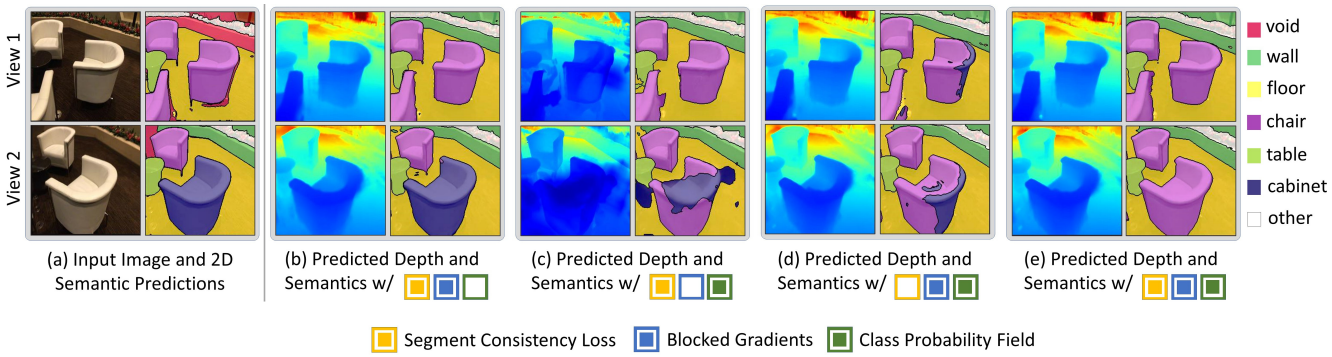

Figure 8. Ablations over components. (a) 2D machine-generated label inconsistency, with the chair labeled as 'chair' and 'cabinet' in different views. (b) Unbounded segmentation field outputs (e.g. MLP predicts logits) can result in inconsistent predictions even with good depth estimates due to improper distribution along the ray. (c) Even with class probability outputs, the model can change the density (and thereby the rendering weights) to better fit noisy 2D labels. (d) Blocking segmentation gradients along with bounding field output produces a mix of 2 classes for the same chair instance, the optima for the NCE loss on segmentations. Using an additional segment consistency loss improves object consistency. (e) Our final method with all components enabled results in clean and consistent segmentations.

4.1.3 Limitations

Panoptic Lifting shows considerable improvements over the state of the art; however, several limitations remain. Our method uses predictions from a pre-trained panoptic segmentation model, and hence is limited to classes with which the original model was trained. In this context, it would be interesting to explore open world segmentation [14, 20] with self-supervised instance clustering. Similar to other NeRF-based approaches, our method is currently run offline due to lengthy pre-processing for pose estimation, 2D segmentation inference, and neural field optimization; here, a promising avenue would be to integrate our approach with state-of-the-art SLAM approaches that run in real-time.

5. Conclusion

We have introduced Panoptic Lifting, a novel approach to lift 2D machine-generated panoptic labels to an implicit 3D volumetric representation. As a result, our model can produce clean, coherent, and 3D-consistent panoptic segmentation masks together with color and depth images for novel views. Compared to state of the art, our model is more robust to the inherent noise present in machine-generated labels, hence resulting in significant improvements across datasets while providing the ability to work on in-the-wild scenes. We believe Panoptic Lifting is an important step towards enabling more robust, holistic 3D scene understanding while posing minimal input requirements.

Acknowledgements

This work was done during Yawar's and Norman's internships at Meta Reality Labs Zurich as well as at TUM, funded by a Meta SRA. Matthias Nießner was also supported by the ERC Starting Grant Scan2CAD (804724). Angela Dai was supported by the Bavarian State Ministry of Science and the Arts coordinated by the Bavarian Research Institute for Digital Transformation (BIDT). We would like to thank Justus Thies, Artem Sevastopolsky, and Guy Gafni for helpful discussions and their feedback.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, June 2022. 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 4, 12

[5] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020. 3

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1, 2, 3, 4, 5, 6, 11, 12

[7] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 3

[8] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34:8282–8293, 2021. 3

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 5, 6, 7, 12, 13

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 2

[11] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *International Conference on 3D Vision (3DV)*, 2022. 2, 5

[12] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021. 2

[13] Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari. Panoster: End-to-end panoptic segmentation of lidar point clouds. *IEEE Robotics and Automation Letters*, 6(2):3216–3223, 2021. 3

[14] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021. 8

[15] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 3

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[17] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 3, 5

[18] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems*, volume 35, 2022. 2

[19] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 2, 4, 5, 6, 7, 8, 11, 12, 13

[20] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 8

[21] Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, Lionel M Ni, Heung-Yeung Shum, et al. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv preprint arXiv:2206.02777*, 2022. 1

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3, 5, 12

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer:

Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 3

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[25] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss. Lidar panoptic segmentation for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8505–8512. IEEE, 2020. 3

[26] Norman Müller, Andrea Simonelli, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. 2

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 2

[28] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4205–4212. IEEE, 2019. 3

[29] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2

[30] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8277–8286, 2019. 3

[31] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10912–10922, 2021. 5, 6, 7, 12, 13

[32] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv preprint arXiv:2002.06289*, 2020. 3

[33] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2397–2406, 2022. 6, 12

[34] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 5

[35] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. Efficientlps: Efficient lidar panoptic segmentation. *IEEE Transactions on Robotics*, 2021. 3

[36] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 5, 6, 7, 12, 13

[37] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2

[38] Matthew Tancik*, Ethan Weber*, Evonne Ng*, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A framework for neural radiance field development, 2022. 4, 6

[39] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *arXiv preprint arXiv:2209.03494*, 2022. 2

[40] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 2

[41] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi SM Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. *arXiv preprint arXiv:2111.13260*, 2021. 2

[42] Bing Wang, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022. 2, 4, 5, 6, 11, 12, 13

[43] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[44] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 2

[45] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV*, 2022. 1

[46] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems*, 34:10326–10338, 2021. 3

[47] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 2, 4, 5, 6, 11, 12

[48] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13194–13203, 2021. 3

In this supplementary document, we discuss additional details about our method Panoptic Lifting. Specifically. in Section A we give additional details about our test time augmentation algorithm. A comparison of rendering performance of our method compared to the baselines is reported in Section B. We also provide implementation details of our method and the baselines (Section C), and the data used for experiments in the main paper (Section D). Finally, we report additional metrics, scene Segmentation Quality ($SQ^{\texttt{scene}}$) and Retrieval Quality ($RQ^{\texttt{scene}}$), in Section E.

A. Test-time Augmentation for Mask2Former

In this section we describe the test-time augmentation strategy we adopt to obtain improved panoptic segmentation masks and per-pixel confidence scores from Mask2Former [6].

A.1. Test-time Augmentation

We run a pre-trained Mask2Former network on multiple augmented versions of each input image, using the following set of transformations: horizontal flip, rescale, contrast, RGB-shift, random gamma, random brightness & contrast, median blur, sharpen, and arbitrary combination of the previously mentioned augmentations. For each transformation, we intercept the Mask2Former outputs before its "panoptic fusion" stage, *i.e.* right after the transformer and pixel decoders (see Sec.3 of [6] for details). These outputs consist of a set of candidate segments, represented as 2D soft masks paired with probability distributions over the classes. After transforming the candidate segments back to the original image resolution and orientation, our next objective is to fuse them into a single, coherent panoptic segmentation.

A.2. Fusing Mask2Former predictions

We denote the candidate segments predicted from all augmented versions of the image as a set of pairs $(\mathsf{m}_i, \mathbf{p}_i), i = 1, \dots, N$, where $\mathsf{m}_i(x, y) \in [0, 1]$ is the predicted probability of pixel (x, y) to belong to segment i, and $\mathbf{p}_i = [p_i^1, ..., p_i^C]$ is the segment's predicted probability distribution over C classes. In the following, we describe a mechanism to combine these predictions into a single panoptic segmentation with associated confidences, following three steps: segment clustering, cluster aggregation and panoptic fusion.

Segment clustering. We build a graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., N\}$, and $\mathcal{E} = \{(i, j) | i, j \in \mathcal{V} \wedge s(i, j) \geq \theta\}$. The matching function $s(i, j)$ is defined as a "soft-IoU"

$$s(i,j) = \frac{\sum_{x,y} \min(\mathsf{m}_i(x,y), \mathsf{m}_j(x,y))}{\sum_{x,y} \max(\mathsf{m}_i(x,y), \mathsf{m}_j(x,y))},$$

and θ is a matching threshold (e.g. $\theta = 0.5$). In other words, we add an edge between two segments if their soft-IoU is greater than θ. By finding the connected component of this graph, we partition the segments into clusters $\mathcal{K} \subset \mathcal{V}$.

Cluster aggregation. After clustering the segments, we define a new set of masks and class probabilities, this time associated with clusters instead of segments. We denote these as $\hat{\mathsf{m}}_\mathcal{K}(x, y)$ and $\hat{\mathbf{p}}_\mathcal{K} = [\hat{p}_\mathcal{K}^1, ..., \hat{p}_\mathcal{K}^C]$, respectively, and compute them by simply averaging the masks and probabilities of all segments belonging to each cluster

$$\hat{\mathsf{m}}_\mathcal{K}(x,y) = \frac{1}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \mathsf{m}_i(x,y)\,,$$
$$\hat{\mathbf{p}}_\mathcal{K} = \frac{1}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \mathbf{p}_i\,.$$

Panoptic fusion. Given this new set of masks and class probabilities, we fuse them into a single overall panoptic prediction with an algorithm akin to the one used in the final stage of [6]. Specifically, we follow these steps:

1. For each cluster $\mathcal{K}$, we determine the most likely class $c_\mathcal{K}^* = \arg\max_c \hat{p}_\mathcal{K}^c$, and the corresponding probability $p_\mathcal{K}^* = \max_c \hat{p}_\mathcal{K}^c$.

2. We scale $\mathcal{K}$'s mask by $p_\mathcal{K}^*$ to obtain $\bar{\mathsf{m}}_\mathcal{K}(x, y) = p_\mathcal{K}^* \hat{\mathsf{m}}_\mathcal{K}(x, y)$.

3. We assign image pixels to clusters with the rule: (x, y) is assigned to $\mathsf{k}^*(x, y) = \arg\max_\mathcal{K} \bar{\mathsf{m}}_\mathcal{K}(x, y)$, and its confidence is set to $\mathsf{s}(x, y) = \max_\mathcal{K} \bar{\mathsf{m}}_\mathcal{K}(x, y)$.

At the end of this process, each pixel will have a class $c_{\mathsf{k}^*(x,y)}^*$ and a confidence $\mathsf{s}(x, y)$. Furthermore, pixels of thing classes can be partitioned into instances according to their cluster assignment $\mathsf{k}^*(x, y)$.

B. Rendering Performance

Tab. 3 compares the time taken to render a batch of 2048 rays for each method on an NVIDIA RTX A6000 GPU. Due to the hybrid representation from TensoRF, our model delivers a faster rendering performance compared to the baselines.

Method	Time to render 2048 rays
PNF [19]	119.7 ms
DM-NeRF [42]	66.5 ms
Semantic-NeRF [47]	65.7 ms
Panoptic Lifting (Ours)	**13.1** ms

Table 3. Time taken to render a batch of 2048 rays on a NVIDIA RTX A6000 GPU.

Method	HyperSim [31]	Replica [36]	ScanNet [9]
Mask2Former [6]	50.52	50.10	43.6
Panoptic Lifting (Ours)	**66.84**	**63.79**	**60.4**

Table 4. Conventional PQ scores on novel views from the test set.

C. Implementation Details

C.1. Panoptic Lifting

Panoptic Lifting uses TensoRF [4] for modeling the scene density and radiance. Specifically, we use the Vector-Matrix (VM) decomposition, with number of density and appearance components set to 16 and 48 respectively. The starting grid resolution is set to 128^3 and goes upto 192^3 at the end of the optimization. 27 color features are decoded with a tiny 2 layer MLP with positional encoding with 2 components to encode the view direction and the features.

To model the semantic class distribution and surrogate identifiers we make use of two small view-independent MLPs. The semantic MLP has 5 layers with a width of 256 and outputs a probability distribution over the target classes for any given input position. The surrogate identifier is a 3 layer MLP which generates a distribution over max k identifiers (set to 50 in our experiments). Neither of these MLPs use positional encoding. We choose to go with MLPs instead of Vector-Matrix decompositions for semantics and surrogate identifiers for memory size constraints. Our model is trained with a batch of 2048 rays, with a learning rate of 0.0005 for MLPs and 0.02 for the TensoRF lines and planes.

C.2. Baselines

We use the publicly available Mask2Former [6] code and models, without any retraining or fine-tuning. For all methods that use Mask2Former instance labels (including ours), instance counts are renumbered to be distinct across frames. For Semantic-NeRF [47] and DM-NeRF [42], we use their publicly released code. Since DM-NeRF outputs the labels as abstract instance identifiers, we create a map from instance to class using the instance's majority class across the train set as its assigned class.

Since Panoptic Neural Fields [19] does not provide a public implementation, we re-implement it based on details from the paper. We do not use their prior-based initialization since it requires additional 3D datasets for the instanced classes. In the original implementation, PNF uses a monocular 3D detector, which is essential when dealing with dynamic objects varying across frames. However, since the task here deals with static scene, it is more fair to use a multi-view detector for getting the bounding boxes. We use a state-of-the-art multiview detector [33] pretrained on ScanNet for getting object bounding boxes for PNF in our experiments. Note that for getting a reasonable 3D detector performance, it is required that the camera poses are scaled and centered similarly to the original ScanNet training data. We perform these pose corrections for Replica [36], Hypersim [31] and in-the-wild scenes. Since this correction requires an estimate of scale, we use for pose correction the ground-truth depth from Replica and Hypersim, and NeRF optimized depth for scenes in the wild. We further show result with a variant of PNF that uses ground-truth detections, except for in-the-wild data where not ground-truth is available.

All models (including ours) are trained with Mask2Former [6] generated labels.

D. Data

Tab. 6 shows the scenes and their corresponding number of frames. The available posed images are split into 75% views for training and 25% intermediately sampled test views. Note that for each of the datasets, the ground-truth semantic and instance labels are only used for evaluation, and are not used for training or refinement of any models.

Since the original model (swin_large_IN21k) was trained on COCO [22], and the labels for evaluation come from different datasets, we map the Mask2Former predictions as well as the ground-truth labels across all the datasets used in our experiments to ScanNet 21 classes (Tab. 7 left). For in the wild scenes, we use 31 ScanNet classes listed in Tab. 7 (right).

E. Additional Results

Tab. 4 reports the conventional PQ scores between our method and Mask2Former [6]. As mentioned in the main paper, this does not take into account the instance consistency across the scene, since matching between ground-truth and predicted instances is done on a per-frame basis. We further report $\mathrm{SQ}_{\mathtt{scene}}$ and $\mathrm{RQ}_{\mathtt{scene}}$ in Tab. 5.

Method	HyperSim [31]		Replica [36]		ScanNet [9]	
	SQ^{scene} ↑	RQ^{scene} ↑	SQ^{scene} ↑	RQ^{scene} ↑	SQ^{scene} ↑	RQ^{scene} ↑
DM-NeRF [42]	62.06	55.45	58.68	47.68	53.26	46.13
PNF [19]	55.33	47.51	53.62	44.10	62.96	50.73
PNF [19] + GT Bounding Boxes	68.23	53.35	62.15	50.81	70.01	55.87
Panoptic Lifting (Ours)	**70.35**	**64.32**	**69.10**	**63.61**	**73.50**	**64.95**

Table 5. SQ and RQ metrics for on novel views from the test set.

Dataset	Scene	# Frames
HyperSim	ai_001_003	100
HyperSim	ai_001_008	100
HyperSim	ai_001_010	300
HyperSim	ai_008_004	63
HyperSim	ai_010_005	100
HyperSim	ai_035_001	200
ScanNet	scene0050_02	874
ScanNet	scene0144_01	678
ScanNet	scene0221_01	780
ScanNet	scene0300_01	929
ScanNet	scene0354_00	563
ScanNet	scene0389_00	708
ScanNet	scene0423_02	855
ScanNet	scene0427_00	659
ScanNet	scene0494_00	740
ScanNet	scene0616_00	758
ScanNet	scene0645_02	726
ScanNet	scene0693_00	866
Replica	office_0	900
Replica	office_2	900
Replica	office_3	900
Replica	office_4	900
Replica	raw	900
Replica	room_0	900
Replica	room_1	900
Replica	room_2	900
In the wild	office	1100
In the wild	bed_room	1100
In the wild	meeting_room	1100

Table 6. Scenes used for evaluations in our experiments. Note that the in the wild scenes are only used for qualitative evaluation (shown in the supplementary video) since ground truth labels are not available for a qualitative comparison with baselines.

Class	Type
wall	Stuff
floor	Stuff
cabinet	Stuff
bed	Thing
chair	Thing
sofa	Thing
table	Stuff
door	Stuff
window	Stuff
counter	Stuff
shelves	Stuff
curtain	Stuff
ceiling	Stuff
refridgerator	Thing
television	Thing
person	Thing
toilet	Thing
sink	Thing
lamp	Stuff
bag	Thing
otherprop	Stuff

Class	Type
wall	Stuff
floor	Stuff
cabinet	Stuff
bed	Things
chair	Things
sofa	Things
table	Stuff
door	Stuff
window	Stuff
counter	Stuff
shelves	Stuff
curtain	Stuff
ceiling	Stuff
refridgerator	Things
television	Things
person	Things
toilet	Things
sink	Things
lamp	Stuff
bag	Things
bottle	Things
cup	Things
keyboard	Things
mouse	Things
book	Things
laptop	Things
blanket	Stuff
pillow	Things
clock	Stuff
cellphone	Things
otherprop	Stuff

Table 7. Classes and their type (*stuff* or *thing*) for dataset experiments (left) and in the wild experiments (right).