In-Place Scene Labelling and Understanding with Implicit Scene Representation

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, Andrew J. Davison
Dyson Robotics Laboratory at Imperial College
Department of Computing, Imperial College London, UK
{s.zhi17, t.laidlow15, s.leutenegger, a.davison}@imperial.ac.uk

Abstract

Semantic labelling is highly correlated with geometry and radiance reconstruction, as scene entities with similar shape and appearance are more likely to come from similar classes. Recent implicit neural reconstruction techniques are appealing as they do not require prior training data, but the same fully self-supervised approach is not possible for semantics because labels are human-defined properties.

We extend neural radiance fields (NeRF) to jointly encode semantics with appearance and geometry, so that complete and accurate 2D semantic labels can be achieved using a small amount of in-place annotations specific to the scene. The intrinsic multi-view consistency and smoothness of NeRF benefit semantics by enabling sparse labels to efficiently propagate. We show the benefit of this approach when labels are either sparse or very noisy in room-scale scenes. We demonstrate its advantageous properties in various interesting applications such as an efficient scene labelling tool, novel semantic view synthesis, label denoising, super-resolution, label interpolation and multi-view semantic label fusion in visual semantic mapping systems.

1. Introduction

Enabling intelligent agents, such as indoor mobile robots, to plan context-sensitive actions in their environment requires both a geometric and semantic understanding of the scene. Machine learning methods have proven to be valuable in both geometric and semantic prediction tasks, but the performance of these methods suffers when the distribution of the training data does not match the scenes observed at test-time. Though the issue can be mitigated by gathering costly annotated data or semi-supervised learning, it is not always feasible in open-set scenarios with various known and unknown classes. For this reason, it is advantageous to have methods that can self-supervise. In particular, there has been recent success in using scene-specific methods (e.g. NeRF [16]) that represent the shape and radiance of a single scene with a neural network trained from scratch using only images and associated camera poses.

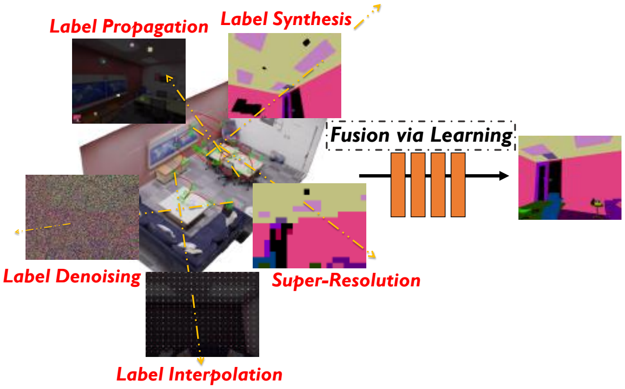


Figure 1: Neural radiance fields (NeRF) jointly encoding appearance and geometry contain strong priors for segmentation and clustering. We build upon this to create a scene-specific 3D semantic representation, Semantic-NeRF, and show that it can be efficiently learned with in-place supervision to perform various potential applications.

Semantic scene understanding means attaching class labels to a geometric model. The tasks of estimating the geometry of a scene and predicting its semantic labels are strongly related, as parts of a scene that have similar shape are more likely to belong to the same semantic category than those which differ greatly. This has been shown in work on multi-task learning [9, 33], where networks that simultaneously predict both shape and semantics perform better than when the tasks are tackled separately.

Unlike scene geometry, however, semantic classes are a human-defined concept and it is not possible to semantically label a novel scene in a purely self-supervised manner. The best that could be achieved would be to cluster self-similar structures of a scene into categories; but some labelling would always be needed to associate these clusters with human-defined semantic classes.

In this paper, we show how to design a scene-specific network for joint geometric and semantic prediction and train it on images from a single scene with only weak semantic supervision (and no geometric supervision). Because our single network must generate both geometry and semantics, the correlation between these tasks means that semantics prediction can benefit from the smoothness, coherence and self-similarity learned by self-supervision for geometry. In addition, multi-view consistency is inherent to the training process and enables the network to produce accurate semantic labels of the scene, including for views that are substantially different from any in the input set.

Our system takes as input a set of RGB images with associated known camera poses. We also supply some partial or noisy semantic labels for the images, such as ground truth labels for a small fraction of the images, or noisy or coarse label maps for a higher number of images. We train our network to jointly produce implicit 3D representations of both the geometry and semantics for the whole scene.

We evaluate our system both quantitatively and qualitatively on scenes from the Replica dataset [28], and qualitatively on real-world scenes from the ScanNet dataset [3]. Generating dense semantic labels for a whole scene from partial or noisy input labels is important for practical applications, like when a robot encounters a new scene and either only a small amount of in-situ labelling is feasible, or only an imperfect single-view network is available.

2. Related Work

Most existing 3D semantic mapping and understanding systems work by attaching (fused) semantic labels to a 3D geometric representation created by a standard reconstruction method. For example, [6] uses point clouds, [13] and [22] use surfels, [18] uses voxels, and [12] uses signed distance fields. These classical 3D geometric representations are all limited in their ability to efficiently represent fine details in complex topologies. Volumetric representations, for example, have a convenient structure for parallel processing or use with convolutional neural networks, but suffer from cubic memory requirements and high computational cost that ultimately limit the resolution it can represent.

2.1. Code-Based Representations

To help overcome these limitations, many learning-based representations have been developed. Code-based representations, for example, use the latent code of an autoencoder as a compact representation of the scene. Generative Query Networks (GQN) [5] can represent simple 3D scenes using a latent scene representation vector. CodeSLAM [1] uses a compact and optimisable latent code to represent dense scene geometry in a view-based visual odometry system. The approach of CodeSLAM was extended by SceneCode [35] to include semantics. SceneCode was able to make inference-time refinements of network predictions by optimising for photometric and semantic label consistency between multiple frames. However, although trained with depth maps, SceneCode was still a view-based representation and lacked true awareness of the 3D geometry.

2.2. Implicit 3D Representations

There has been much promising recent work on using neural implicit scene representations. As these are continuous representations, they can easily handle complicated topologies and do not suffer from discretisation error, with the actual representative resolution depending on the capacity of the neural network used. The Scene Representation Network (SRN) [25] was one of the first methods to use a multi-layer perceptron (MLP) as the neural representation of a learned scene given a collection of images and associated poses. DeepSDF [19] and DIST [10] used deep decoders to learn implicit signed distance functions of various shape instances of the same class, and Occupancy Networks [15, 21] learned an implicit 3D occupancy function for shapes or large scale scenes given 3D supervision.

Kohli *et al*. [8] also proposed to learn a joint implicit representation of appearance and semantics for 3D shapes on top of an SRN using a linear segmentation renderer. After being trained in a two-step semi-supervised manner, the network can synthesise novel view semantic labels from either colour or semantic observations.

The methods mentioned above involve extensive pre-training on collections of data to learn priors about the shapes or scenes they are used to represent. Although promising generalisation capability has been shown across different instances or scenes, it is not always possible to get adequate data for various unseen environments. The alternative is a scene-specific representation which requires minimum in-place labelling effort.

NeRF [16] and other systems based on it [34, 11, 31, 27] use MLPs to overfit input from a single bounded scene and act as an implicit volumetric representation for realistic view-synthesis. In this paper, we treat NeRF as a powerful scene-specific 3D implicit representation, and extend it to include semantic representation which can be efficiently learned from sparse or noisy annotations (Figure 1).

3. Method

3.1. Preliminaries

Given multiple images of a static scene with known camera intrinsics and extrinsics, NeRF [16] uses MLPs to implicitly represent the continuous 3D scene density σ and colour $\mathbf{c} = (r, g, b)$ as a function of continuous 5D input vectors of spatial coordinates $\mathbf{x} = (x, y, z)$ and viewing directions $\mathbf{d} = (\theta, \phi)$. Specifically, $\sigma(\mathbf{x})$ is designed to be a function of only 3D position while the radiance $\mathbf{c}(\mathbf{x}, \mathbf{d})$ is a function of both 3D position and viewing direction.

To compute the colour of a single pixel, NeRF [16] approximates volume rendering by numerical quadrature with hierarchical stratified sampling. Within one hierarchy, if $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ is the ray emitted from the centre of projection of camera space through a given pixel, traversing

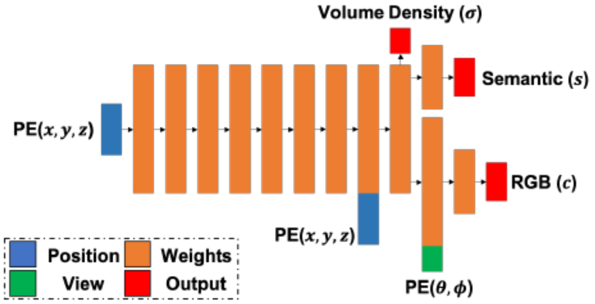


Figure 2: Semantic-NeRF network architecture. 3D position (x, y, z) and viewing direction (θ, ϕ) are fed into the network after positional encoding (PE). Volume density σ and semantic logits $\mathbf{s}$ are functions of 3D position while colours $\mathbf{c}$ additionally depend on viewing direction.

between near and far bounds (t_n and t_f), then for selected K random quadrature points $\{t_k\}_{k=1}^{K}$ between t_n and t_f, the approximated expected colour is given by:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\, \alpha\left(\sigma(t_k)\delta_k\right) \mathbf{c}(t_k)\,, \tag{1}$$

$$\text{where} \quad \hat{T}(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma(t_k)\delta_k\right), \tag{2}$$

where $\alpha\,(x) = 1 - \exp(-x)$, and $\delta_k = t_{k+1} - t_k$ is the distance between two adjacent quadrature sample points.

Given multi-view training images of the observed scene, NeRF uses stochastic gradient descent (SGD) to optimise σ and $\mathbf{c}$ by minimising photometric discrepancy.

3.2. Semantic-NeRF

We now show how to extend NeRF to jointly encode appearance, geometry and semantics. As shown in Figure 2, we augment the original NeRF by appending a segmentation renderer before injecting viewing directions into the MLP.

We formalise semantic segmentation as an inherently view-invariant function that maps only a world coordinate $\mathbf{x}$ to a distribution over C semantic labels via pre-softmax semantic logits $\mathbf{s}(\mathbf{x})$:

$$\mathbf{c} = F_\Theta(\mathbf{x}, \mathbf{d})\,, \quad \mathbf{s} = F_\Theta(\mathbf{x})\,, \tag{3}$$

where F_Θ represents the learned MLPs.

The approximated expected semantic logits $\hat{\mathbf{S}}(\mathbf{r})$ of a given pixel in the image plane can be written as:

$$\hat{\mathbf{S}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\, \alpha\left(\sigma(t_k)\delta_k\right) \mathbf{s}(t_k)\,, \tag{4}$$

$$\text{where} \quad \hat{T}(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma(t_k)\delta_k\right), \tag{5}$$

with $\alpha\,(x) = 1 - \exp(-x)$ and $\delta_k = t_{k+1} - t_k$ is the distance between adjacent sample points. Semantic logits can then be transformed into multi-class probabilities through a softmax normalisation layer.

3.3. Network Training

We train the whole network from scratch under photometric loss L_p and semantic loss L_s:

$$L_p = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left\| \hat{\mathbf{C}}_c(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right\|_2^2 + \left\| \hat{\mathbf{C}}_f(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right\|_2^2 \right], \tag{6}$$

$$L_s = -\sum_{\mathbf{r}\in\mathcal{R}} \left[\sum_{l=1}^{L} p^l(\mathbf{r}) \log \hat{p}_c^l(\mathbf{r}) + \sum_{l=1}^{L} p^l(\mathbf{r}) \log \hat{p}_f^l(\mathbf{r}) \right], \tag{7}$$

where $\mathcal{R}$ are the sampled rays within a training batch, and $\mathbf{C}(\mathbf{r})$, $\hat{\mathbf{C}}_c(\mathbf{r})$ and $\hat{\mathbf{C}}_f(\mathbf{r})$ are the ground truth, coarse volume predicted and fine volume predicted RGB colours for ray $\mathbf{r}$, respectively. Similarly, p^l, $\hat{p}_c^l$ and $\hat{p}_f^l$ are the multi-class semantic probability at class l of the ground truth map, coarse volume and fine volume predictions for ray $\mathbf{r}$, respectively. L_s is chosen as a multi-class cross-entropy loss to encourage the rendered semantic labels to be consistent with the provided labels, whether these are ground-truth, noisy or partial observations. Hence the total training loss L is:

$$L = L_p + \lambda L_s\,, \tag{8}$$

where λ is the weight of the semantic loss and is set to 0.04 to balance the magnitude of both losses [8]. In practice we find that actual performance is not sensitive to λ value and setting λ to 1 gives us similar performance. These photometric and semantic losses naturally encourage the network to generate multi-view consistent 2D renderings from the underlying joint representation.

3.4. Implementation

A scene-specific semantic representation is obtained by training the network from scratch for each scene individually. We use setup and hyper-parameters similar to [16]. Specifically, we use hierarchical volume sampling to jointly optimise coarse and fine networks, where the former provides importance sampling bias so that the latter can distribute more samples to positions likely to be visible. Positional encoding of length 10 and 4 [32, 30] are applied to 3D positions and viewing directions, respectively. In addition, since we have no depth information, we set the bounds of ray sampling to 0.1m and 10m respectively across experiments without careful tuning to span indoor scenes.

Training images are resized to 320x240 for all the experiments. We implement our model in PyTorch [20] and train it on a single RTX2080-Ti GPU with 11GB memory. The batch size of rays is set to 1024 due to memory limitations. We train the neural network using the Adam optimiser [7] with a learning rate of 5e-4 for 200,000 iterations.

4. Experiments and Applications

After training on colour images and semantic labels with associated poses, we obtain a scene-specific implicit 3D semantic representation. We evaluate its effectiveness quantitatively by projecting the 3D representation back into 2D image space where we have direct access to explicit ground truth data. We aim to show the benefits and promising applications of efficiently learning such a joint 3D representation for semantic labelling and understanding. We kindly urge readers to inspect more qualitative results in the appendix and video included in the supplementary material.

4.1. Indoor Scene Datasets and Data Preparation

Replica Replica [28] is a reconstruction-based 3D dataset of 18 high fidelity scenes with dense geometry, HDR textures and semantic annotations. We use the Habitat simulator [23] to render RGB colour images, depth maps and semantic labels from randomly generated 6-DOF trajectories similar to hand-held camera motions. We follow the procedure from SceneNet RGB-D [14], and lock the roll angle with the camera up-vector pointing along the y-axis.

We use the provided 88 semantic classes from Replica in scene-specific experiments and also manually map these labels to the popular NYUv2-13 definition [24, 4] in Section 4.4 for multi-view label fusion, following the mapping convention from ScanNet [3]. For each Replica scene of rooms and offices, we render 900 images at resolution 640x480 using the default pin-hole camera model with 90 degree horizontal field of view. We sample every 5th frame from the sequence to compose the training set and also sample intermediate frames to make the test set.

ScanNet ScanNet [3] is a large-scale real-world indoor RGB-D video dataset of 2.5M views in 1513 scenes with rich annotations including semantic segmentation, camera poses and surface reconstructions. We train Semantic-NeRF on ScanNet scenes using only the provided colour images, camera poses and 2D semantic labels. The sequences in each scene are evenly sampled so that the total amount of training data is roughly 300 frames. During experiments we select several indoor room-scale scenes and train one Semantic-NeRF per scene using posed images and semantic labels from the NYUv2-40 definition.

4.2. Semantic Neural Radiance Fields

We check the influence of semantics on appearance and geometry by quantitatively computing the quality of rendered RGB images and depth maps on Replica scenes with and without semantic prediction enabled. Experiments show that there is no clear difference which suggests that the current network has the capacity to learn these tasks jointly. Note that we might expect that significant high quality semantic labelling information could feasibly improve reconstruction quality, but in this paper we are focused on how geometry can help semantics in the opposite situation where semantic labelling is sparse or noisy.

4.3. Semantic View Synthesis with Sparse Labels

We first train our semantic-NeRF framework for novel view semantic label synthesis using all available RGB images with camera poses and corresponding semantic labels (i.e., 180 images) from a randomly generated sequence of a certain scene. This fully-supervised setup acts as an upper bound on the semantic segmentation performance of Semantic-NeRF given abundant labelled training data.

However, in practice it is expensive and time-consuming to acquire accurate dense semantic annotations for all observed images in a scene. Considering the redundancy in semantic labels among overlapping frames, we borrow the idea of key-framing from SLAM systems and hypothesise that providing labels for only selected frames should be enough to train the semantic representation efficiently. We choose key-frames by evenly sampling from sequences and train the networks from scratch with semantic labels only coming from those selected key-frames, while the synthesis performance is always evaluated on all test frames.

Figure 3 and 4 validate our assumption that semantics can be efficiently learned from sparse annotations with a sparsity ratio ranging from 0% to 95%, together with corresponding camera motion baselines as a complementary indication. Only marginal performance loss occurs when less than 10% semantic frames are used, and this is mainly caused by renderings of regions which are unobserved or occluded from key-frames. To take this even further, we manually select just two key-frames (99% sparsity ratio) from each scene to cover as much of the scene as possible. It turns out that our network, trained only with two labelled keyframes, can render accurate labels from various viewpoints.

4.4. Semantic Fusion

In addition to being able to learn the semantic representation with sparse annotations due to the redundancy present in the semantic labels, another important property of Semantic-NeRF is that multi-view consistency between semantic labels is enforced.

In semantic mapping systems (e.g. [29, 13, 18]), multiple 2D semantic observations are integrated into a 3D map or target frames to produce a more consistent and accurate semantic segmentation. Multi-view consistency is the key concept and motivation in semantic fusion, and the training process of Semantic-NeRF itself can be seen as a multi-view label fusion process. Given multiple noisy or partial semantic labels, the network can fuse them into a joint implicit 3D space so that we can extract a denoised label when

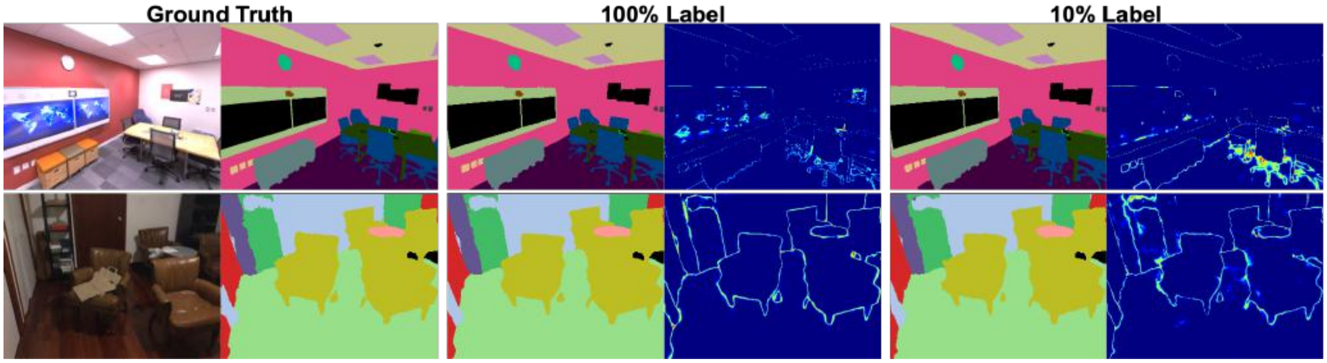


Figure 3: Synthesised semantic labels at testing poses given 100% and 10% of ground truth labels during training. From left to right we show the ground truth colour and semantic images for reference, and rendered semantic labels and their information entropy given 100% and 10% supervision, respectively. Bright parts of the entropy map match well to object boundaries or ambiguous/unknown regions in the corresponding training set-up.

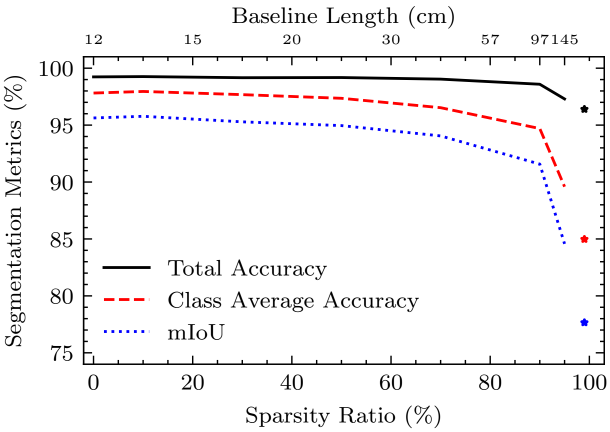


Figure 4: Quantitative performance of Semantic-NeRF trained on Replica with sparse semantic labels. Sparsity ratio is the percentage of frames dropped compared to full sequence supervision. Three standard metrics are used to evaluate semantic segmentation performance on test poses (higher is better). Performance gracefully degrades with fewer labels due to uncovered or occluded regions, indicating the possibility of efficient dense labelling from fewer annotations. Results with only two labelled key-frames ($\star$) show remarkably competitive performance.

we re-render the semantic labels from the learned representation back to input training frames.

We show the capability of Semantic-NeRF to perform multi-view semantic label fusion under a number of different scenarios: pixel-wise label noise, region-wise label noise, low-resolution dense or sparse labelling, partial labelling, and using the output of an imperfect CNN.

Semantic Label Denoising

Labels with Pixel-wise Noise We corrupt ground-truth training semantic labels by adding independent pixel-wise noise. Specifically, we randomly select a fixed portion of pixels per training frame and randomly flip their labels to arbitrary ones (including the void class). After training using only these noisy labels, we obtain denoised semantic labels by rendering back to the same training poses.

Figure 5 shows qualitative results from label denoising. When 90% of training pixels are randomly flipped, and it is difficult even for a human to recognise the underlying structure of the scene, the denoised labels still retain accurate boundaries and detail, especially for fine structures. Compared with Figure 3, the entropy in this denoising task is higher because the noisy training labels lack the multi-view consistency of clean ones. In addition, regions with void class tend to have the highest uncertainty since noisy pixels in void regions are not optimised during training. Quantitative results shown in Table 1 also confirm that accurate denoised labels are obtained after training-as-fusion.

While pixel-wise denoising with such severe corruption is not a realistic application, it is still a very challenging task and, more importantly, highlights our key observation that training itself is a fusion process which enables coherent renderings benefiting from the internal consistency of implicit joint representation.

Labels with Region-wise Noise We further validate the effectiveness of semantic consistency by randomly flipping the class labels of certain whole instances instead of pixels in the label maps. This is a better simulation of the behaviour of real single-view CNNs because a whole object can easily be labelled as a similar but incorrect class from an obstructed or ambiguous view.

We choose Replica Room_2 containing 8 instances of chairs as the testing scene. For each chair instance, we compute the occupied area ratio (i.e., ratio of the number of pixels belonging to that instance to the total number of pixels in the image for each ground truth label frame) and then sort the label maps in the sequence based on this occupied area ratio. Two criteria are used for selecting frames in which

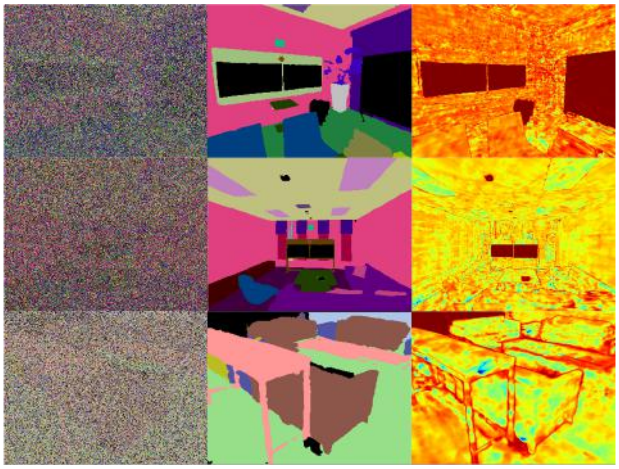


Figure 5: Qualitative results for semantic denoising. Even when 90% of all training labels are randomly corrupted, we can recover an accurate denoised semantic map. From left to right are noisy training labels, denoised labels rendered from the same poses after training, and information entropy. The overall high entropy we see in denoising tasks indicates the large inconsistency among noisy training labels.

Pixel-Wise Denoising		Metrics		
Noise Ratio		mIoU	Avg Acc	Total Acc
50%	Input Label	0.191	0.534	0.533
	Denoised Label	0.951	0.969	0.994
90%	Input Label	0.041	0.145	0.145
	Denoised Label	0.877	0.908	0.989

Region-Wise Denoising		mIoU		
Noise Ratio		30%	40%	50%
Sort	Input Label	0.866	0.842	0.793
	Denoised Label	0.895	0.893	0.803
Even	Input Label	0.741	0.692	0.684
	Denoised Label	0.796	0.747	0.733

Table 1: Quantitative evaluation for label denoising on Replica. Noise ratio is the percentage of changed pixels per frame, and for each instance the percentage of changed frames meeting selected criterion, respectively. mIoU is used for region denoising as it is more sensitive to the incorrect predictions on chair classes within the scene. Both tables are computed against clean training labels.

to randomly perturb each instance: (1) **Sort**: Select label maps with the least occupied area ratio. The intuition for this is that frames with partial observations are more likely to be mislabelled by semantic label prediction networks due to ambiguous context. (2) **Even**: Select label maps evenly from the sorted sequences introducing more large inconsistent regions into the training process.

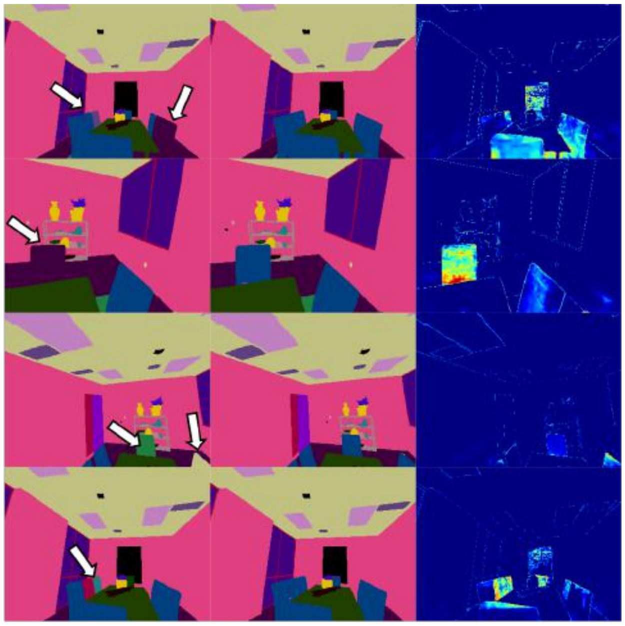


Figure 6: Qualitative results of rendered labels when we randomly change the training semantic class label (blue) of chair instances. From left to right: training label with region-wise noise; recovered semantic labels rendered from the same poses; and information entropy, highlighting regions with noisy predictions.

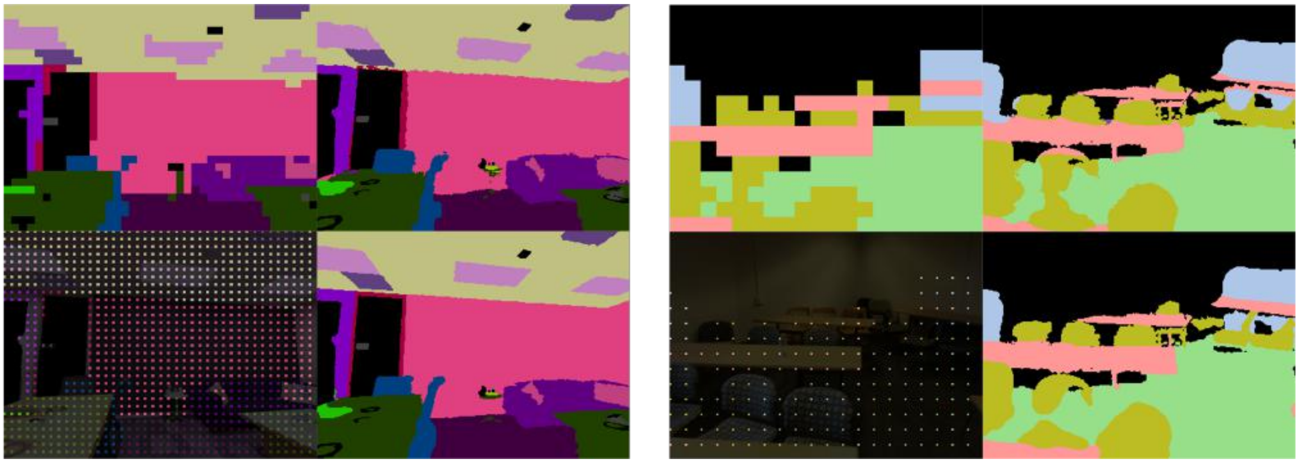
Figure 6 shows the qualitative results of the re-rendered semantic labels after training. We indeed observe that semantic labels of the chair instances can be corrected due to the enforcement of multi-view consistency during training. Table 1 also shows that there are steady improvements while it becomes much harder to render improved labels when a larger fraction of labels are perturbed.

Super-Resolution

Semantic label super-resolution is a useful application for scene labelling as well. In an incremental real-time semantic mapping system, a light-weight CNN predicting low-resolution semantic labels might be adopted to reduce computational cost (e.g. [17]). Another possible use case is in a scene labelling tool, since manual annotation in coarse images is much more efficient.

Here we show that we can train Semantic-NeRF with only low-resolution semantic information but then render accurate super-resolved semantics for either the input viewpoints or novel views. We test two different strategies to generate low-resolution training labels, with and without interpolation as shown in Figure 7. Given a down-scaling factor $S = 8$ for instance:

(1) All ground truth labels are down-scaled from 320×240 to 40×30 before being up-scaled back to the original



(a) Super Resolution Scale ×8

(b) Super Resolution Scale ×16

Figure 7: Super-resolution: we train Semantic-NeRF with only low resolution labels (sparsely sampled or interpolated) and obtain super-resolved labels by re-rendering semantics from the same poses. Note that the sparse labels have been zoomed-in 4 times and overlaid on top of colour images for the ease of visualisation.

Super-Resolution		Metrics		
Down-Scaling Factor		mIoU	Avg Acc	Total Acc
Dense	S=8	0.610	0.710	0.923
	S=16	0.433	0.535	0.855
Sparse	S=8	0.887	0.928	0.987
	S=16	0.800	0.866	0.977

Table 2: Quantitative evaluation of label super-resolution, with good performance with either sampled or interpolated low-resolution labels. The mIoU metric shows that sparse but geometrically accurate labels are more helpful for fine structures at high resolution.

size using nearest neighbour interpolation.

(2) All pixels except those from the low-res label map (row and column divisible by 8) are masked by the void class so as not to contribute to the training loss.

While method (1) uses interpolated labels to provide 'dense' supervision to a sampled ray batch but will incorrectly interpolate some pixels, method (2) provides sparse but geometrically accurate labels. We report super-resolution performance on training poses from all Replica scenes with two scales $S = 8$ and $S = 16$ in Table 2. Figures 1 and 7 show some examples where detailed semantic information is recovered through the fusion of many low-resolution or sparsely annotated semantic frames.

The promising results in semantic label denoising and super-resolution demonstrate one of the main advantages of jointly representing appearance, geometry and semantics: that missing or corrupted semantic labels in any one frame can be corrected through the fusion of many other frames. We explore this property in more detail in the next section.

Label Propagation

Our super-resolution experiments have shown the ability of Semantic-NeRF to interpolate rich details from low-resolution annotations. For a practical scene-annotation tool, straightforward annotations from a user in the form of clicks or scratches are desirable, and expected that those sparse clicks can expand and propagate to accurately and densely label the scene.

To simulate user annotations, for each class within label maps we randomly select a continuous sub-region with which to apply a ground-truth label while leaving the rest unlabelled. Results in Figure 8 and Table 3 show that supervision from one single pixel per class can lead to surprisingly high quality rendered labels with well preserved global and fine structure. Object boundaries are gradually refined when more supervision is available and the incremental improvements from more labels tend to saturate.

Multi-view Semantic Fusion

We have shown that a semantic representation can be learned from sparse or noisy/partial supervisions. Here we further validate its practical value in multi-view semantic fusion using CNN predictions.

There have been several classical pixel-wise semantic fusion approaches [6, 13, 12] to integrate monocular CNN predictions from multiple viewpoints to refine segmentation. For fair comparison, here we have separated out the multi-view fusion approaches from such systems. Two baseline techniques are: Bayesian fusion, where multi-class label probabilities are multiplied together and then renormalised (e.g. [13]), and average fusion, which simply takes the average of all label distributions (e.g. [12]).

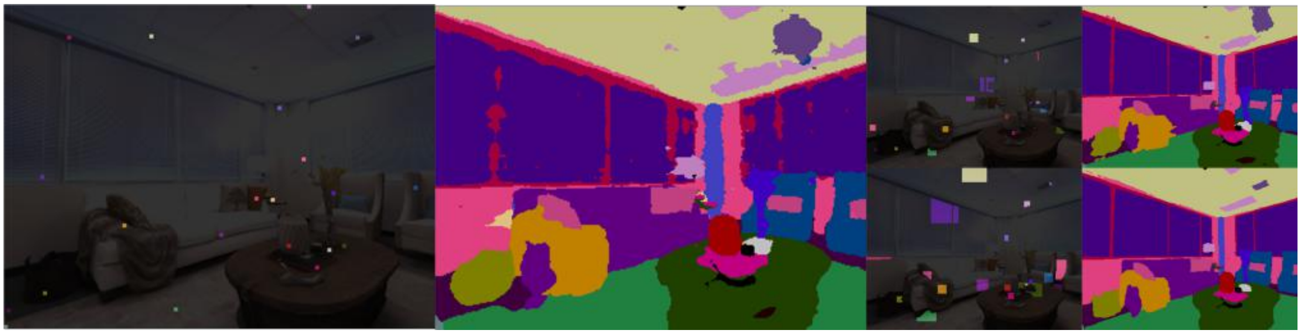


Figure 8: Label propagation results using partial annotations of a single-pixel, 1% or 5% of pixels per class within frames, respectively. Accurate labels can be achieved even from single-clicks, which are zoomed-in 9 times for visualisation purposes.

Label Propagation	Metrics		
# Labelling per Class	mIoU	Avg Acc	Total Acc
Single Click	0.602	0.937	0.908
1%	0.706	0.934	0.944
5%	0.836	0.946	0.971
10%	0.884	0.957	0.980

Table 3: Evaluation of label interpolation and propagation on Replica scenes using test poses. Even single-pixel supervision leads to competitive performance on the accuracy metrics, which highlights the effectiveness of the representation for interactive scene labelling.

To prepare training data in Replica dataset, we render two different sequences per Replica scene to cover various parts of scenes. Each sequence consists of 90 frames evenly sampled from 900 renderings of size 640×480 with semantic labels remapped to NYUv2-13 class convention.

We choose DeepLabV3+ [2] with a ResNet-101 backbone as the CNN model for monocular label predictions. To generate decent monocular CNN predictions and avoid over-fitting, we train DeepLab on SUN-RGBD [26], and then fine-tune it using data from all Replica scenes except the one chosen for training Semantic-NeRF and label fusion evaluation. We repeat this fine-tuning process and train one individual DeepLab CNN model for each test scene.

Monocular CNN predictions of the test scene are used for two purposes: (1) training supervision for our scene-specific Semantic-NeRF model; (2) monocular predictions (per-pixel dense softmax probabilities) for baseline multi-view semantic fusion methods. We train Semantic-NeRF using posed colour images together with CNN-predicted labels for 200,000 steps and then re-render the fused semantic labels back to the training poses as fusion results.

It is important to note that both baseline fusion techniques require depth information to compute the dense correspondences between frames while ours only requires posed images. We report the average performance across all testing scenes in Table 4, in which ground truth depth maps are used for the two baseline approaches to represent a ‘best case scenario’. Our method achieves the highest improvement across all metrics, showing the effectiveness of our joint representation in label fusion.

Semantic Fusion	mIoU	Avg Acc	Total Acc
Monocular	0.659	0.763	0.855
Bayesian Fusion *	0.668	0.764	0.865
Average Fusion *	0.586	0.703	0.814
Bayesian Fusion †	0.666	0.761	0.862
Average Fusion †	0.586	0.708	0.808
NeRF-Training (Ours)	**0.680**	**0.772**	**0.870**

* Using ground truth depth for data association.
† Using learned depth of Semantic-NeRF for data association.

Table 4: Comparison of multi-view semantic label fusion methods. Our approach relying on consistency of scene representations outperforms baselines aided with depth maps.

5. Conclusion and Future Work

We have shown that adding a semantic output to a scene-specific implicit MLP model of geometry and appearance means that complete and high resolution semantic labels can be generated for a scene when only partial, noisy or low-resolution semantic supervision is available. This method has practical uses in robotics or other applications where scene understanding is required in new scenes where only limited labelling is possible.

An interesting direction for future research is interactive labelling, where the continually training network asks for the new labels which will most resolve semantic ambiguity for the whole scene.

6. Acknowledgements

Research presented in this paper has been supported by Dyson Technology Ltd. Shuaifeng Zhi holds a China Scholarship Council-Imperial Scholarship. We are very grateful to Edgar Sucar, Shikun Liu, Kentaro Wada, Binbin Xu and Sotiris Papatheodorou for fruitful discussions.

References

[1] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. CodeSLAM — learning a compact, optimisable representation for dense visual SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 8

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4

[4] David Eigen and Rob Fergus. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015. 4

[5] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 2

[6] Alexander Hermans, Georgios Floros, and Bastian Leibe. Dense 3D semantic mapping of indoor scenes from RGB-D images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 2, 7

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 4

[8] Amit Kohli, Vincent Sitzmann, and Gordon Wetzstein. Inferring semantic information with 3d neural scene representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2020. 2, 3

[9] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[10] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[11] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[12] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, and S. Leutenegger. Fusion++:volumetric object-level slam. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2018. 2, 7

[13] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 2, 4, 7

[14] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. SceneNet RGB-D: Can 5M synthetic images beat generic ImageNet pre-training on indoor segmentation? In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017. 4

[15] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[16] 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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3

[17] Yoshikatsu Nakajima, Keisuke Tateno, Federico Tombari, and Hideo Saito. Fast and accurate semantic mapping through geometric-based incremental segmentation. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2018. 6

[18] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2019. 2, 4

[19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NIPS)*, 2019. 3

[21] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[22] Martin Runz, Maud Buffier, and Lourdes Agapito. Maskfusion: Real-time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2018. 2

[23] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 4

[24] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 4

[25] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-

structure-aware neural scene representations. In *Neural Information Processing Systems (NIPS)*, 2019. 2

[26] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 8

[27] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[28] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 4

[29] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid. Meaningful maps with object-oriented semantic mapping. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2017. 4

[30] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Neural Information Processing Systems (NIPS)*, 2020. 3

[31] Alex Trevithick and Bo Yang. GRF: Learning a general radiance field for 3d scene representation and rendering. *arXiv preprint arXiv:2010.04595*, 2020. 2

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems (NIPS)*, 2017. 3

[33] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[34] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[35] S. Zhi, M. Bloesch, S. Leutenegger, and A. J. Davison. SceneCode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

Supplementary Material for
In-Place Scene Labelling and Understanding with Implicit Scene Representation

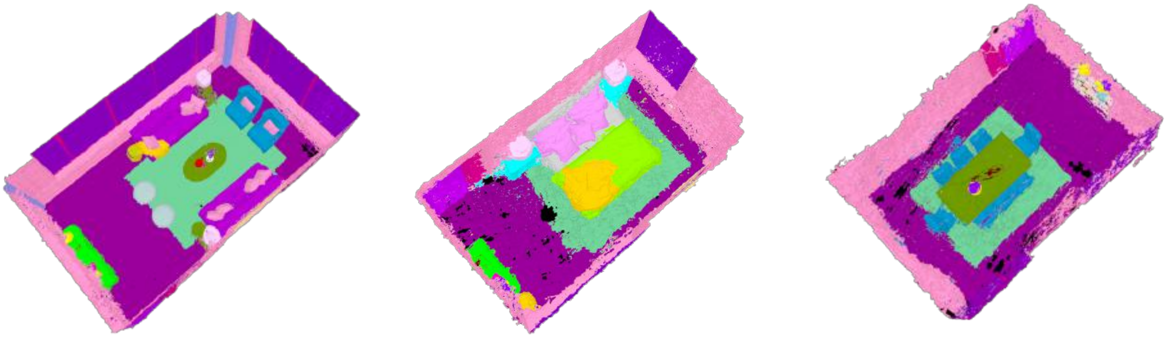


Figure 9: Semantic 3D reconstruction obtained using Semantic-NeRF. Note that our learned scene-specific 3D representation predicts decent geometry and semantics in occluded regions and fills the holes caused by unobserved regions to some extent.

2D Depth Metrics	
Abs Rel	$\frac{1}{n}\sum \lvert d - d_{gt}\rvert / d_{gt}$
Abs Diff	$\frac{1}{n}\sum \lvert d - d_{gt}\rvert$
Sq Rel	$\frac{1}{n}\sum \lvert d - d_{gt}\rvert^2 / d_{gt}$
RMSE	$\sqrt{\frac{1}{n}\sum \lvert d - d_{gt}\rvert^2}$
$\delta < 1.25^i$	$\frac{1}{n}\sum (\max(\frac{d}{d_{gt}}, \frac{d_{gt}}{d}) < 1.25^i)$

Table 5: Definitions of depth metrics: n is the number of valid depth pixels, d and d_{gt} are rendered depths at testing poses and ground truth depths, respectively.

A. Effects of Learning Semantics to Radiance and Geometry

Corresponding to Section 4.2 of the main paper, Table 6 shows quantitative results for photometric and geometric reconstruction quality when projected to 2D on Replica scenes with and without semantics enabled. We observe no obvious difference between these two set-ups. Peak signal-to-noise ratio (PSNR) is used to measure the quality of the rendered colour images and the metrics used to evaluate the 2D depth maps are shown in Table 5.

B. Semantic 3D Reconstruction from Posed Images

After training semantic-NeRF with in-place annotation, we can also extract an explicit 3D scene from the learned MLP to inspect the implicit 3D representation. Geometric meshes are extracted by first querying the MLP on dense 3D grids of the scene and then applying marching cubes. Attached semantic texture is rendered by treating the *negative* normal direction of vertices in the mesh as the ray marching directions during volume rendering. We show qualitative results for three Replica room scenes in Figure 9.

C. More Qualitative Results

Here we show more examples of qualitative results in Figure 10, 11, 12 for semantic view synthesis, label denoising and super-resolution, respectively.

We kindly urge readers to watch our *submitted video* which highlights the accuracy and consistency of semantic renderings in various situations and applications.

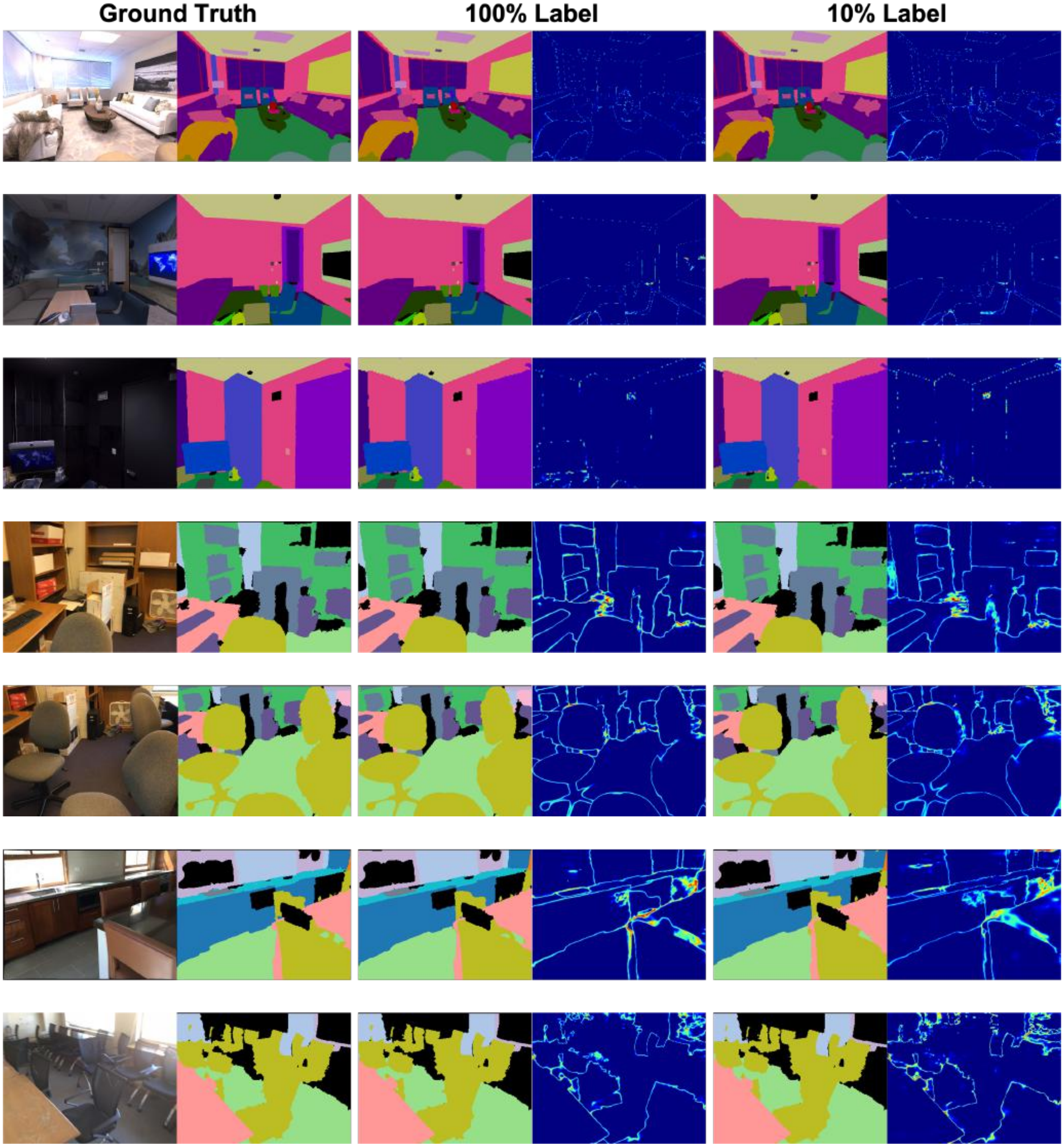


Figure 10: View-synthesis results.

Network Set-up	Depth							RGB
	AbsRel↓	AbsDiff↓	SqRel↓	RMSE↓	$\delta < 1.25$↑	$\delta < (1.25)^2$↑	$\delta < (1.25)^3$↑	PSNR↑
W/ Semantics	0.017	0.032	0.007	0.096	0.993	0.997	0.998	32.27
W/O Semantics	0.018	0.032	0.009	0.102	0.993	0.996	0.998	32.80

Table 6: Quantitative evaluation of effects of predicting semantics on appearance and geometry on Replica dataset.

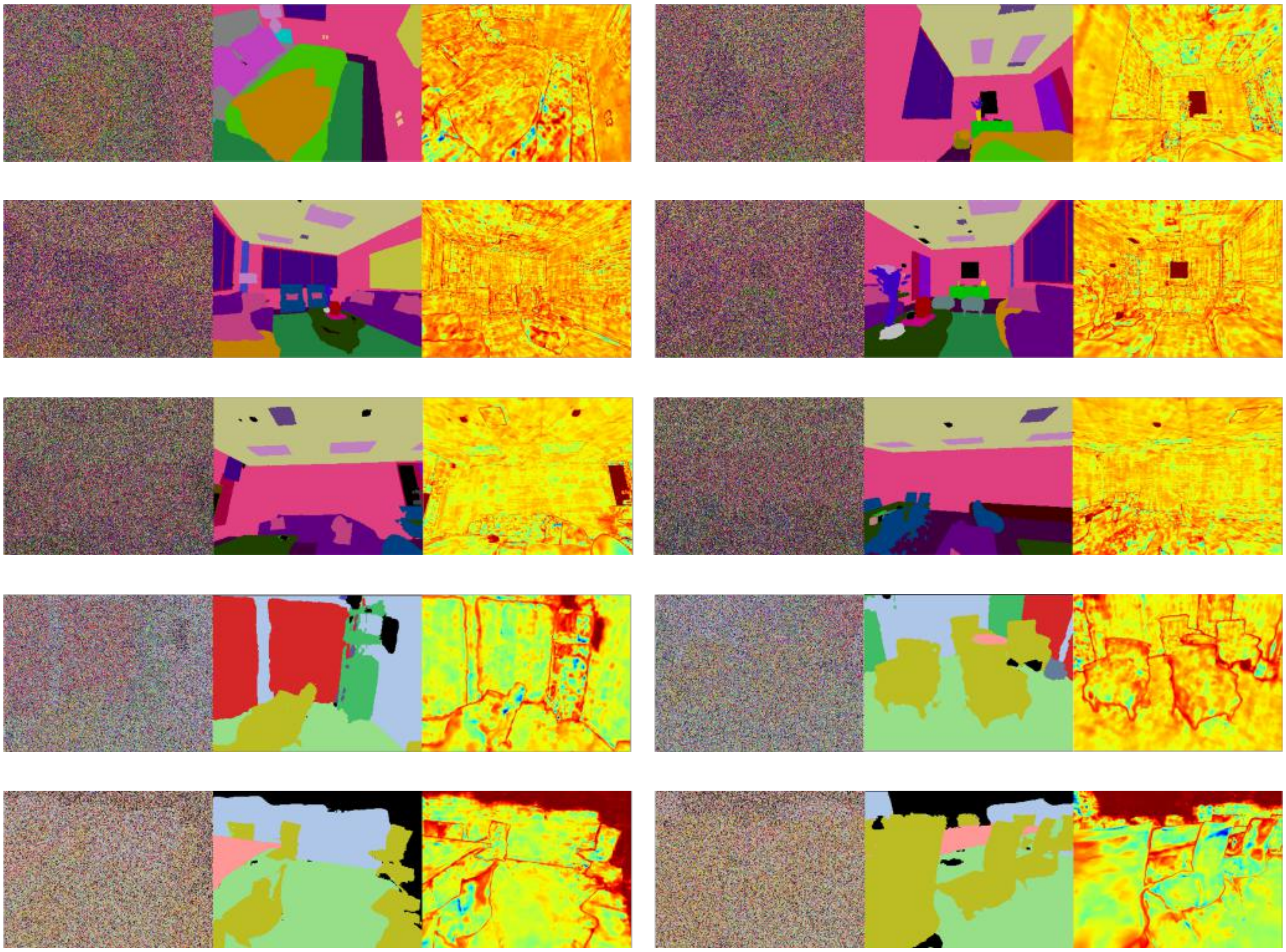
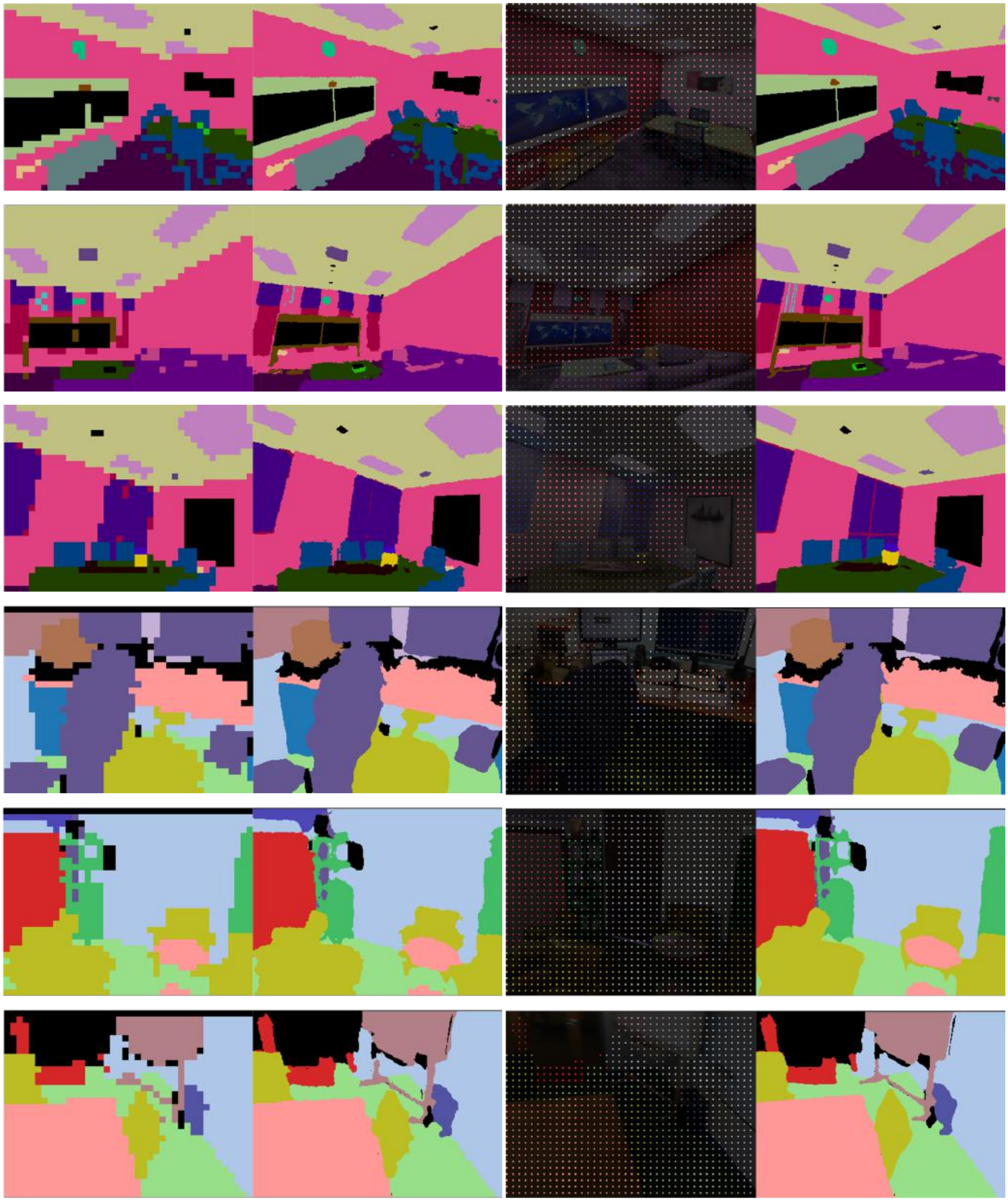


Figure 11: Pixel-wise denoising of semantic labels with 90% noise ratio.



(a) Super-resolution using coarse label

(b) Super-resolution using sparse label

Figure 12: Label super-resolution (×8) results.