Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation

Abhijit Kundu [1] Kyle Genova [1] Xiaoqi Yin [1] Alireza Fathi [1] Caroline Pantofaru [1]
Leonidas Guibas [1,4] Andrea Tagliasacchi [1,3] Frank Dellaert [1,2] Thomas Funkhouser [1]

[1]Google Research [2]Georgia Tech [3]Simon Fraser University [4]Stanford University

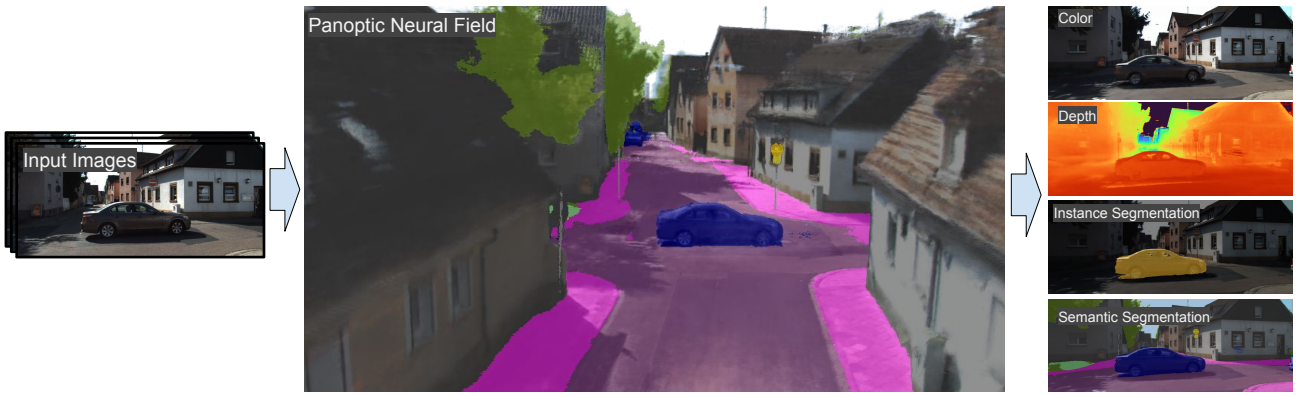

Figure 1. *Panoptic Neural Fields (PNF)* is an object-aware neural scene representation that decomposes a dynamic 3D scene into a set of objects (*things*) and background (*stuff*), each represented by a separate MLP based neural function. Our model predicts a panoptic-radiance field that represents the color, density, instance, and category label of any 3D point at any given time. The model is trained from RGB images alone and can describe dynamic challenging 3D scenes as shown above.

Abstract

We present Panoptic Neural Fields (PNF), an object-aware neural scene representation that decomposes a scene into a set of objects (things) and background (stuff). Each object is represented by an oriented 3D bounding box and a multi-layer perceptron (MLP) that takes position, direction, and time and outputs density and radiance. The background stuff is represented by a similar MLP that additionally outputs semantic labels. Each object MLPs are instance-specific and thus can be smaller and faster than previous object-aware approaches, while still leveraging category-specific priors incorporated via meta-learned initialization. Our model builds a panoptic radiance field representation of any scene from just color images. We use off-the-shelf algorithms to predict camera poses, object tracks, and 2D image semantic segmentations. Then we jointly optimize the MLP weights and bounding box parameters using analysis-by-synthesis with self-supervision from color images and pseudo-supervision from predicted semantic segmentations. During experiments with real-world dynamic scenes, we find that our model can be used effectively for several tasks like novel view synthesis, 2D panoptic segmentation, 3D scene editing, and multiview depth prediction.

1. Introduction

The ability to understand the content within an image is an essential task in computer vision, and over time we have witnessed a rapid increase in task complexity. Over a short period of time, we have progressed from the task of identifying the overall presence of objects within an image (i.e. classification [28] and object detection [18, 22]), to fine grained pixel-by-pixel classification (i.e. semantic segmentation [35, 51]), and to the ability to differentiate between object instances of the same class (*i.e.* panoptic segmentation [6, 26]).

However image level representations described above have limited applications. Instead we are interested in full 3D scene understanding which is important for autonomous driving [23], semantic mapping [62], and many other applications involving navigation or operation in the physical world [7]. Given a sequence of RGB images, our goal is to infer: 1) a 3D reconstruction of the observed geometry, 2) a radiance field of the scene, 3) a decomposition of the scene into potentially dynamic things (e.g., cars) and background stuff (e.g., grass), 4) a category and instance label for every 3D point, as illustrated in Figure 1.

In recent years, neural 3D scene representations like

NeRF [37] have made significant advancements [59, 64]. NeRF represents a scene using a multi-layer perceptron (MLP) that maps positions and directions to densities and radiances which can then be used to synthesize an image from a novel view. However NeRF lacks semantic understanding and is also not object aware. In this work we explore neural scene representations for semantic 3D scene understanding tasks beyond the usual view synthesis task.

Some recent work augments NeRF to infer semantics [70], adding an extra head to predict semantic logits for any 3D position along with the usual density/color. Other recent work decomposes a scene into a set of NeRFs associated with foreground objects separated from the background [20, 44, 65]. However, these systems have several limitations in the context of our goals: 1) they do not produce panoptic segmentations, 2) they learn from scratch for every scene; and 3) they share MLPs for multiple objects, which limits their ability to reproduce specific instances.

We address these issues in our proposed Panoptic Neural Fields (PNF), an object-aware neural scene representation that explicitly decomposes a scene into a set of objects (*things*) and amorphous *stuff* background. Each object instance is represented by a *separate* MLP to evaluate the radiance field within the local domain of a potentially moving and semantically labeled 3D bounding box. The semantic-radiance field of the *stuff* background is also represented by a MLP which includes an additional semantic head. Together the *stuff* and *things* MLPs jointly define a panoptic-radiance field that describes the density, color, category, and instance label of any 3D point over time.

Our object aware representation makes it possible to describe scenes with multiple moving objects and also paves the way to incorporate constraints that objects of the same category have similar shape and appearance. Previous object-aware frameworks [44, 65] used a shared MLP with instance-specific latent codes to incorporate this prior. In our model, each object instance is represented by a *separate* MLP that is initialized with a category-specific prior using meta-learning. The separation of learning of object category priors via meta-learning makes it possible to represent instance-specific details with smaller MLPs, which speeds inference in scenes with many objects.

Given a collection of images captured from a scene, we employ off-the-shelf algorithms to predict camera parameters [39] and 2D semantic segmentations [6] for all images, plus a set of 3D object detections with 3D oriented bounding boxes and category labels [45]. We initialize the weights of the MLPs for our panoptic neural field model either with object category-specific meta-learned initialization or simple biased initialization of density activation layers. We then jointly optimize the bounding box and MLP parameters to minimize *analysis-by-synthesis* style losses that measure differences in color and semantic images synthesized with volumetric rendering (as in NeRF [37]). Thus, our approach provides an unified framework for optimizing 3D shape, appearance, semantics, and object poses all from a set of color images.

We evaluate our method on several scene understanding and synthesis tasks using experiments on the KITTI [16] and KITTI-360 [33] dataset, including 3D panoptic reconstruction, and scene editing. The output panoptic-radiance field can also be used to synthesize 2D image-level outputs like semantic segmentation, panoptic segmentation, depth images, and colored images of both observed and novel views. We demonstrate the utility of the proposed method for these scene understanding tasks, as well as for novel-view synthesis method with movable scene components.

Our contributions can be summarized as follows:

- We propose, to the best of our knowledge, the first method that can derive a panoptic-radiance field of complex dynamic 3D scenes from images alone.
- Our single unified model achieves state-of-the-art quality across multiple tasks and benchmarks on KITTI and KITTI-360 datasets.
- We incorporate object shape and appearance priors via category-specific meta-learned initialization. This allows our object MLPs to be much smaller and faster than previous object-aware representations.
- We jointly optimize all (*stuff* and *things*) neural fields and object poses, allowing our method to cope with noisy object poses and image segmentations.

2. Related Work

The most relevant related work is summarized in Table 1 and can be broadly divided into three categories: (1) Learning based single image 3D semantic and/or instance segmentation, (2) Multi-view 3D reconstruction and segmentation methods, and (3) Neural fields.

Single image reconstruction and segmentation. 3D-RCNN [30] and Mesh-RCNN [17] takes as input a single RGB image and predicts 3D mesh and pose of object instances in the image. Total3DUnderstanding [43] combines layout estimation, 3D object detection, and object mesh generation. More recently [8] showed 3D panoptic reconstruction and segmentation from a single RGB image.

Multi-view reconstruction and segmentation: Incorporating semantics into SLAM and SfM systems has a long history [2, 21, 29, 54]. More recently, PanopticFusion [41] is an incremental, online mapping approach that fuses a sequence of RGB-D images into a consistent panoptic segmentation. ATLAS [40] reconstructs and labels the 3D geometry from *multiple* posed RGB images. However, ATLAS produces only semantic segmentations (without instances). Both PanopticFusion and ATLAS works only for

Paper	Sem	Obj	Pan	Dyn	Opt	Syn
MeshRCNN [17]		✓				
Total3D [43]	✓	✓				
Atlas [40]	✓					
SLAM++ [54]		✓.				
PanopticFusion [41]	✓		✓			
Kimera [52]	✓					
DynSceneGraphs [53]	✓	✓	✓	✓	✓	
SemanticNerF [70]	✓					✓
NSG [44]		✓		✓		✓
ObjectNeRF [65]		✓				✓
PNF (Ours)	✓	✓	✓	✓	✓	✓

Table 1. Comparison to properties of related work. The check marks indicate which prior methods have the following capabilities: "Sem" = performs semantic segmentation; "Obj" = performs object decomposition; "Pan" = performs panoptic segmentation; "Dyn" = handles dynamic objects; and "Opt" = optimizes object bbox parameters. "Syn" = allows for novel view synthesis.

static scenes, requires 3D supervision, and relies upon convolutions on a discrete voxel grid, which limits its resolution. Kimera [52] takes a stereo sequence and does online reconstruction, meshing, and semantic labeling of the mesh using *ground-truth labels*, as a proxy for any 2D segmentation method. Dynamic Scene Graphs [53] expands on that by inferring object instances, even dynamic ones in case of people. Both methods, though representing impressive systems, were only demonstrated in simulation and rely on ground truth semantic labels.

Neural radiance fields (NeRFs). This work builds upon NeRF [37], which represents a scene using a multi-layer perceptron (MLP) that maps positions and directions to densities and radiances. From that representation, novel views can be synthesized using volumetric rendering and compared to input views in a self-supervised optimization procedure to infer MLP weights for an observed scene. However, NeRF only works for static scenes and trains for hours (from scratch) for every set of input views.

NeRFs with semantics. Recent work has considered using neural representations to infer semantics [70]. In particular, SemanticNeRF [70] adds an extra head to NeRF to predict semantic labels for any 3D position along with the usual density and color. Concurrent to our work [12] also demonstrated neural panoptic fusion from multiple views. However both of these work are not object aware and cannot handle dynamic scenes.

NeRFs with dynamics disentangle a scene into a canonical volume and its time-varying deformation, represented by a second MLP. This approach has been applied for deforming faces [13, 15, 49], moving human bodies [47, 57, 60], and objects [10, 32, 46, 48, 63]. In contrast, we consider dynamic scenes that contain many moving objects.

NeRFs with object decompositions [20, 44, 65] decompose a scene into a set of NeRFs associated with foreground objects separated from the background. ObjectNeRF [65] uses the object branch to render rays with masked areas for foreground objects conditioned on a latent code. Similarly, Neural Scene Graphs (NSGs) [44] uses a separate conditional NeRF for each object category, and a multiplane neural representation for the background. However, these systems have several limitations in the context of our goals: 1) they do not produce panoptic segmentations, 2) they learn from scratch for every scene; and 3) they share NeRFs for multiple objects (which limits the ability to reproduce specific discrete instances).

Conditional NeRFs infer latent codes as pioneered in GRAF [55], piGAN [4], and PixelNeRF [66], as well as the recent CodeNeRF [24], which also optimizes over object poses. All these works incorporate category-specific priors by sharing MLP weights across object instances, combined with instance specific codes. We instead use instance-specific MLPs for representing each object, which allows each MLP to be smaller, resulting in faster inference speed on scenes with multiple objects. Object appearance and shape priors are incorporated via category-specific meta-learned initialization [11, 25, 42, 58] of the MLP weights.

3. Method

This section introduces the panoptic neural field representation and our computational pipeline (Figure 2). In Sec. 3.1 we describe the representation itself, which stores a panoptic-radiance field that can be used to query the color, density, semantic and instance labels at any 3D point at any time. In Sec. 3.2, we describe how this panoptic-radiance field can be rendered using NeRF-style volume rendering by *over* compositing along sampling of points along rays. In Sec. 3.3 we explain how the model is trained in analysis-by-synthesis style by comparing the rendered color and semantic segmentation with observed 2D color and predicted 2D semantic labels.

A key difference between our framework and previous object-aware frameworks [44, 65] is how we train and represent *things*. As illustrated in Fig. 3, our framework uses instance-specific fully weight encoded functions to represent each object, in comparison to the traditional approach of using a shared MLP with instance-specific latent codes. This design choice is driven by several factors. First, since the MLP only needs to represent a single object instance, we can have a smaller MLP compared to shared MLPs, resulting in faster inference speed on scenes with multiple objects. Second, this allows the object MLP to use its full capacity to describe and overfit to a specific novel object instance, which may not be possible [9] with latent encoding. Third, it is simpler and does not require any change to the core NeRF model architecture. Object-level priors can also be incorporated to our instance-specific models using

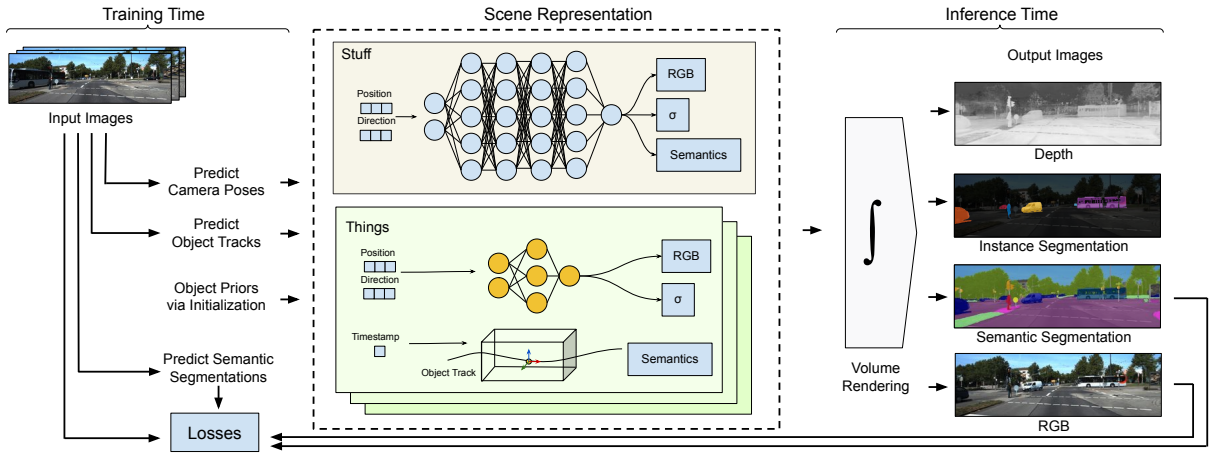

Figure 2. Overview of *Panoptic Neural Feld (PNF)* representation being learned from input color images. The background *stuff* is represented by an MLP that produces RGB, density, and semantic logits, while each object is represented by a dynamic track and a smaller individual MLP. Once trained the representation can be used for several tasks by simple volume rendering.

meta-learning based initialization (See Sec. 3.4).

3.1. Scene Representation

The core of our framework is the panoptic-radiance field representation. This representation accepts input queries consisting of a point position $\mathbf{x} \in \mathbb{R}^3$, view direction $\mathbf{d} \in \mathbb{R}^3$, and time $\mathrm{t} \in \mathbb{R}$. The outputs of a query are color, density, a semantic label, and an instance label. This field is the composition of multiple distinct neural functions. There are separate fields for each 3D object (*things*), and another, larger field for the background (*stuff*). The field associated with one object is defined inside a mobile 3D oriented bounding box. The background is represented by another neural function defined inside a larger scene bounding box. It encodes density, appearance, and semantic labels.

Things: Foreground objects in our representation are represented by a neural function inside a dynamic bounding box. To instantiate the set of object tracks in a scene, we first run an RGB-only 3D object detector [45] and tracker [61]. This provides a bounding box track T_k and semantic class for each recognized object instance k. The track is parameterized by a sequence of transformation matrices, one at each of at a set of discrete timestamps. For each timestamp, we create a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. There is also a box extent $\mathbf{s} \in \mathbb{R}^3$ along each axis that is time-invariant. To determine the coordinate frame of an object at an arbitrary real-valued timestamp, we interpolate the discrete track steps.

For each object instance, we instantiate a separate time-invariant MLP with the standard NeRF architecture [37]. Its weights are initialized using the techniques described in Section 3.4. To query this MLP, positions and directions are transformed from the world frame to the bounding box frame defined by the track at the current timestamp. We optimize all parameters of the MLP and object track T_k jointly. Optimizing object track parameters is important as initial boxes (even GT boxes) may be noisy. In order to optimize rotation, we orthogonalize $\mathbf{R}$ after each gradient descent step using SVD, which projects it back onto $\mathrm{SO}(3)$.

Stuff: We represent the static background *stuff* with a single neural function. In addition to predicting density and color at every 3D point, the *stuff* function also learns a semantic label per point. We again use an MLP to represent the learned function. The architecture is similar to NeRF, but with an additional head for semantic logits. This head is direction-invariant to encode the inductive bias that 3D points have multi-view consistent semantic labels. Note that unlike the MLPs for objects, which are bounded, the *stuff* MLP must handle the unbounded nature of real world scenes. Therefore for large scenes, we follow [67] and use a separate foreground and background *stuff* MLPs.

Panoptic-Radiance Field: The final panoptic-radiance field at a 3D point is computed from aggregating the contributions of the individual *thing* and *stuff* MLPs. For any given output channel (color, density, etc.), our function takes the sum of all contributions from any bounding box hits, defaulting to the *stuff* output if there is no intersection. For the color field $\mathbf{c}$, this is:

$$\mathbf{c}(\mathbf{x} \,|\, \boldsymbol{\theta}) = \mathbb{1}_S(\mathbf{x})\mathbf{c}_s(\mathbf{x} \,|\, \boldsymbol{\theta}) + \sum_k \mathbf{c}_k(\mathbf{T}_k^{-1}\mathbf{x} \,|\, \boldsymbol{\theta}) \quad (1)$$

where $\mathbb{1}_S$ is 1 if and only if the point intersects no bounding boxes, $\mathbf{c}_s$ is the *stuff* color field, and $\boldsymbol{\theta}$ denote the MLP weights for *stuff* and *thing* MLPs. For other fields, we simply substitute the radiance $\mathbf{c}$ with the density, semantic, or instance function. Object boxes contribute a one-hot semantic logit vector for their class, which handles the merging of *stuff* and *thing* semantics. Similarly, the instance label function is a vector of length K, with one dimension per detected object k. Objects contribute a one-hot vector for

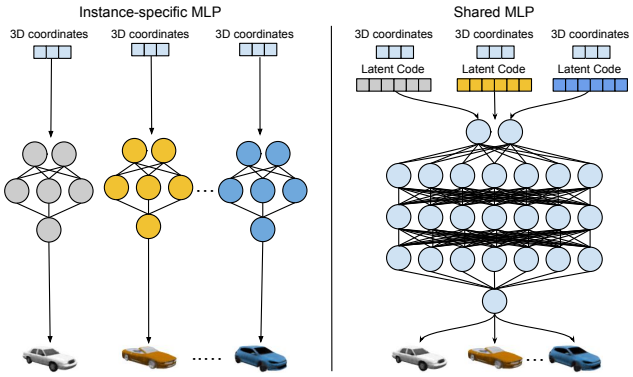

Figure 3. Our framework uses instance-specific fully weight encoded functions to represent each object (left), in comparison to a traditional approach of using one shared MLP with instance-specific latent codes (right). Instance-specific MLPs can be smaller, since they just need to have enough capacity to express a single object instance, as compared to a shared MLP. On scenes with multiple objects our approach can be significantly faster.

their instance while the *stuff* function's instance output is always zero.

3.2. Rendering Panoptic-Radiance Fields

Given the complete panoptic-radiance field representation, a 2D image can be synthesized with volume rendering. This process is described in more detail in NeRF [37]. Our image synthesis approach is the similar to NeRF, with the addition of support for extra output channels and dynamic boxes. To render a single ray $\mathbf{r} = \mathbf{o} + t\mathbf{d}$ we uniformly sample $N = 1024$ points $\mathbf{x}$ (with jitter) along the ray and alpha-composite the result of the output channel C we wish to render (RGB, depth, semantic, instance):

$$C(\mathbf{r} \,|\, \boldsymbol{\theta}) \approx \sum_{i=1}^{N} w(t_i) f(\mathbf{r}(t_i) \,|\, \boldsymbol{\theta}). \tag{2}$$

Above, $w(t)$ is the final weight associated with each sample, determined by *over* compositing the opacity values of each sample along the ray. The function f returns the representation value for the channel in question at the query point. For semantics, this is logits, while for instance it is a one-hot encoding of the object instance identifier k.

3.3. Model Losses and training

We jointly optimize all network parameters θ and object tracks $\mathbf{T}$ to reproduce the observed RGB images and predicted 2D semantic images:

$$\underset{\boldsymbol{\theta}, \mathbf{T}_{k,i}}{\arg\min} \; \mathcal{L}_{\text{rgb}}(\boldsymbol{\theta}, \mathbf{T}_{k,i}) + \mathcal{L}_{\text{sem}}(\boldsymbol{\theta}, \mathbf{T}_{k,i}) \tag{3}$$

At each gradient descent step, we randomly sample mini-batches of rays. Our RGB loss is the mean squared error between the synthesized and ground truth color, summed over

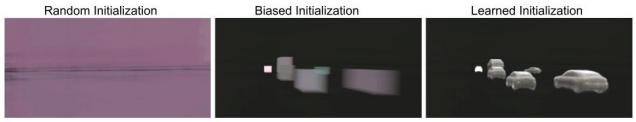

Figure 4. This figure illustrates different initialization schemes of instance-specific MLP weights for "things" in the car category.

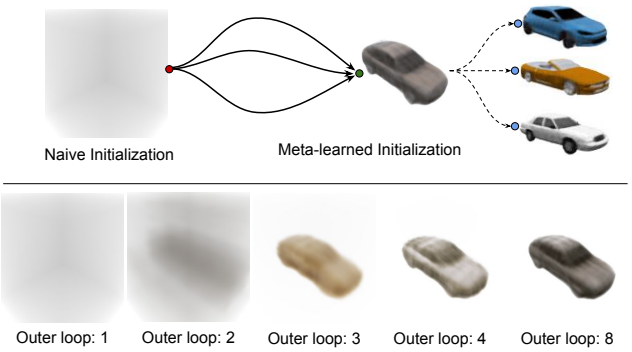

Figure 5. Category specific learned initialization with federated averaging. **Upper:** Rather than random initialization of MLP weights, we use meta-learning implemented using FedAvg [25,36] to find a good category specific initialization. This allows improves generalization and convergence behavior. **Lower:** We visualize the learned initialization model for the car class with increasing outer loops of the federated averaging algorithm.

sampled rays as in NeRF [37]. Our semantic loss is applied at the same pixel locations, and compares the synthesized semantics with the input 2D semantic segmentation prediction [6]. For this loss, we apply a per-pixel softmax-cross entropy function rather than mean squared error.

3.4. Incorporating priors via initialization

One of the core benefits of an object aware approach, is the ability to incorporate inductive bias that objects instances within same category, often have similar 3D shape and appearance. One possible way to incorporate such priors is to have shared MLP weights across all object instances, combined with some instance specific codes. Our framework instead uses separate MLPs for representing each object instance. As illustrated in Fig. 3, this allows each MLP to be smaller as it only needs to represent a single object instance, resulting in faster inference speed on scenes with multiple objects. Object shape and appearance priors are instead incorporated via initialization of the MLP weights of the neural functions. We present two approaches (see Fig. 4) of initializing our model, one based on category specific meta-learning and another based on simple bias initialization of the activation function of the MLPs.

Biased initialization: This simple initialization scheme improves convergence behavior and training performance without requiring a large dataset from which to learn a shape

Figure 6. Images of color, depth, instance segmentation, and semantic segmentation rendered on dynamic KITTI scenes from our model trained **only** from RGB images.

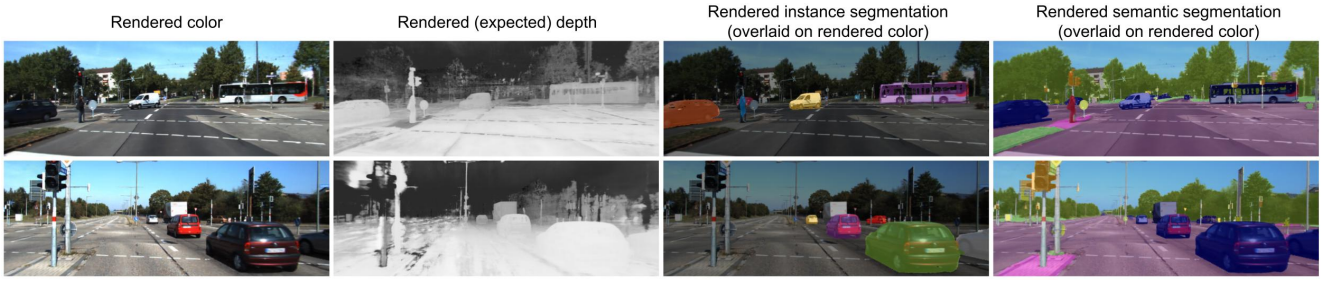

Figure 7. Rendered color, depth, instance segmentation, and semantic segmentation images of our models representing dynamic KITTI scenes with various object categories. These results used KITTI **provided** object tracks. Note that our instance specific object MLPs can also reconstruct novel object categories like *truck* and *bus*.

prior. In real-world outdoor scenes, most of the *stuff* volume is empty space. By contrast, most of the volume inside each *things* object bounding box is non-empty. We incorporate this prior directly by biasing the density prediction layer of the MLPs. For the *stuff* MLP, we initialize our bias to -5.0, whereas for all *thing* MLPs, we initialize the final bias layer to 0.1. Furthermore, for all MLPs, we use the *softplus* activation for the fully connected layer predicting the density outputs [69]. We have found that this simple injection of prior knowledge via initializing the bias values is quite effective and robust compared to random initialization, since dense content in *stuff* can suppress gradients from distant objects.

Category specific learned initialization: If a large shape collection is available for certain object categories, we can further improve on the bias initialization scheme. In particular, we use meta-learning [11] to capture category-specific shape and appearance priors. This process is illustrated in Fig. 5. First, we meta-learn category-specific initial weights by pre-training on ShapeNet [5]. Then, we use these weights to initialize the *thing* MLPs in our model when training on a novel scene.

To meta-learn a category-specific shape prior, we use the `FedAvg` [36] algorithm. This algorithm is known to be equivalent [25] to REPTILE [42] meta-learning, used before for NeRF initialization in [58]. To do one meta-step of `FedAvg`, we independently optimize a set of MLPs, each on a separate ShapeNet shape. We then average the model weights across all NeRF models, and start another meta-step. Fig. 5 visualizes the evolution of the learned initialization across several outer loops of `FedAvg`. In our experiments, we pre-train using the 2D rendered car images [56] of ShapeNet [5] to obtain the learned initialization model. When reconstructing a full scene, we then initialize the MLPs for each car instance track $\mathbf{T}$ to this initialization.

4. Evaluations

We performed a series of experiments to evaluate our model on multiple computer vision tasks, including view synthesis, reconstruction, 2D panoptic segmentation, 2D depth prediction, and scene editing. See the supplemental video for comprehensive visualizations of our results. All experiments used either the KITTI [16], Virtual KITTI [3,14] or the recent KITTI-360 [33] datasets. These datasets involves difficult forward facing cameras in complex outdoor dynamic scenes. KITTI-360 is the first benchmark that evaluates both the task of synthesising color and appearance images from novel views. Our model outperforms every other method in that leaderboard for both tasks as shown in

Method	Semantics mIoU	Appearance PSNR
NeRF [37] + PSPNet [68]	53.01	21.18
FVS [50] + PSPNet [68]	67.08	20.00
PBNR [27] + PSPNet [68]	65.07	19.91
Mip-NeRF [1] + PSPNet [68]	51.15	21.54
Ours	**74.28**	**21.91**

Table 2. Results on novel view color and semantic synthesis tasks on KITTI-360 [33] dataset. Rendered semantic segmentation and color images from our model is the best performing method for both the tasks in KITTI-360 leaderboard.

	SRN [56]	NeRF [37]	NeRF + time	NSG [44]	Ours
PSNR ↑	18.83	23.34	24.18	26.66	**27.48**
SSIM ↑	0.590	0.662	0.677	0.806	**0.870**

Table 3. Comparison of image reconstruction quality in dynamic KITTI scenes following the the experiement setup of NSG [44].

Tab. 2. Since scenes in KITTI-360 chosen for the tasks are all static, we also evaluate our model on dynamic scenes from the KITTI [16] dataset. Rendered views from these dynamic scenes are shown in Fig. 6 and Fig. 7. Additional results are available in Appendix B. Below we evaluate our model for each task in more detail.

Novel View Synthesis: How well a particular representation describes a scene is reflected in the quality of rendered views. As shown in Tab. 2, color images rendered by our model achieves the best PSNR and is competitive with latest view synthesis models [1, 27, 37, 50]. Since the scenes used in KITTI-360 [33] novel view synthesis task are all static, we attribute the improved performance to our model benefiting from separate object-aware MLPs and incorporation of category-level priors. To study the novel view synthesis capabilities of our model on dynamics scenes, we also experiment on several dynamic scenes from KITTI dataset as shown in leftmost columns of Fig. 6 and Fig. 7. Notice that the rendered color images accurately captures the moving vehicles in the scene. A quantitative analysis of synthesized colored images for dynamic KITTI scenes is available in Tab. 3, wherein we follow the same experimental setup as described in Sec. 5.2 of Ost *et al.* [44]. As expected, our method significantly outperforms representations like SRN [56] and NeRF [37] which rely on static world assumption. Note that our method also outperforms NSG [44] even though our instance specific object MLPs are much smaller (10× fewer FLOPs) compared to those used in NSG [44]. The improvement over NSG also demonstrates the advantage of incorporating category-specific priors derived from meta-learning with images from ShapeNet.

Panoptic Segmentation: Semantic and instance segmentation of an arbitrary view can be obtained from our model by simple rendering (see Eq. (2)) along the desired view. The two right columns of Fig. 6 and Fig. 7 demonstrates the rendered semantic and instance segmentation images from our model. As shown in Tab. 2, our model achieves *state-of-the-art* 74.28 mIoU for the task of novel

Method	mIoU	PQ
2D Deeplab [6] on ground-truth RGB	49.9	43.2
2D Deeplab [6] on NeRF rendered RGB	32.1	24.9
Ours without *thing* MLPs (≈*SemanticNeRF*)	45.3	-
Ours	**56.5**	**45.9**

Table 4. Ablation study of the rendered segmentation quality on KITTI. Our model achieves better segmentation in terms of mean IoU, panoptic quality (PQ) [26]. Also notice that without *thing* MLPs, the segmentation quality is significantly worse and misses all dynamic objects. This demonstrates the advantage of our model over non object-aware representations like *SemanticNeRF* [70].

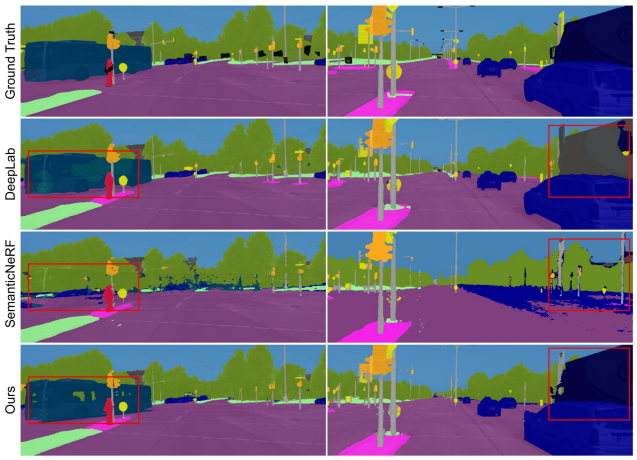

Figure 8. Comparison of semantic segmentation output from Panoptic-DeepLab [6] on ground-truth RGB image, SemanticNeRF [70], and the semantic segmentation rendered from **our** model from the same view. Segmentation produced by **our** model is significantly better as highlighted by the red boxes in the figure.

view semantic segmentation on KITTI-360. One approach of generating segmentation images for any arbitrary view is to first synthesize color image from a desired view using view synthesis methods like [1, 27, 37, 50]; followed by 2D image segmentation [6, 68]. However as demonstrated in Tab. 2, our unified model significantly outperforms all such two-stage baselines. Moreover the rendered segmentation images from our model are temporally consistent and works for dynamic scenes. We also perform a ablation study of the rendered segmentation images on dynamic scenes from KITTI. Quantitative and qualitative results are shown in Tab. 4 and Fig. 8 respectively. Our model significantly out performs (+9.2 mIoU) non object-aware models like SemanticNeRF [70], since they cannot model dynamic objects. Our model also improves upon single image state-of-the-art segmentation models [6] by fusing information from multiple views.

2D Depth Estimation: We also demonstrate rendered depth images from our model, obtained by over compositing point depths along rays with opacity values as described in Eq. (2). Rendered depth images are shown in the second

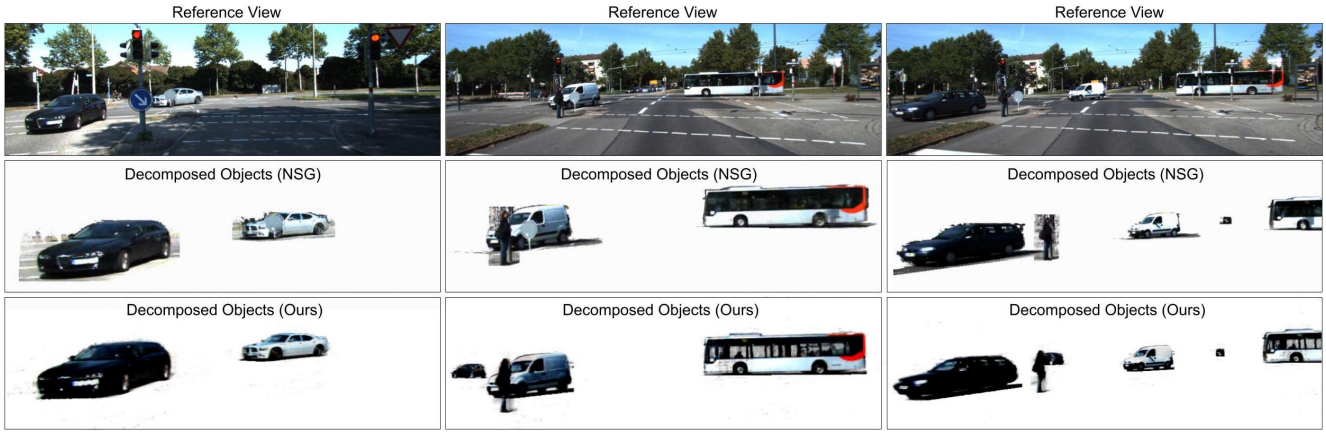

Figure 9. Scene decomposition comparison. Top row: reference views. Middle and Bottom: renderings of objects (without background stuff) of NSG [44] and PNF (ours) model respectively. Note that the traffic sign-posts in front of the cars are entangled with the rendered cars in NSG, but not in our results. Also the windows of the bus are correctly reconstructed as translucent by our model.

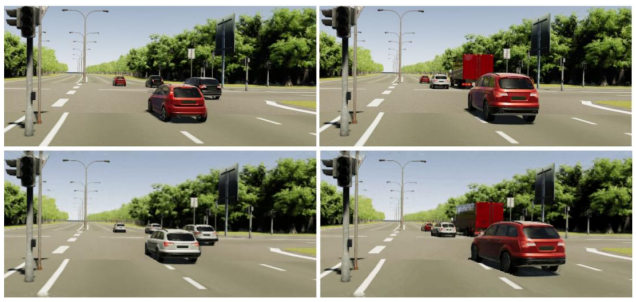

Figure 10. Scene editing. We manipulate input ground truth images from KITTI Virtual [14] (top) by cloning all cars to be the same (lower left) or changing their orientations (lower right).

columns of Figs. 6 and 7, along with other outputs of the model. Please note the sharp reconstruction of shapes of moving cars on the road. Of course, those cars would be missing or blurred in standard NeRF models. This demonstrates the benefits of our object-aware approach that handles dynamic scenes and instance-specific object MLPs that accurately captures each object instance.

Object Decomposition: Fig. 9 shows visualizations of how images from a dynamic KITTI scene are decomposed into MLPs representing different object instances. These images are rendered without the (stuff) background. Compared to NSG [44], our method does a better job of disentangling objects from the (stuff) background. In particular, note that the occluding traffic sign-posts in front of the car are entangled with the rendered cars in NSG, but not in our results. Better object decomposition from our model is also important for the scene editing task discussed below.

Scene Editing: Since our model separates objects from the background and builds a full 3D radiance field for each object, it is possible to edit images using the model by removing objects, adding new objects, and transforming object bounding boxes and poses. Fig. 10 shows few scene editing examples on Virtual KITTI dataset. The top row shows original images, and the bottom row shows edits. In bottom-left, we demonstrate *cloning* of cars by replicating the weights of all object MLPs to a same car. In bottom right of the figure we independently rotate each vehicle object.

5. Limitations

Like most other NeRF-style methods, our model is compute-intensive and hence currently only suited for offline applications. However, we expect advances in neural rendering [38, 59] will alleviate some of these speed issues in near future. It also does not incorporate more complex light transport effects, such as shadowing, under object motion. Our framework optimizes and corrects bounding box poses from noisy 3D object detection and tracking, but has not been designed to handle other errors such as missing and duplicate detections and incorrect class predictions. Finally, our framework does not handle deformable objects and is restricted to scenes with rigid moving objects.

6. Conclusion

This paper presents Panoptic Neural Fields (PNF), an object-aware neural scene representation that decomposes a scene into a set of MLPs associated with object instances (*things*) and the background (*stuff*). Our model learns a 4D panoptic radiance representation of dynamic scenes from images alone. This representation can be queried to obtain the color, density, instance, and category label of any 3D point over time. Several tasks like scene editing, view synthesis, panoptic segmentation are derived by simply rendering the representation from the desired views. Results of experiments on several KITTI scenes demonstrate state-of-the-art performance for novel view synthesis and panoptic segmentation for challenging outdoor scenes with multiple dynamic objects.

A. Additional Model Details

In this section we provide additional details for training and inference of the proposed panoptic neural field model described in Sec. 3.

A.1. Network Architecture and Training Details

For *stuff* MLP, we use 8 hidden layers of width 256. For *thing* MLP, we use MLP with 4 hidden layers and width 128. We use positional encoding of $L = 10$ frequencies to encode position coordinates for *stuff* MLP and $L = 6$ frequencies to encode position coordinates in object coordinate space for each *thing* MLP.

The density and semantic output does not depend on view directions, whereas the color output is additionally conditioned on view directions similar to [37]. This encodes the assumption that structure and semantics of the static background and individual objects only varies with position coordinates in their respective coordinate spaces. We use $L = 2$ frequencies to encode the view directions before feeding them to the *stuff* and *thing* MLPs. For view directions, we find it beneficial to gradually activate the encoding frequencies over the course of the optimization, similar to [34, 46].

To conserve memory we do not perform hierarchical sampling [37]. Instead we sample additional points with stratified sampling strategy along rays while training and inference. So unlike [37] which uses a pair of MLPs corresponding to coarse and fine sampling, only one set of MLPs are required in our model. We used 1024 samples per ray in our experiments on KITTI dataset.

As described in Sec. 3.3, we use both color loss and semantic loss. Since semantic loss is realized with softmax cross-entropy function compared to mean squared error function to realize the RGB (color) loss, we scale the semantic loss lower to ensure that the semantic loss does not dominate the training process. Apart from the color loss and semantic loss, the optimization objective also enforces the constraint that rotation component $\mathbf{R} \in \mathbb{R}^{3\times3}$ of the optimized tracks are valid $\mathrm{SO}(3)$ rotation matrices using the SVD orthogonalization [31] method. We use Adam as our optimizer for all our training routines including optimizing Panoptic Neural Field model to a new scene and also during inner loops of `FedAvg` based meta-learning (see Sec. 3.4) for learned initialization.

A.2. Model Initialization Details

As described in Sec. 3.4, we incorporate priors via initialization. For *cars* and *vans*, we use category specific initialization of *thing* MLP weights. This category specific initialization is meta-learned from 2D rendered images of 3D *cars* models from ShapeNet dataset. For all other object categories, we use the biased initialization described in Sec. 3.4. We use a simplified federated averaging algorithm [36] as described in Algorithm 1 to realize our category specific learned initialization. Also see Fig. 5 which visualizes the evolution of the learned initialization across several outer loops of `FedAvg`. This algorithm is known to be equivalent [25] to REPTILE [42] meta-learning. The main advantage of the simple `FedAvg` is that it allows decentralized federated training. In Fig. 5, we assumed simple SGD as the optimization algorithm, but can be easily adapted for other optimizers.

Algorithm 1 `FedAvg`. The K clients each working on images of an unique object instance are indexed by k; $\mathcal{D}_k$ is the data (bundle of rays corresponding to every observed pixel) for object instance k; B is the local minibatch size *i.e.* number of rays used per batch for inner epochs on each client; E is the number of inner (local) epochs, and η is the learning rate; the network parameters (MLP weights) is denoted by θ_t after t outer epochs on the server.

procedure SERVERUPDATE
  initialize θ_0
  for each outer epoch $t = 1, 2, \ldots$ **do**
    for each client $k \in 1, 2, \ldots, K$ **in parallel do**
      $\theta^k_{t+1} \leftarrow$ CLIENTUPDATE(k, θ_t)
    end for
    $\theta_{t+1} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \theta^k_{t+1}$
  end for
end procedure

function CLIENTUPDATE(k, θ)
  $\mathcal{B} \leftarrow$ (split $\mathcal{D}_k$ into batches of ray bundles of size B)
  for each inner epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $\theta \leftarrow \theta - \eta \nabla \ell(\theta; b)$
    end for
  end for
  return θ to server
end function

B. Additional Results

In this section we provide additional results on KITTI [16] and KITTI-360 [33] for the tasks of novel view synthesis of color, semantics and depth images, along with scene editing. We also evaluate the benefits of our proposed category specific meta-learned initialization of *thing* MLP weights on the ShapeNet [5] dataset.

B.1. Novel view renderings

Additional qualitative results of our model on KITTI [16] and KITTI-360 [33] are shown on Fig. 12 and Fig. 11 respectively. Just like our experiments in Sec. 4, we only

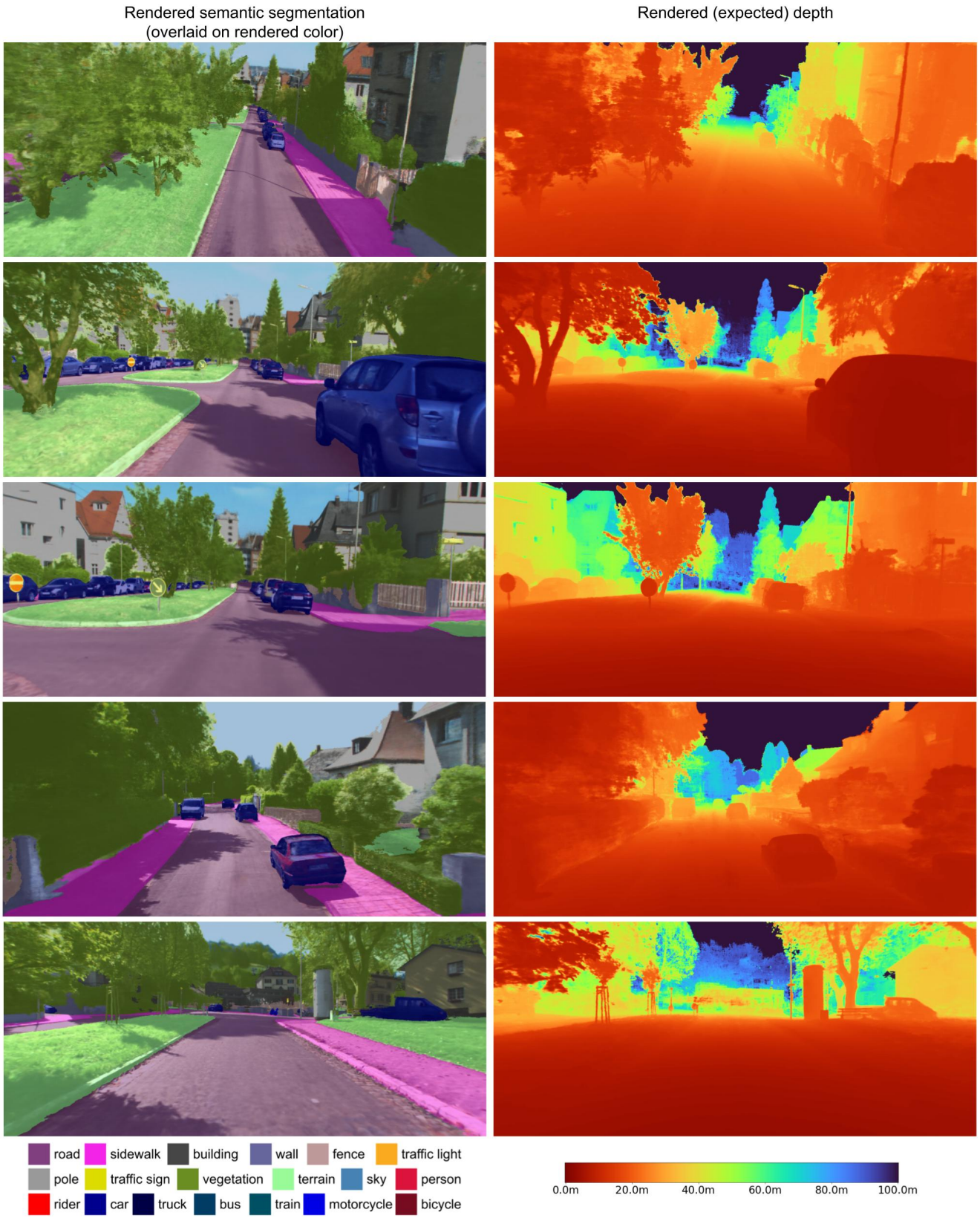

Figure 11. Rendered novel views (semantic segmentation and depth) from our Panoptic Neural Field models trained on KITTI-360 [33] scenes. Only the forward facing cameras (captured at ≈5Hz) are used for these results. Note that even thin structures like lamp poles and sign-posts are accurately reconstructed and segmented.

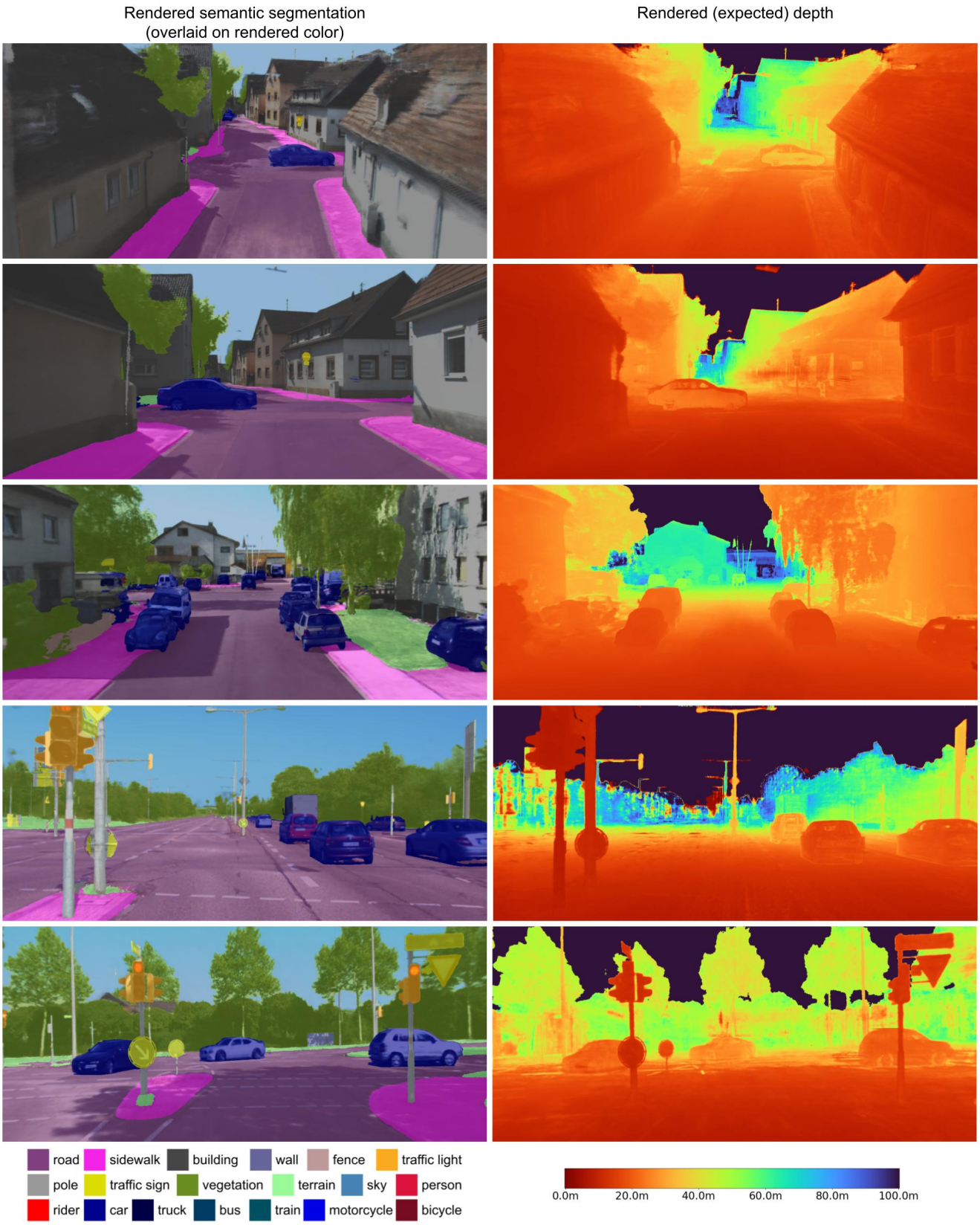

Figure 12. Rendered novel views (semantic segmentation and depth) from our Panoptic Neural Field models trained on KITTI [16] scenes. Only the forward facing cameras (captured at $\approx$ 10Hz) are used for these results. Note that several of the scenes have moving cars but are still accurately reconstructed and segmented.

Figure 13. Scene editing results. The left column shows rendered color images along novel viewpoints of our panoptic neural field representations learned on various scenes. Rendered color images from the same view but with edited scene representation are shown in right column. In the first two rows, new objects are introduced to the scene. The third and fourth row shows *cloning* results wherein all *thing* MLP weights are duplicated to that of a single object in the scene. The last row shows manipulation of 3D pose of the objects.

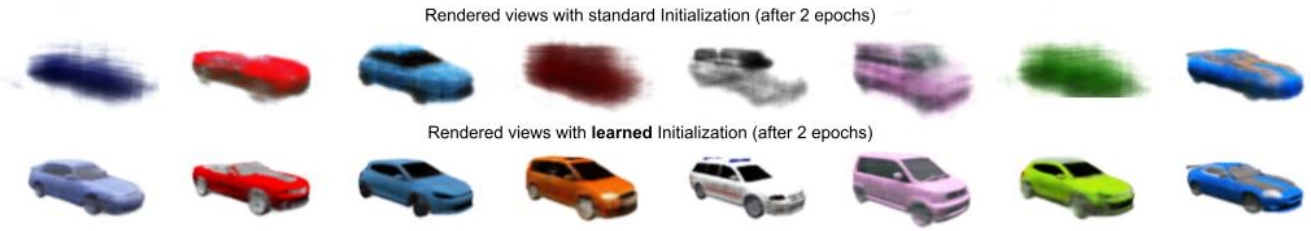

Figure 14. Comparison of rendered views with different initialization methods on ShapeNet dataset. **Top row:** Standard xavier (glorot) [19] initialization. **Bottom row:** Learned initialization via federated averaging. In both cases models were **only** optimized for two full epochs after the respective initialization. All models were trained on at-least 50 views. See Fig. 15 for analysis with sparse views.

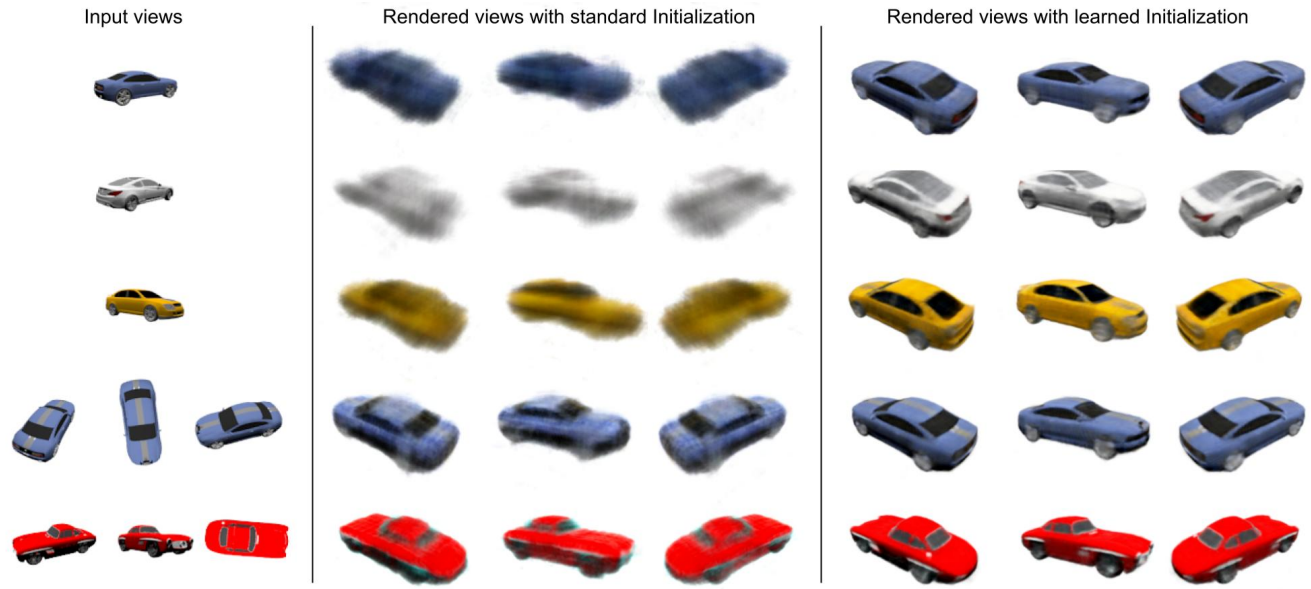

Figure 15. Comparison of the effect of initialization methods on training *thing* models on ShapeNet dataset with **very sparse** input views. We demonstrate using 1 or 3 image views as input to the model. The input views are shown left. Rendered views of the model when using the standard xavier (glorot) [19] initialization are shown in the middle columns. Rendered views with our proposed learned initialization are shown in the right columns. Even with just one image input, our model can reconstruct the cars pretty well.

used the forward facing cameras images for these results. Note that even though KITTI-360 [33] dataset provides side facing fisheye camera images, only the forward facing stereo images are available for the novel view test sequences used for the experiments. Each row of Fig. 12 and Fig. 11 shows a novel view of a scene generated by rendering the panoptic neural field representation of that scene. The left columns shows the rendered semantic segmentation overlaid on top of the rendered color image so that it is easier to judge the segmentation quality. The colormap used for visualizing the segmentation and depth images are shown at the bottom. The corresponding rendered depth images from those views are shown in the right columns. Note that even the difficult thin structures like lamp poles and sign posts are both accurately reconstructed and segmented by our model. Also note the accurate reconstruction and segmentation of multiple moving cars in Fig. 12. Additionally, rendered color images from some novel viewpoints are also available the left column of Fig. 13.

B.2. Scene Editing

Since our proposed panoptic neural field scene representation is object aware, it allows seamless manipulation and editing of different objects present in the scene. In addition to the scene editing results on Virtual KITTI [3] dataset discussed in Sec. 4 and Fig. 10, we show scene editing results on KITTI and KITTI-360 datasets in Fig. 13. Each row in Fig. 13 shows rendered color images along some novel view of both the original (left) and edited (right) scene representations. More specifically we show adding and removing new objects into the scene, changing the 3D pose of objects, and object *cloning* where the *thing* MLP parameters are replicated for all objects in the scene.

B.3. Benefits of our learned initialization

In real world scenes, objects are often captured from a sparse set of views. For example in self driving car scenes like KITTI, most objects (*e.g.* cars) often get a limited set of views from one side only. Thus incorporating prior knowledge becomes important for completeness and accurate reconstruction.

We learn the category specific priors from a large collection of objects on ShapeNet, as part of a separate meta-learning process and distill that knowledge as initialization when training on a novel scene. Thus our inference time scene representation network is more efficient and only focus on the individual set of object instances present in the scene. As demonstrated in Tab. 3, our model does a better job in reconstructing images of a dynamic scene compared to other object aware approaches like NSG [44], even though we use a much smaller MLP (10x fewer FLOPs) per object.

The learned initialization also provides other benefits like faster convergence and better completeness when reconstructed from sparse partial observations. We demonstrate these two benefits on ShapeNet [5] dataset in Fig. 14 and Fig. 15. Specifically, we used rendered images of *cars* from ShapeNet [5] provided by [56].

Fig. 14 qualitatively compares the rendered color images when using learned initialization over the standard xavier (glorot) [19] initialization after two full epochs of training. For this experiment each model has at-least 50 input views. As seen in Fig. 14, the proposed initialization offers clear benefits in terms of faster convergence even when there is a dense set of input views.

The advantage of the learned initialization over standard initialization is more pronounced when we have few sparse input views of an object. This is demonstrated in Fig. 15. Using the proposed learned initialization, our model can reconstruct novel object instances even with just a single image as input. As shown in Fig. 15, even when only a partial view of the objects are used as input, the category specific object priors distilled via the initialization results in a more complete reconstruction.

B.4. Video Results

We also encourage the readers to also look at supplemental videos demonstrating results of our framework and a overview of the method. Most of our results on dynamic scenes are better visualized in the video.

C. Potential Negative Societal Impact

Our contribution is an intermediate representation for comprehensive 3D scene understanding. We believe this can enable applications with a beneficial impact on society. However, it could also enable applications with potential negative impact. While it is impossible to anticipate all possible such applications, we discuss a few below.

Because our method supports comprehensive tracking of objects and people, it could be extended for use in crowd monitoring, traffic density reports and beneficial applications stemming from that. However, it could also be incorporated into surveillance systems. We will include stipulations in the license agreement for the code limiting its applications to academic research.

In addition, because our methods support view synthesis of 3D scenes, it is conceivable that it could be used to create imagery of fictional events, with the potential to disseminate fake news and/or propaganda. Because our method supports scene editing, actual events could be altered and used in similar ways. Of course, we will clearly mark all images generated by our system as "synthetic." Additionally, we will include a requirement to do the same in the download instructions for our code.

Mitigation of the above issues is hard: many computer vision contributions are intermediate representations like ours. Segmentation, feature tracking, and object recognition can be put together into diverse functioning applications. As a profession, we should strive for the ethical application of these new technologies.

References

[1] Jonathan Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul Srinivasan. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *ICCV*, 2021. 7

[2] Maros Blaha, Christoph Vogel, Audrey Richard, Jan D Wegner, Thomas Pock, and Konrad Schindler. Large-Scale Semantic 3D Reconstruction: An Adaptive Multi-Resolution Model for Multi-Class Volumetric Labeling. In *CVPR*, 2016. 2

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020. 6, 13

[4] Eric Chan, Marco Monteiro, Peter Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *CVPR*, 2021. 3

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015. 6, 9, 14

[6] 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 *CVPR*, 2020. 1, 2, 5, 7

[7] Jonathan Crespo, Jose Carlos Castillo, Oscar Martinez Mozos, and Ramon Barber. Semantic information for robot navigation: A survey. *Applied Sciences*, 10(2):497, 2020. 1

[8] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3D Scene Reconstruction From a Single RGB Image. In *NeurIPS*, 2021. 2

[9] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. On the effectiveness of weight-encoded neural implicit 3d shapes. *arXiv preprint arXiv:2009.09808*, 2020. 3

[10] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua Tenenbaum, and Jiajun Wu. Neural Radiance Flow for 4D View Synthesis and Video Processing. In *ICCV*, 2021. 3

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 3, 6

[12] 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. *arXiv preprint arXiv:2203.15224*, 2022. 3

[13] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. In *CVPR*, 2021. 3

[14] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 6, 8

[15] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait Neural Radiance Fields from a Single Image. *arXiv preprint arXiv:2012.05903*, 2020. 3

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 2, 6, 7, 9, 11

[17] Justin Johnson Georgia Gkioxari, Jitendra Malik. Mesh R-CNN. In *ICCV*, 2019. 2, 3

[18] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 1

[19] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 13, 14

[20] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-Centric Neural Scene Rendering. *arXiv preprint arXiv:2012.08503*, 2020. 2, 3

[21] Christian Hane, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3D scene Reconstruction and Class Segmentation. In *CVPR*, 2013. 2

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1

[23] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020. 1

[24] Wongbong Jang and Lourdes Agapito. CodeNeRF: Disentangled Neural Radiance Fields for Object Categories. In *ICCV*, 2021. 3

[25] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019. 3, 5, 6, 9

[26] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 1, 7

[27] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)*, 40(4):29–43, 2021. 7

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[29] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M Rehg. Joint Semantic Segmentation and 3D Reconstruction from Monocular Video. In *ECCV*, 2014. 2

[30] Abhijit Kundu, Yin Li, and James M Rehg. 3D-RCNN: Instance-level 3D Object Reconstruction via Render-and-Compare. In *CVPR*, 2018. 2

[31] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An Analysis of SVD for Deep Rotation Estimation. In *NeurIPS*, 2020. 9

[32] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *CVPR*, 2021. 3

[33] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv preprint arXiv:2109.13410*, 2021. 2, 6, 7, 9, 10, 13

[34] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-Adjusting Neural Radiance Fields. In *CVPR*, 2021. 9

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[36] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 5, 6, 9

[37] Ben Mildenhall, Pratul Srinivasan, Matthew Tancik, Jonathan Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020. 2, 3, 4, 5, 7, 9

[38] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *arXiv preprint arXiv:2201.05989*, 2022. 8

[39] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 2

[40] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3D scene reconstruction from posed images. In *ECCV*, 2020. 2, 3

[41] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In *IROS*, 2019. 2, 3

[42] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 3, 6, 9

[43] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020. 2, 3

[44] Julian Ost, Fahim Mannan, Nils Thürey, Julian Knodt, and Felix Heide. Neural Scene Graphs for Dynamic Scenes. In *CVPR*, 2021. 2, 3, 7, 8, 14

[45] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is Pseudo-Lidar needed for monocular 3D object detection? In *ICCV*, 2021. 2, 4

[46] Keunhong Park, Utkarsh Sinha, Jonathan Barron, Sofien Bouaziz, Dan Goldman, Steven Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. In *ICCV*, 2021. 3, 9

[47] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021. 3

[48] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2021. 3

[49] Amit Raj, Julian Tanke, James Hays, Minh Vo, Carsten Stoll, and Christoph Lassner. ANR: Articulated neural rendering for virtual avatars. In *CVPR*, 2021. 3

[50] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 7

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[52] A. Rosinol, M. Abate, Y. Chang, and L. Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *ICRA*, 2020. 3

[53] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone. 3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans. In *RSS*, 2020. 3

[54] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *CVPR*, 2013. 2, 3

[55] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *NeurIPS*, 2020. 3

[56] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *NeurIPS*, 2019. 6, 7, 14

[57] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-NeRF: Surface-free Human 3D Pose Refinement via Neural Rendering. *arXiv preprint arXiv:2102.06199*, 2021. 3

[58] Matthew Tancik, Ben Mildenhall, Terrence Wang, Divi Schmidt, Pratul Srinivasan, Jonathan Barron, and Ren Ng. Learned Initializations for Optimizing Coordinate-Based Neural Representations. In *CVPR*, 2021. 3, 6

[59] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in Neural Rendering. *arXiv preprint arXiv:2111.05849*, 2021. 2, 8

[60] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Vid2Actor: Free-viewpoint Animatable Person Synthesis from Video in the Wild. *arXiv preprint arXiv:2012.12884*, 2020. 3

[61] Hai Wu, Wenkai Han, Chenglu Wen, Xin Li, and Cheng Wang. 3d multi-object tracking in point clouds based on prediction confidence-guided data association. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–10, 2021. 4

[62] Linlin Xia, Jiashuo Cui, Ran Shen, Xun Xu, Yiping Gao, and Xinying Li. A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots. *International Journal of Advanced Robotic Systems*, 17(3):1729881420919185, 2020. 1

[63] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time Neural Irradiance Fields for Free-Viewpoint Video. In *CVPR*, 2021. 3

[64] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *arXiv preprint arXiv:2111.11426*, 2021. 2

[65] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning Object-Compositional Neural Radiance Field for Editable Scene Rendering. In *ICCV*, 2021. 2, 3

[66] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images. In *CVPR*, 2021. 3

[67] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 4

[68] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 7

[69] Hao Zheng, Zhanlei Yang, Wenju Liu, Jizhong Liang, and Yanpeng Li. Improving deep neural networks using softplus units. In *IJCNN*, 2015. 6

[70] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-Place Scene Labelling and Understanding with Implicit Scene Representation. In *ICCV*, 2021. 2, 3, 7