Beyond RGB: Scene Analysis by Synthesis with Neural Radiance Fields

Anonymous CVPR submission

Paper ID 3979

Abstract

Comprehensive 3D scene understanding, both geometrically and semantically, is important for various real-world applications such as robot perception. Most of the existing work has focused on developing data-driven discriminative models for different scene analysis problems. This paper provides a new way for scene analysis from a generative modeling perspective, by leveraging the recent progress on implicit 3D representation and neural rendering. Building upon the great success of Neural Radiance Fields (NeRFs), we develop scene analysis by synthesis with NeRF (SaS-NeRF) that is able to not only render photo-realistic RGB images from novel viewpoints, but also render various accurate scene properties (e.g., appearance, geometry and semantics) paired with the synthesized images. By doing so, we facilitate addressing a variety of scene understanding tasks under a unified framework, including semantic segmentation, surface normal estimation, reshading, 2D-keypoint detection, and edge detection. Our SaS-NeRF framework can be a powerful tool for bridging generative learning and discriminative learning and thus be beneficial to the investigation of a wide range of interesting problems, such as studying task relationships from a generative modeling perspective, facilitating downstream discriminative tasks as ways of data augmentation, and serving as auto-labeller.

1. Introduction

Consider a domestic robot that is navigating in a room and performing various types of household tasks. To do so, the robot needs a comprehensive geometric and semantic understanding of the scene, uncovering the complete 3D spatial layout, functional attributes, semantic labels of the scene, its constituent objects, *etc*. [35]. Most of the existing work on 3D scene understanding has focused on developing data-driven *discriminative* models for various scene analysis problems [22, 25], such as semantic segmentation, object detection, and surface normal estimation. This paper provides a new way for scene analysis from a *generative modeling* perspective, by leveraging the recent progress on implicit 3D representation and neural rendering.

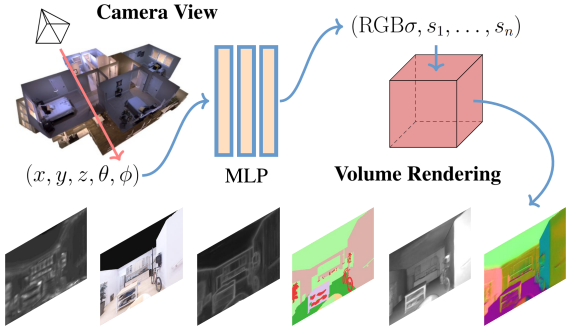

Figure 1. We represent the scene as an implicit function and develop a NeRF-style model, SaS-NeRF, that is able to not only render images from novel viewpoints, but also render various scene properties (*e.g.*, appearance, geometry and semantics) paired with the synthesized images, under a unified framework.

An important first step towards such *generative scene analysis* is being able to render photo-realistic scenes. One of the most influential advances in this direction is Neural Radiance Fields (NeRF) [33] which, given a handful of images of a static scene, learns an implicit volumetric representation of the scene that can be rendered from novel viewpoints. By sampling the coordinates along each camera ray from various views, NeRF represents a complex scene as a continuous 5D implicit function with a multilayer perceptron network, which regresses from a single 5D coordinate to a single volume density and view-dependent RGB color. In the end, NeRF accumulates those colors and densities into a 2D image through volume rendering. The implicit representation is optimized by minimizing the residual between synthesized images and ground truth of various views. NeRF has inspired significant follow-up work that has primarily focused on improving the quality of rendered images [38, 51], speeding up the training and rendering time [11, 16, 30, 45], handling unbounded scenes [70] and dynamic scenes [31], and incorporating uncertainty [52].

In this paper, we are interested in a different question:

Could this implicit representation be extended to synthesize richer scene properties beyond RGB color? The answer is **yes!** As illustrated in Figure 1, we develop a NeRF-style model that is able to not only render photo-realistic RGB images from novel viewpoints, but also render various accurate scene properties (*e.g.*, appearance, geometry and semantics) corresponding to the synthesized images, *under a unified framework*. This thus facilitates comprehensive scene understanding including semantic segmentation, surface normal estimation, reshading, 2D-keypoint detection, and edge detection. We call our framework as *scene analysis by synthesis* with NeRF (SaS-NeRF).

Naturally, these different scene properties can be grouped into two types: one is *view-dependent* that responses to the change of the observations (*e.g.*, surface normal and reshading), while the other is *view-independent* that is consistent across different view directions (*e.g.*, semantic labels, edges, and 2D-keypoints). SaS-NeRF is able to deal with these different types of properties in a coherent way (Figure 2), yielding realistic synthesis for all of them.

As a general, flexible framework, SaS-NeRF further facilitates the investigation of a variety of interesting problems. For example, within the SaS-NeRF framework, we analyze the relationship among different scene properties and show that a learned common implicit representation enables the flow of knowledge across different synthesis tasks, so that they can benefit one another. While similar phenomena have been widely investigated in multi-task discriminative learning such as Taskonomy [69], they are largely under-explored in the generative learning context. Moreover, we explore two applications of SaS-NeRF. We show that the SaS-NeRF synthesized examples (RGB images paired with scene properties) are useful augmented data for improving the corresponding downstream discriminative tasks. In addition, we show that, because of its learned underlying 3D geometry and scene representations, SaS-NeRF can work as an auto-labeller to refine the pseudo-labels produced by state-of-the-art discriminative models.

Our contributions are three-fold:

- We introduce a novel problem of "scene analysis by synthesis" that exploits generative modeling for a variety of scene understanding tasks.
- We propose an implicit representation based model SaS-NeRF that extends NeRF to simultaneously render novel-view images and their corresponding view-dependent and view-independent scene properties, such as semantic labels, surface normal, reshading, 2D-keypoints, and edges.
- We show that our resulting SaS-NeRF framework can be a powerful tool for bridging generative learning and discriminative learning and thus be beneficial to the investigation of a variety of problems, such as studying task relationships from a generative modeling perspective, facilitating downstream discriminative tasks as ways of data augmentation, and serving as auto-labeller.

2. Related Work

Generative-based View Synthesis aims to generate photo-realistic images of a scene from multiple viewpoints. Recent Generative Adversarial Networks [18] (GANs)-based models have shown promising results for view synthesis [3, 8, 17, 26, 37, 71]. Though some works also investigate explicitly modeling the geometry properties of scenes [6, 19] or introducing 3D shape representations as inductive bias [23, 62, 74], these models still cannot learn the implicit 3D representations of the scenes.

Implicit 3D Representations have gained popularity in learning-based 3D reconstruction [39, 42, 47, 53] and image synthesis [7, 44]. Recent works on hybrid continuous grid representations enable differentiable rendering process, so as to learn continuous shape and texture functions [54, 65].

Combining the implicit neural model and the volume rendering technology, Neural Radiance Field (NeRF) [33] achieves impressive novel view synthesis of complicated scenes. It learns an implicit 3D geometry representation of scenes with a standard multiple layer perception (MLP) and implements image synthesis by volume rendering. Some following works further improve the generalization capability [1, 20, 51, 66], compositionality [21, 38, 41, 70] and efficiency of inference [11, 16, 30, 45]. Inductive biases, such as depth and multi-view consistency, are also introduced to facilitate NeRF-style architectures [40, 59, 61]. However, these methods still focus on the RGB synthesis task, while our SaS-NeRF scales from RGB synthesis to various other pixel-level generative tasks with a shared 3D geometry and scene representation. Notice that Semantic-NeRF [72] also extends the NeRF-style architecture to semantic segmentation (which can be viewed as a special case of our framework), but our model works in a more general manner and can satisfy different task settings.

Scene Understanding is one of the high-level goals for machines vision, aiming to extract a similar level of knowledge and features and achieve human-like cognition. Recent works have gained impressive results of semantic segmentation [15, 22, 29, 43], object detection [5, 25, 73], visual reasoning [4, 9, 24, 63], *etc*. Though with great achievements, few of them focus on understanding scenes from a generative modeling perspective. In comparison, SaS-NeRF considers an implicit representation of 3D shape and scene properties, allowing for knowledge transfer and feature sharing across different tasks and thus capturing the underlying image generation mechanism for more comprehensive scene understanding than being done within indi-

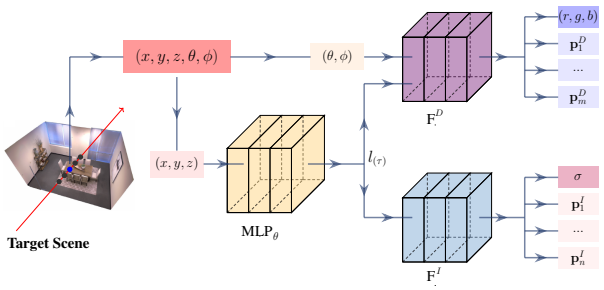

Figure 2. SaS-NeRF architecture. The model takes the 3D coordinates and view directions as the input and is able to synthesize different paired scene properties. SaS-NeRF uses a shared scene encoding network MLP_θ to conduct the 3D positional embedding, followed by two separate decoder networks F^D and F^I which produce view-dependent and view-independent predictions. F^D takes the additional view input and produces the (r, g, b) of a given point. F^I also produces the density of the given point.

vidual tasks.

Multi-task Learning aims to jointly solve different tasks through leveraging useful shared information from related tasks [10]. Recent works mainly use either soft parameter sharing [34, 64] or hard parameter sharing [13, 28] strategies [46] and have obtained great achievements. Beyond solving multi-task learning, the task relationships among different tasks have also been studied. *Taskonomy* and the following works [2,55,57,67,69] extensively exploit the task relationships to gain the best performance. Compared to these works, SaS-NeRF, as a general framework, can also be scaled to solve multiple visual tasks jointly, and task relationships can be further investigated in a generative manner.

3. Methodology

Figure 2 illustrates the architecture of our proposed SaS-NeRF framework (Scene Analysis by Synthesis with NeRF). In this section, we first introduce the basic conception of Neural Radiance Fields in Section 3.1. Then we propose a novel problem of scene analysis by synthesis and describe its setting in Section 3.2. Finally, in Section 3.3, we instantiate SaS-NeRF with five representative tasks which are widely investigated in scene understanding (semantic segmentation, surface normal estimation, reshading, 2D-keypoint detection, and edge detection) and discuss our model design in more detail.

3.1. Neural Radiance Fields

Given a 3D coordinate in $\mathbf{x} = (x, y, z)$ and a viewing direction $\mathbf{d} = (\theta, \phi)$, NeRF [33] learns an implicit scene representation f to map the 5D input to an RGB color $\mathbf{c} = (r, g, b)$ and volume density σ: $f(\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma)$.

Then NeRF calculates the single pixel color value by hierarchically tracking and sampling the single camera rays $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, which is emitted from the center of the projection of the camera space through that pixel. Specifically, it randomly samples K quadrature points $\{t_k\}_{k=1}^K$ with color $c(t_k)$ and density $\sigma(t_k)$ between the near boundary t_n and far boundary t_f. Then the approximated color of that pixel is given by:

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

where

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

denotes the accumulated transmittance. $\alpha(d) = 1 - \exp(-d)$ and δ_k is the distance between the two continuous samples ($\delta_k = t_{k+1} - t_k$).

3.2. Problem Setting

Inspired by the fact that NeRF learns the implicit 3D geometry and the scene representation, which can also be useful across different tasks, we generalize the basic NeRF problem setting from single RGB synthesis to rendering additional *pixel-wise scene properties* (*e.g.*, semantic labels, edges, surface normal, *etc.*). Specifically, for a new property $\mathbf{p}_i$, we want to estimate it for each 3D location and view direction: $f(\mathbf{x}, \mathbf{d}) \mapsto \mathbf{p}_i$. Notice that one property may or may not respond to the change of the observations, *e.g.* color varies from different viewpoints but the density is consistent among different views; we further divide the properties into two categories: *view-dependent* properties and *view-independent* properties. For the view-dependent property $\mathbf{p}_i^D$, we estimate with $f^D(\mathbf{x}, \mathbf{d}) \mapsto \mathbf{p}_i^D$; for the view-independent property $\mathbf{p}_j^I$, we estimate with $f^I(\mathbf{x}) \mapsto \mathbf{p}_j^I$.

Since the implicit function encodes the geometry, shape, and texture information of the scene, which are shareable across different property prediction tasks, we argue that different properties can be learned together with shared knowledge. Thus, here we propose a novel "scene analysis by synthesis" problem which is described as follows. For a bunch of *M+N* properties $P = [\{\mathbf{p}_m^D\}_{m=1}^M, \{\mathbf{p}_n^I\}_{n=1}^N]$, we want to learn a function that can map the 3D coordinates and the view directions to the corresponding properties $f(\mathbf{x}, \mathbf{d}) \mapsto P$.

3.3. SaS-NeRF

Model Architecture. To solve this novel problem, we propose SaS-NeRF, whose model architecture is shown in Figure 2. The whole model takes the 5D vector, coordinates and view directions, as the input and aims to predict

the corresponding properties. We use a shared positional encoding network MLP_θ to embed the 3D coordinates into higher dimensions. Then, we adopt two decoding networks: F^D takes the additional view input $\mathbf{d} = (\theta, \phi)$ for view-dependent tasks, and F^I for view-independent tasks. Notice that here we treat the RGB image and density predictions as the basic tasks which will always be trained together with other tasks. The reason is that the density is the fundamental property of the scene and the color information is easiest to obtain and is also the most informative property.

Instantiation of SaS-NeRF. Our SaS-NeRF is a general framework and is applicable to a variety of scene property synthesis tasks. Here we instantiate SaS-NeRF with five representative scene properties that are important in practice [55, 69], together with the color image synthesis. These properties are: **Semantic Segmentation (SS), Surface Normal (SN), Reshading (RE), 2D-Keypoints (KP), and Edge Detection (ED)**.

Specifically, for these tasks, we first feed the 3D coordinate (x, y, z) to the positional encoding network MLP_θ and obtain a latent representation $l_{(\tau)}$:

$$l_{(\tau)} = MLP_\theta(x, y, z). \tag{3}$$

Then, for RGB and the *view-dependent tasks*, labels are predicted using the information of both $l_{(\tau)}$ and the 2D view (θ, ϕ):

$$\{\mathbf{p}_m^D\}_1^M = F^D(l_{(\tau)}, \theta, \phi). \tag{4}$$

For density σ and the *view-independent tasks*, since they have no relationships with the camera direction (θ, ϕ), they are directly predicted from the latent representation $l_{(\tau)}$:

$$\{\mathbf{p}_n^I\}_1^N = F^I(l_{(\tau)}). \tag{5}$$

We train a specific task along with image synthesis. Our loss is simply the sum of photo-metric loss and the standard loss of the specific task as follows:

$$\mathcal{L}_{\text{whole}} = \mathcal{L}_{\text{rgb}} + \lambda_{T_i}\mathcal{L}_{T_i}, \tag{6}$$

where $\mathcal{T} = \{T_1, T_2, \cdots, T_i\}$ is the set of multiple tasks, λ_{T_i} is the weights of different tasks. For example, the loss of semantic segmentation is obtained as:

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

where $p^l, \hat{p}_c^l, \hat{p}_f^l$ are the ground truth, coarse volume prediction, and fine volume prediction of multi-class semantic probability of class l, respectively. $\mathcal{R}$ is the set of rays $\mathbf{r}$ in each batch for the segmentation task. Coarse and fine predictions $\hat{p}_c^l, \hat{p}_f^l$ are processed by a softmax layer after volume rendering. For RGB prediction and surface normal estimation, we adopt the $\mathcal{L}2$ loss:

$$\mathcal{L}_{\text{MSE}} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left\| \hat{T}_c(\mathbf{r}) - T(\mathbf{r}) \right\|_2^2 + \left\| \hat{T}_f(\mathbf{r}) - T(\mathbf{r}) \right\|_2^2 \right], \tag{8}$$

where $T(\mathbf{r}), \hat{T}_c(\mathbf{r}), \hat{T}_f(\mathbf{r})$ are the ground truth, coarse volume prediction, and fine volume prediction for task T, respectively. $\mathcal{R}$ is the set of rays $\mathbf{r}$ in each batch for all tasks above. While for the tasks of reshading, keypoint detection and edge detection, we adopt the $\mathcal{L}1$ loss:

$$\mathcal{L}_{\text{ABSE}} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left\| \hat{T}_c(\mathbf{r}) - T(\mathbf{r}) \right\|_1 + \left\| \hat{T}_f(\mathbf{r}) - T(\mathbf{r}) \right\|_1 \right]. \tag{9}$$

Scene	SS(↑)	SN(↓)	RE(↓)	KP(↓)	ED(↓)
FRL_apt_2	0.8926	0.0280	0.0362	0.0039	0.0287
FRL_apt_4	0.8552	0.0423	0.0404	0.0041	0.0341
Office_0	0.8969	0.0369	0.0629	0.0043	0.0430
Office_3	0.8426	0.0254	0.0341	0.0039	0.0307
Avg.	0.8718	0.0332	0.0434	0.0041	0.0341

Table 1. SaS-NeRF's performance on individual tasks. SS: Semantic Segmentation; SN: Surface Normal; RE: Reshading; KP: 2D Keypoints; ED: Edge Detection. Following the standard practice, for the segmentation task, we use mIoU as the evaluation metric; for the rest of the tasks, we adopt $\mathcal{L}1$ error as the evaluation metric. Our model reaches high quantitative scores for all the tasks, indicating that we can render accurate scene properties with a similar distribution of the ground-truth.

4. Experimental Evaluation

4.1. Experimental Setting

Datasets. We conduct extensive experiments on the commonly-used Replica [56] dataset. The Replica dataset is a high-quality synthetic scene dataset containing photo-realistic 3D modelings for 18 scenes in total. While there are other indoor-scene datasets with multiple scene property annotations, they are not satisfied for generative models, which require densely sampled multi-view images. The Replica dataset provides an engine to render the images from any views, which can give us enough data to train a NeRF-based model. Following the previous work [72], we conduct experiments on four scenes and for each scene, we rendered 50 images. Specifically, we render RGB images with corresponding semantic segmentation annotations at resolution 640×480 pixels for Replica. The rendering process is based on the Habitat platform [48, 58]. We normalize the maximum scale of the camera parameters to 10m and set near and far sample bounds to 0.1m and 10m, respectively.

We also investigate the application of SaS-NeRF on an LLFF-based [32] dataset, which we call LLFF dataset

in the following parts. LLFF is a method to produce photo-realistic novel views for well-sampled forward-facing scenes. LLFF is also one of the common datasets used for view synthesis. The original image size of LLFF is 4032×3024 pixels. We follow the same experimental setting as NeRF [33].

Tasks. Following the observation in [55], we adopt 6 representative visual tasks including the color image synthesis in our experiments: Semantic Segmentation (SS), Surface Normal Estimation (SN), Reshading (RE), 2D-Keypoint Detection (KP), Edge Detection (ED), and also color image synthesis. Given that, except for semantic segmentation, there is no human annotated ground-truth available for other tasks on the datasets, we therefore follow the standard procedure of [69] and use a pre-trained oracle annotator [68] to generate the pseudo-annotations. We also map the original 88-way semantic classes to commonly-used NYUv2-13 [14, 36] definition.

Implementation Details. We adopt COLMAP [49, 50], a well-performing pipeline for multi-view stereo (MVS), to reconstruct the poses and camera matrix for our model. We optimize our model for each scene separately. For the weights of the multiple tasks, we adopt $\lambda_{\text{SN}} = 1$ for surface normal estimation, $\lambda_{\text{SS}} = 0.04$ for semantic segmentation, $\lambda_{\text{RE}} = 0.1$ for reshading, $\lambda_{\text{KP}} = 2$ for keypoint detection, and $\lambda_{\text{ED}} = 0.4$ for edge detection, in order to balance the numerical values of all loss terms roughly at the same scale. We use the Adam Optimizer [27] with an initial learning rate of 5×10^{-4} and set hyperparameters at default values of $\beta_1 = 0.9, \beta_2 = 0.999$. We train our model for 200k iterations on one scene, and it takes about 9 hours on a single NVIDIA GeForce RTX 2080 Ti GPU.

Evaluation Metrics. We use mean Intersection-over-Union (mIoU) to evaluate the semantic segmentation performance and use absolute error to measure the performance of other tasks.

4.2. Performance on Tasks Beyond RGB

We first show our results with individual training for single tasks. We report the quantitative results in Table 1. SaS-NeRF reaches a high performance for all five tasks, indicating that our model can well capture the original data distribution of the ground-truth. We also include comparisons with some naive baselines in the supplementary material. Note that the main objective of this paper is to show that, with SaS-NeRF, it is able to synthesize different scene properties paired with the rendered images; therefore, there are no existing works as baselines for a more comprehensive comparison. We mainly focus on the comparisons between different variants of SaS-NeRF. While there has been a large body of works on training discriminative models to predict the scene properties for *real* images, it is difficult to make an apple-to-apple comparison between these discriminative models and our generative models. Conceptually, the generative models can in principle produce infinite paired annotations, while the discriminative models are constrained by the given data. Furthermore, in Section 4.4.1, we show that our SaS-NeRF is in fact *complementary and beneficial* to existing discriminative models.

We also visualize our rendered scene properties and compare with the corresponding ground-truth in Figure 3. All of the images manifest the good quality of our SaS-NeRF's novel view synthesis results for additional tasks beyond RGB.

4.3. Modeling Without RGB

In Section 4.2 for all scenarios, we always built our models with the color image synthesis. We argue that the color property is a fundamental property of the scene, and it can be beneficial to learning the underlying geometry and scene representations and so can facilitate the learning of the other tasks. To verify this idea, here we build a variant of SaS-NeRF without the RGB image (**SaS-NeRF**$_{w/o\ c}$) synthesis during the training process. If the color information is totally disentangled with other tasks, training without RGB will have almost the same results or even better results, since the model only needs to focus on the tasks which we are interested in.

The experimental results of models trained without RGB are shown in Figure 4. It is obvious to see that for the semantic segmentation task, the results of SaS-NeRF$_{w/o\ c}$ perform much worse than the main SaS-NeRF model. It is also worth noting that for keypoint detection, when RGB supervision is missing, the model inevitably collapses, ending up predicting an all-zero map. We hypothesize the underlying reason might be that the understanding of 3D shape for SaS-NeRF is based on the original image synthesis. Thus, RGB supervision is crucial for understanding the scenes and learning other visual tasks.

4.4. Further Explorations within SaS-NeRF

In addition to predicting labels of a given camera pose, SaS-NeRF can generate more data with new poses. This major benefit of solving visual tasks with generative models, compared with discriminative models, enables us to make richer explorations within our SaS-NeRF framework.

4.4.1 Data Augmentation for Multi-task Learning

Given that we can render photo-realistic images and their corresponding scene property annotations, one natural, interesting question is: how can we make use of these paired synthesized data? Inspired by [2] and [12], we design the following experiment. We adopt a task network (*i.e.*, a standard discriminative model) to evaluate each task, and then we train this model with two different data settings:

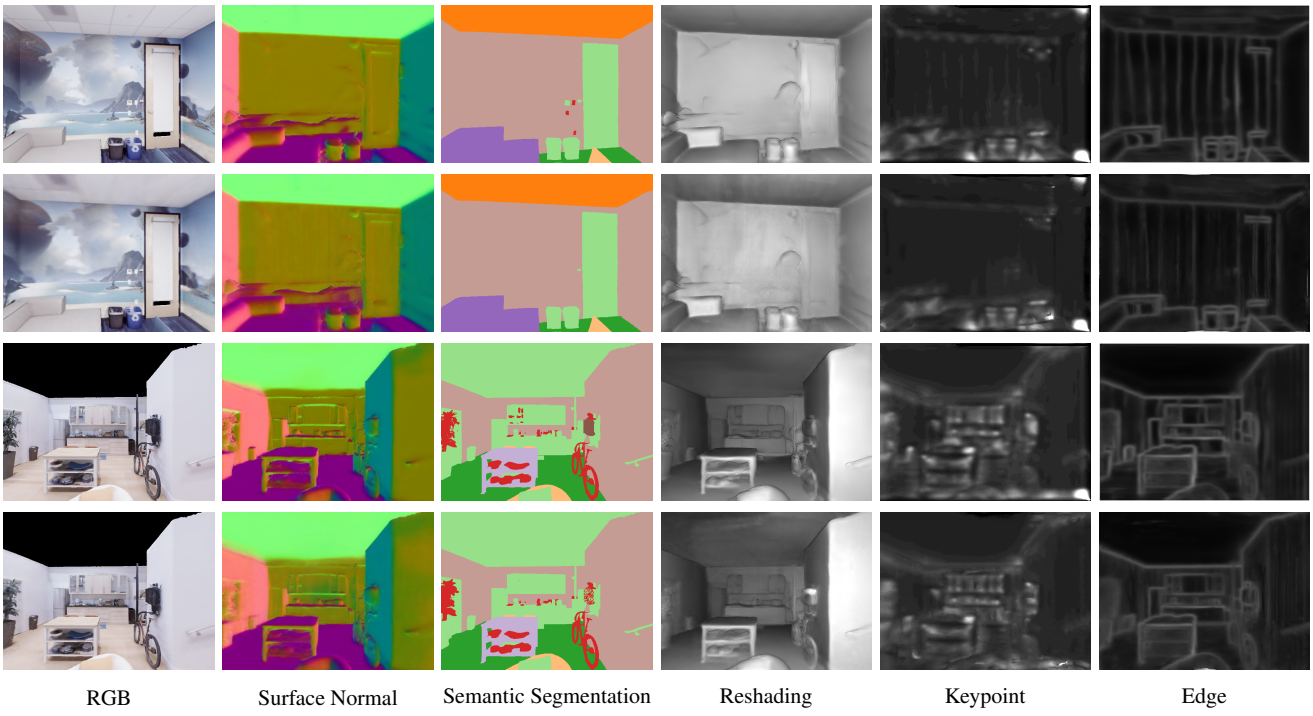

Figure 3. Two qualitative results of testing views. **Top row:** ground-truth; **Bottom row:** our predictions. The predicted image is from the semantic segmentation task. SaS-NeRF generates *realistic and matching images and other properties* on all the tasks.

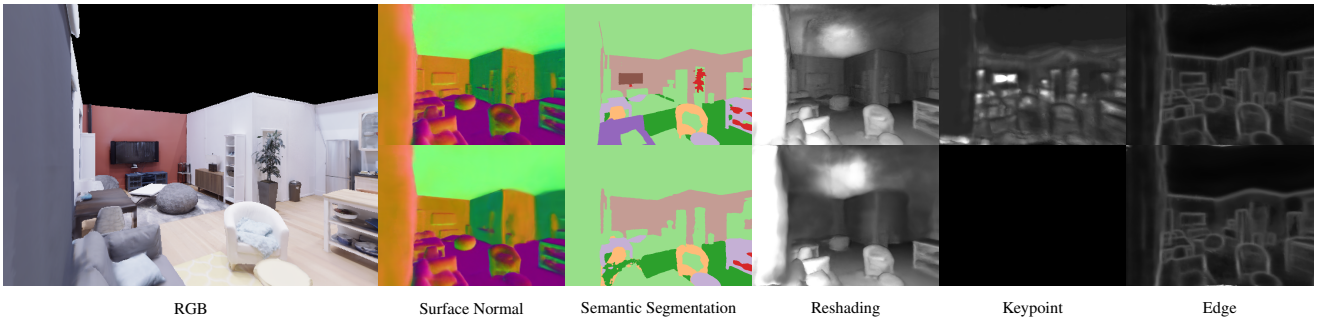

Figure 4. Qualitative comparison between models w/ and w/o color RGB image supervision. **Top row:** SaS-NeRF; **Bottom row:** SaS-NeRF$_{w/o\ c}$. RGB supervision is crucial for understanding the scenes and learning other visual tasks. Without RGB supervision, the model performs much worse, *e.g.*, for keypoint detection, when RGB supervision is missing, the model inevitably collapses, ending up predicting an all-zero map.

(1) ground-truth (GT); (2) ground-truth and the augmented data generated by our model (GT+SaS-NeRF). If we can reach a similar level of performance between GT and SaS-NeRF, and also achieve a better performance will GT+SaS-NeRF, then it shows that our rendered data has a similar distribution as the ground-truth and it can be thus useful for downstream discriminative tasks. For the GT+SaS-NeRF data setting, we generate paired data with the same poses as GT. For the task network, we adopt a standard Taskonomy encoding-decoding architecture [69]. Different from the main experiment, we combine all the data from the four scenes to train and evaluate the task network. We train all the models for 200 epochs.

The results are shown in Table 2. For the GT+SaS-NeRF data setting, we obtain better performance for three out of five tasks, and for the other two tasks, we can reach a similar performance. These results indicate that SaS-NeRF can generate both visually realistic and useful data, making it attractive to be applied to benefit the learning of other visual perception tasks.

Data Setting	SS(↑)	SN(↓)	RE(↓)	KP(↓)	ED(↓)
GT	0.8209	0.0189	0.0255	**0.0020**	**0.0092**
GT + SaS-NeRF	**0.8484**	**0.0186**	**0.0218**	**0.0020**	0.0093

Table 2. Comparison of the two data settings. SS: Semantic Segmentation; SN: Surface Normal; RE: Reshading; KP: 2D Keypoints; ED: Edge Detection; GT: paired ground-truth data; GT+SaS-NeRF: GT data and augmented data rendered by SaS-NeRF. For the segmentation task, we use mIoU as the evaluation metric; for the rest of the tasks, we adopt $\mathcal{L}1$ error as the evaluation metric. SaS-NeRF synthesizes both visually realistic and useful data, so it can be used as an effective way of data augmentation to benefit the learning of other visual tasks.

4.4.2 Task Relationship within Generative Models

Task relationship is crucial for multi-task visual learning, and it has been well studied in the frame of discriminative models [55, 69]. Investigating task relationships in a generative manner is largely under-explored, but it can help better understand the inner mechanism of machine vision from a complementary perspective. Our SaS-NeRF model can also simultaneously learn scene representations and share knowledge within multiple visual tasks, so as to further benefit individual tasks. Taking semantic segmentation (SS) as an example, we further build five model variants under different task settings to investigate – whether other tasks can benefit semantic segmentation in the framework of SaS-NeRF. We first add the other four tasks to jointly train our model with the SS task (named SS+"additional task"), and also build a model trained with all the five tasks together (SS+All). Besides the mIoU, we also measure the pixel accuracy (Acc) and mean class accuracy (Class Acc) for all the models. Notice that further combinations of the tasks and similar explorations for other tasks can also be conducted within the SaS-NeRF framework.

We show the results for the basic SS setting and the five variants in Table 3. We have the following observations regarding the results: (1) SN almost consistently benefits the target SS task for all scenes, while the reshading hurts. It may be because the surface normal can reflect the texture and geometry of the scene, so it makes the model better detect the semantics; by contrast, the shading represents higher-order relationships between the camera ray and the scene, which tend to be noise for SS. (2) Jointly training with all the tasks generally will not benefit the target task, since the model will be optimized to balance all the tasks rather than focusing on a single task. (3) Model performance also varies in different scenes, indicating that the task relationships might also rely on the scene structures, and the relationship among tasks might not be stationary for generative models. Interestingly, these observations are also consistent with those for discriminative models [55, 69].

4.4.3 Auto-Labelling for Real-World Scenes

One important application for the multi-task discriminative models is that they work as auto-labeller to annotate the real-world data, after pre-trained on synthetic or academic small-scale datasets. Our SaS-NeRF model can be used as auto-labller as well. Note that, different from discriminative models that directly operate on real images, our SaS-NeRF simultaneously renders images and their per-pixel scene property annotations. Considering this difference, we introduce a *two-stage* procedure for leveraging SaS-NeRF as auto-labller. With some pre-trained discriminative models, we first produce initial ground-truth annotations. Such annotations are not guaranteed to be correct and could be flawed – *e.g.*, they might be inconsistent across different views. Then, we train our SaS-NeRF with these weak annotations. Because SaS-NeRF can implicitly learn the 3D geometry and scene representation, it can *correct these inconsistencies* during network optimizations. This shows that SaS-NeRF can work as auto-labeller for real-world scenes, and most importantly, it can work to **refine** the noisy labels produced by the other models.

Based on this insight, we move to a real-world dataset without annotations – the LLFF dataset [32]. We also use the same pre-trained annotator [68] to generate weak annotations for this dataset (2nd and 4th columns in Figure 5). Due to the data distribution gap between LLFF and the Taskonomy dataset, the quality of these annotations is quite poor, *e.g.*, for the surface normal estimation, there are sharp faults in the object boundaries. Then we train SaS-NeRF with these flawed annotations and we show the results for surface normal and reshading on two scenes of the LLFF dataset in Figure 5. It is clear to see that our SaS-NeRF produces smoother results, contains more details and reflects better 3D structures of the scene. We argue that the refinement comes from the joint modeling and understanding of the scenes, inherent within the SaS-NeRF framework, showing the capability of our model in scene analysis. In addition, this general idea of auto-labelling and refinement can be in principle applied to other real-world data and jointly work with other discriminative models.

5. Discussion and Conclusion

Key Insights. This works shows that a comprehensive scene representation with implicitly encoded 3D geometry and scene structure which is powered by the NeRF-style architecture can be useful for not only RGB image synthesis tasks, but also for various visual tasks. Inspired by this, we propose a unified framework SaS-NeRF, which allows shared knowledge and feature representation across different tasks. This novel strategy of solving visual problems with a generative model provides a new viewpoint for multi-task learning, which is normally tackled in the context of

Setting	FRL_apt_2			FRL_apt_4			Office_0			Office_3		
	Acc	Class Acc	mIOU	Acc	Class Acc	mIOU	Acc	Class Acc	mIOU	Acc	Class Acc	mIOU
SS	0.9830	0.9072	0.8926	0.9722	0.9155	0.8552	0.9863	0.9029	0.8969	0.9771	0.8578	0.8426
SS + SN	**0.9851**	**0.9162**	**0.9056**	0.9691	**0.9270**	**0.9098**	**0.9921**	**0.9067**	**0.9016**	**0.9810**	**0.9105**	**0.8927**
SS + RE	0.9791	0.8747	0.8582	0.9689	**0.9296**	**0.9091**	0.9850	**0.9055**	0.8878	0.9760	0.8576	0.8395
SS + KP	0.9791	0.8709	0.8582	0.9640	**0.9291**	**0.8985**	**0.9870**	**0.9040**	0.8967	**0.9819**	**0.9107**	**0.8848**
SS + ED	**0.9913**	**0.9547**	**0.9479**	0.9722	0.9015	0.8507	0.9699	0.8951	0.8812	**0.9823**	**0.9117**	**0.8982**
SS + All	**0.9867**	**0.9495**	**0.9390**	0.9663	**0.9242**	**0.9038**	0.9510	0.8292	0.8005	0.9395	0.8236	0.7755

Table 3. Model performance with additional tasks for semantic segmentation. SS: semantic segmentation; SN: surface normal; RE: reshading; KP: 2D keypoints; ED: edge detection; All: all the four additional tasks. Bold and red numbers indicate performance increasing, while blue numbers indicate performance drop. SN almost consistently benefits the target SS task for all scenes, indicating a closer relationship between these two tasks.

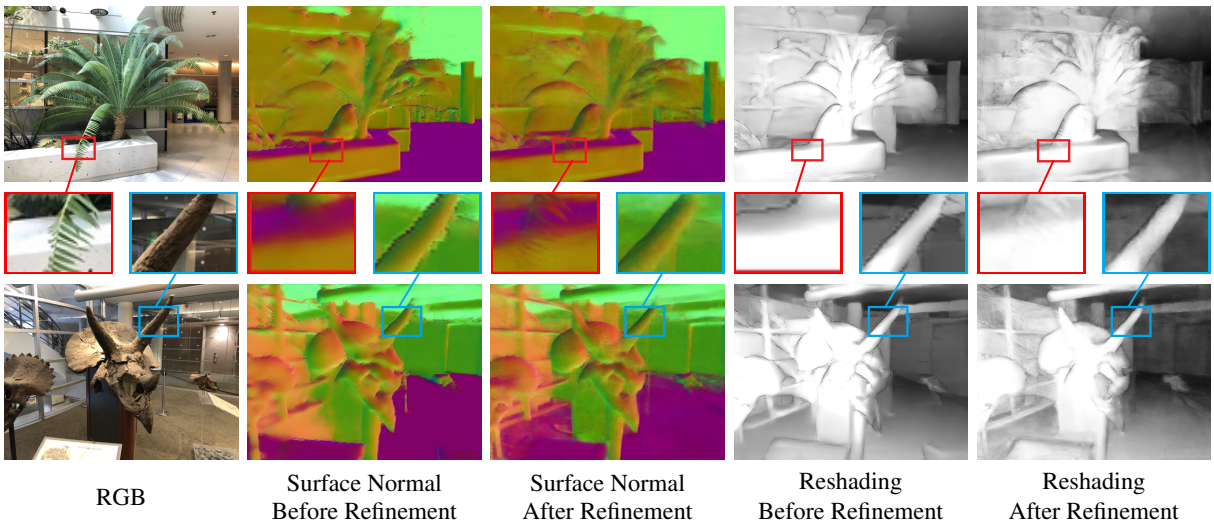

Figure 5. Surface normal and reshading predictions with real-world images from the LLFF dataset. We use pre-trained annotators to obtain the initial labels which are noisy and flawed, and we retrain SaS-NeRF with these labels. SaS-NeRF can refine these flawed annotations and restore more details by joint modeling and understanding of the scenes.

discriminative models. We further show that some interesting observations and analysis, such as the task relationships with generative models, can be made via our SaS-NeRF framework.

Applications. In Section 4.4, we explored some applications of SaS-NeRF. Intuitively, other applications could be also possible within SaS-NeRF, such as (1) knowledge transfer from task to task, and from scene to scene, (2) super-resolution and de-noising, and (3) jointly learning with discriminative models to solve multiple visual tasks.

Limitations. One major limitation of SaS-NeRF is its generalization capability. SaS-NeRF builds upon the original NeRF model. NeRF itself is a scene-dependent model, making it hard to transfer the learned knowledge from one scene to another. Following works [1, 20, 51, 66] provide some solutions to enhance the generalizability of NeRF. We believe similar techniques can be applied to SaS-NeRF. Another limitation is that SaS-NeRF requires accurate pose annotations to learn scene representations, which might not be accessible for all the datasets. In our main experiments on Replica [56], we used COLMAP [49, 50] to estimate the poses. We believe powerful pose estimation methods can help us overcome this limitation to some extent. Also, NeRF - - [60] suggests how to jointly learn the pose together with the RGB synthesis, which could further enhance our model capability.

Future Work. We will overcome the limitations mentioned above and explore additional applications of SaS-NeRF. We also plan to investigate *conditional* scene property generation – learning the shared knowledge across scenes. We hope SaS-NeRF can be adopted in the community as a general computational tool for scene analysis by synthesis.

Societal Impact. We do not see any immediate negative societal impact of our work. The potential negative impact is likely the same as other research on image synthesis.

References

[1] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. In *NeurIPS*, 2021. 2, 8

[2] Zhipeng Bao, Martial Hebert, and Yu-Xiong Wang. Generative modeling for multi-task visual learning. *arXiv preprint arXiv:2106.13409*, 2021. 3, 5

[3] Zhipeng Bao, Yu-Xiong Wang, and Martial Hebert. Bowtie networks: Generative modeling for joint few-shot recognition and novel-view synthesis. In *ICLR*, 2021. 2

[4] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018. 2

[5] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *ECCV*, 2020. 2

[6] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 2

[7] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *ECCV*, 2020. 2

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016. 2

[9] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, 2018. 2

[10] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 3

[11] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021. 1, 2

[12] Jeevan Devaranjan, Amlan Kar, and Sanja Fidler. Metasim2: Unsupervised learning of scene structure for synthetic data generation. In *ECCV*, 2020. 5

[13] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 3

[14] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 5

[15] Rui Fan, Hengli Wang, Peide Cai, and Ming Liu. Sne-roadseg: Incorporating surface normal information into semantic segmentation for accurate freespace detection. In *ECCV*, 2020. 2

[16] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *arXiv preprint arXiv:2103.10380*, 2021. 1, 2

[17] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, 2019. 2

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[19] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019. 2

[20] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 2, 8

[21] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. 2

[22] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *ICCV*, 2019. 1, 2

[23] Paul Henderson, Vagia Tsiminaki, and Christoph H Lampert. Leveraging 2d data to learn textured 3d mesh generation. In *CVPR*, 2020. 2

[24] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017. 2

[25] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, 2021. 1, 2

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2

[27] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[28] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 3

[29] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *ECCV*, 2020. 2

[30] David B Lindell, Julien NP Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 1, 2

[31] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 1

[32] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4), 2019. 4, 7

[33] 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 *ECCV*, 2020. 1, 2, 3, 5

[34] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. 3

[35] Muzammal Naseer, Salman Khan, and Fatih Porikli. Indoor scene understanding in 2.5/3d for autonomous agents: A survey. *IEEE access*, 2018. 1

[36] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5

[37] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019. 2

[38] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 1, 2

[39] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, 2019. 2

[40] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *arXiv preprint arXiv:2104.10078*, 2021. 2

[41] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, 2021. 2

[42] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2

[43] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. In *ECCV*, 2020. 2

[44] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020. 2

[45] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *arXiv preprint arXiv:2103.13744*, 2021. 1, 2

[46] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 3

[47] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. 2

[48] 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 *CVPR*, 2019. 4

[49] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5, 8

[50] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 5, 8

[51] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *NeurIPS*, 2020. 1, 2, 8

[52] Jianxiong Shen, Adria Ruiz, Antonio Agudo, and Francesc Moreno-Noguer. Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. *arXiv preprint arXiv:2109.02123*, 2021. 1

[53] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 33, 2020. 2

[54] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

[55] Trevor Standley, Amir R Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020. 3, 4, 5, 7

[56] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 4, 8

[57] Ximeng Sun, Rameswar Panda, and Rogerio Feris. Adashare: Learning what to share for efficient deep multi-task learning. *arXiv preprint arXiv:1911.12423*, 2019. 3

[58] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *arXiv preprint arXiv:2106.14405*, 2021. 4

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

[60] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 8

[61] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 2

[62] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NeurIPS*, 2016. 2

[63] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021. 2

[64] Yongxin Yang and Timothy M Hospedales. Trace norm regularised deep multi-task learning. In *ICLR*, 2017. 3

[65] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020. 2

[66] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 8

[67] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020. 3

[68] Amir R. Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J. Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020. 5, 7

[69] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 2, 3, 4, 5, 6, 7

[70] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 1, 2

[71] Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. Modular generative adversarial networks. In *ECCV*, 2018. 2

[72] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. *arXiv preprint arXiv:2103.15875*, 2021. 2, 4

[73] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. In *ECCV*, 2020. 2

[74] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representation. In *NeurIPS*, 2018. 2