Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement

Prafull Sharma1∗ Ayush Tewari1 Yilun Du1
Sergey Zakharov4 Rares Ambrus4 Adrien Gaidon4
William T. Freeman1 Frédo Durand1 Joshua B. Tenenbaum1,2,3 Vincent Sitzmann1

1MIT CSAIL 2MIT BCS 3MIT CBMM 4Toyota Research Institute
prafullsharma.net/see3d

Abstract

Human perception reliably identifies movable and immovable parts of 3D scenes, and completes the 3D structure of objects and background from incomplete observations. We learn this skill not via labeled examples, but simply by observing objects move. In this work, we propose an approach that observes unlabeled multi-view videos at training time and learns to map a single image observation of a complex scene, such as a street with cars, to a 3D neural scene representation that is disentangled into movable and immovable parts while plausibly completing its 3D structure. We separately parameterize movable and immovable scene parts via 2D neural ground plans. These ground plans are 2D grids of features aligned with the ground plane that can be locally decoded into 3D neural radiance fields. Our model is trained self-supervised via neural rendering. We demonstrate that the structure inherent to our disentangled 3D representation enables a variety of downstream tasks in street-scale 3D scenes using simple heuristics, such as extraction of object-centric 3D representations, novel view synthesis, instance segmentation, and 3D bounding box prediction, highlighting its value as a backbone for data-efficient 3D scene understanding models. This disentanglement further enables scene editing via object manipulation such as deletion, insertion, and rigid-body motion.

1 Introduction

Parsing a scene into movable objects and immovable background is a critical aspect of visual perception. Humans succeed at this task given just a single, static image. Furthermore, our perception is not restricted to 2D, as we are capable of forming a belief over the 3D geometry of the occluded parts of objects, such as a mug or a car, given a partial observation from only a single viewpoint. These fundamental skills of scene understanding are largely self-supervised, and learned simply by moving in and interacting with our 3D environment. In this work, we present a self-supervised approach which aims to solve the same problem by learning to reconstruct representations of 3D scenes that disentangle static background from movable objects, while completing occluded regions. Inspired by research on human perception that suggests that motion is a critical cue to learn our prior for object discovery, we leverage motion as an objectness cue and train on multi-view videos. At test time, our model can reconstruct disentangled 3D representations from just a single static image.

∗Email: prafull@mit.edu, vsitzmann@mit.edu

Preprint. Under review.
Recent work in self-supervised learning has made significant progress towards the goal of object discovery via self-supervised object-centric representation learning for images [1, 2] and videos [3]. These methods segment an image or video into non-overlapping objects and infer a latent code for each of them, however, they are either constrained to simple toy environments, or require video with additional annotations, such as bounding boxes, at test time. The resulting object-centric latent codes can be decoded into object-centric 3D scene representations [4–6]. Here, 3D notions, such as 3D scale and connectedness, can serve as additional training signal, but methods are similarly limited to simple scenes. Our work is further inspired by recent work on learning 3D representations of dynamic scenes [7–9]. While most of these approaches are focused on high-quality novel-view synthesis, a few also enable disentanglement of static and dynamic components [10, 11]. However, most of these approaches are overfit to a single scene, and can thus only disentangle movable and immovable scene parts in that single scene, without acquiring knowledge about objectness that generalizes across scenes. They further do not support inference from partial observations, such as a single image.

In this work, we propose an approach that learns to map a single image to a 3D neural scene representation that is disentangled into static background and movable object 3D representations. Specifically, given a single image, our model infers a neural ground plan, a 2D grid of learned features aligned with the ground plane. Continuous 3D points may be decoded for volume rendering by projecting them onto the ground plan, retrieving the respective features, and decoding them via an MLP. By observing multi-view video at training time, our model learns an objectness prior that, at test time, enables it to map a single image observation to separate static “background” and movable “object” ground plans. Our model is trained self-supervised via neural rendering, without pseudo-ground truth, bounding boxes, or any instance labels.

Neural ground plans exploit the fact that in scenes with physical objects moving mostly under their own power, such as streets with cars, pedestrians, and bicyclists, as a consequence of gravity, most objects move on the 2D ground plane. Here, neural ground plans offer a dense, yet memory-efficient neural scene representation. Ground plans have been shown to be effective 3D representation in robotics and autonomous driving applications [12], as well as in computer graphics for unconditional synthesis of indoor scenes [13]. Our method shows how we can reconstruct disentangled neural ground plans from monocular images of unbounded scenes using self-supervised learning via neural rendering. We demonstrate that our model enables instance segmentation, recovery of 3D object-centric representations, and 3D bounding box prediction via a simple heuristic leveraging that connected regions of 3D space that move together belong to the same object. The proposed representation also enables intuitive editing of the scenes using manipulation of individual objects. On a dataset of high-quality renderings of street-scale scenes, our model outperforms prior pixel-aligned approaches in novel view synthesis fidelity as well as prior work on the self-supervised discovery of object-centric 3D representations in terms of object discovery. In summary, our contributions are:

- We introduce self-supervised training of conditional neural ground plans, a hybrid discrete-continuous 3D neural scene representation that can be reconstructed from a single image, enabling efficient processing of scene appearance and geometry directly in 3D.

- We leverage object motion as a cue to learn to disentangle static background and movable foreground objects given only a single input image.
• Using a simple hand-crafted heuristic, our structured neural scene representation enables strong single-image 3D instance segmentation and 3D bounding box prediction, demonstrating its value as a backbone for data-efficient prior-based 3D processing.

2 Related Work

Neural Scene Representation and Rendering. Several works have explored learning neural scene representations for downstream tasks in 3D. Earlier approaches used voxel grids as the 3D representation [14, 15]. However, voxel grids are memory intensive, and thus, it is difficult to scale these methods to high resolutions. Emerging neural scene representations enable self-supervised reconstruction of geometry and appearance at high-resolutions given only image observations. A large part of recent work focuses on the case of reconstructing a single 3D scene given dense observations [16–20], enabling high-quality novel view synthesis with exciting applications in computer graphics. Alternatively, differentiable neural rendering may be used to supervise encoders to enable 3D reconstruction from incomplete image observations [21–30]. Fu and Zhang et al. [31] use neural rendering as a tool to recover high-quality 2D panoptic segmentation annotations from a set of sparse images and noisy 3D bounding primitives and 2D predictions. Hybrid discrete-continuous neural scene representations enable faster rendering [32–36]. Neural ground plans and axis-aligned 2D grids enable high-quality unconditional generation of 3D scenes [13, 32]. Axis-aligned feature grids have also been used for reconstruction of 3D geometry from pointclouds [37]. We similarly use axis-aligned 2D grids of features for self-supervised scene representation via neural rendering, but reconstruct them directly from few or a single 2D image observations.

Birds-Eye View Representations. Birds-eye view has been explored as a 3D representation in robotics, particularly for autonomous driving applications. Prior work uses ground-plane 2D grids as representations for object detection and segmentation [12, 38–41], layout generation and completion [42–45], and next-frame prediction [46, 47]. The birds-eye view is generated either directly without 3D inductive biases [44], or similar to our proposed approach, by using 3D geometry-driven inductive biases such as unprojection into a volume [38, 48, 41], or by generating a 3D point cloud [39, 46]. However, prior approaches are supervised, using ground truth bounding boxes or semantic segmentation as supervision. In contrast, we present the first self-supervised conditional ground plan representation, learned only from posed images via neural rendering. While we show that our self-supervised representation can be used for rich inference tasks using simple heuristics, our method may be extended for more challenging tasks using the techniques developed in prior work.

Dynamic-Static Disentanglement. Our work is related to prior art on learning to disentangle dynamic objects and static background. Some prior work leverages object motion across video frames to learn separate representations for movable foreground and static background in 2D [49–51], while other recent work can also learn 3D representations [10, 11]. Our approach is similar in using object motion as cue for disentanglement and multi-view as cue for 3D reconstruction, but uses it as supervision to train an encoder-based approach that enables reconstruction from a single image instead of scene-specific disentanglement from multiple video frames.

Object-centric Scene Representations. Prior work has aimed to infer object-centric representations directly from images, with objects either represented as localized object-centric patches [52–56] or scene mixture components [2, 57–61], with the slot attention module [1] increasingly driving object-centric inference. Resulting object representations may be decoded into object-centric 3D representations and composed for novel view synthesis [4, 6, 62–68]. BlockGAN and GIRAFFE [69, 70] build unconditional generative models for compositions of 3D-structured representations, but only tackle generation, not reconstruction. Some methods rely on annotations such as bounding boxes, object classes, 3D object models, or instance segmentation to recover object-centric neural radiance fields [71–74]. Several scene reconstruction methods [65–67, 75] use direct supervision to train an object representation and detector to infer an editable 3D scene from a single frame observation. Kipf et al. [3] leverage motion as a cue for self-supervised object disentanglement, but do not reconstruct 3D and require additional conditioning in the form of bounding boxes. We propose to reconstruct separate 3D representations of movable objects and static background in 3D prior to inferring object-centric representations. This drastically simplifies object discovery, and we demonstrate that a simple heuristic is sufficient to perform discovery of object-centric representations in street-scale scenes. However, we highlight that our work is complimentary to the work discussed...
Figure 2: **Ground plan inference.** Given a context image, we first extract a set of CNN features. We unproject the features into 3D and re-sample them at “pillars” on top of the location of ground plan vertices. Pillars are aggregated into ground plan features using a softmax-weighted sum. The resulting 2D grid of features is decomposed into separate dynamic and static ground plans by a 2D CNN. The coordinate-encoding MLP is not visualized in this figure. Please refer to Sec. 3 for details.

here: slot attention and related object-centric algorithms can be run on our already sparse ground plan of movable 3D regions, faced with a dramatically easier task than when run on images directly.

## 3 Conditional Neural Ground Plans

We introduce conditional neural ground plans as a hybrid discrete-continuous scene representation for 3D scene understanding, reconstructed from single images and trained via multi-view video. We first describe the case without static-dynamic disentanglement (“Entangled Neural Ground Plan” in Fig. 3), which we will discuss in the next section. Please see Fig. 2 for an illustration of the reconstruction of a neural ground plan from a single image.

### Compactified neural ground plans for unbounded scene representations.

A neural ground plan is a 2D grid of features aligned with the ground plane of the 3D scene, which we define to be the $xz$–plane. A 3D point is decoded by projecting it onto the ground plane and retrieving the corresponding feature vector using bilinear interpolation. This feature is then concatenated with the vertical $y$-coordinate of the query point and decoded into radiance and density values via a fully connected network, enabling novel view synthesis using volume rendering. In this definition, however, it is only possible to decode 3D points that lie within the boundaries of the neural ground plan, which precludes reconstruction and representation of unbounded scenes. We thus compactify $\mathbb{R}^2$ by implementing a non-linear coordinate re-mapping as proposed in [76]. $xz$-coordinates within a radius $r_{\text{inner}}$ around the ground plan origin remain unaffected, but $xz$-coordinates of points outside this radius are contracted. For any point $p \in \mathbb{R}^2$ on the ground plan, the contracted coordinate can be computed as $p' = ((1 + k) - k/\|u\|)(u/\|u\|)r_{\text{inner}}$, where $u = p/r_{\text{inner}}$, and $k$ is a hyperparameter which controls the size of the contracted region.

### Assumptions and benefits.

Parameterizing the scene as 2D neural ground plans has several advantages over alternative scene representations. Compared to parameterizing the entire scene as a single, monolithic MLP, rendering is significantly cheaper, as in other hybrid discrete-continuous neural fields [77]. Note that while it is possible for a feature to parameterize more than one object per tile, it is difficult to reconstruct stacks of unseen numbers of objects. When reasoning about dynamics, the neural ground plan encodes an inductive bias that most motion happens in the ground plane. In applications such as self-driving, a ground plan is an appropriate memory-efficient representation. For other tasks, such as stacking of boxes, it is prudent to expand the ground plan to a voxel grid along the $y$-axis. A core benefit of both voxel grids and ground plans as 3D representations is that they enable shift-equivariant processing of the 3D scene via convolutions, without concern for occlusions and perspective distortion, and enable straightforward editing of the 3D scene.

### Reconstructing neural ground plans from images.

Inferring a neural ground plan from one or several images proceeds in three steps: (1) feature extraction, (2) feature unprojection, (3) pillar aggregation. Given a single image $I$, we first extract per-pixel features via a CNN encoder to yield a feature tensor $F$. We define the camera as the world origin and center the neural ground plan accordingly, approximately aligned with the ground level. We unproject the image features along their respective rays as parameterized via the intrinsic and extrinsic camera parameters to create a
Figure 3: **Learning Static-Dynamic Disentanglement.** Given multiple frames of a video, we extract per-frame, compactified static and dynamic ground plans according to Fig. 2. Static ground plans are pooled into a time-invariant ground plan. We then composite per-frame dynamic and static time-invariant ground plans via differentiable volume rendering. Our model is supervised only via a re-rendering loss on video frames. We encourage the model to explain as much of the scene density as possible with the static ground plan via a sparsity loss on per-frame dynamic volume rendering densities. The surface loss is not visualized here.

Disentangling static and dynamic neural ground plans. We leverage multi-view video as the training signal. We pick two frames of a video. For each frame, we infer an entangled neural ground plan as described in the previous section. Features in this entangled neural ground plan parameterize both static and dynamic features of the scene, for instance, a car as well as the road below it. We feed this ground plan into a fully convolutional 2D network, which disentangles it into two separate ground plans containing dynamic and static features. The per-frame static ground plans are pooled to obtain a single, time-independent static ground plan.

**4 Learning Static-Dynamic Disentanglement**

We now describe how motion over time is leveraged to disentangle dynamic and static scene parts, and how we use the resulting scene representation for self-supervised 3D object discovery and 3D bounding box prediction in street scenes via a simple heuristic. Please see Fig. 3 for an overview of the multi-frame training for dynamic-static disentanglement.
Figure 4: Single-image reconstruction, disentanglement of static and dynamic objects, and novel view synthesis. Given a single input image, our method can disentangle the observed scene into static and object components based on what the model observed as not-moving and moving in the training data. In these examples, the cars are isolated in the object component as the model was training on video data of cars moving on the road.

The $xz-$coordinates in the entangled ground plan are linearly spaced. The CNN also transforms the ground plans to the compactified $\mathbb{R}^2$ space as explained earlier. Vertices in this compactified space are arranged on regular grids close to the camera and grow increasingly sparse with increasing distance. Note that a 2D CNN in the ground plan space is a 3D operation, since ground plans implicitly parameterize 3D scene representations.

**Compositing ground plans.** To render a scene disentangled into static and dynamic ground plans, we first decode query points using both ground plans, yielding two sets of (density, color) values for each point. We now follow STaR [10] to compose the contribution from static and dynamic components along the ray. Given the color and density for static $(c^s, \sigma^s)$ and dynamic $(c^d, \sigma^d)$ parts, the density of the combined scene is calculated as $\sigma^s + \sigma^d$. The color at the sampled point is computed as a weighted linear combination $w^s c^s + w^d c^d$, where $w^s = (1 - \exp(-\delta \sigma^s))/(1 - \exp(-\delta (\sigma^s + \sigma^d)))$, $w^d = (1 - \exp(-\delta \sigma^d))/(1 - \exp(-\delta (\sigma^s + \sigma^d)))$, and $\delta$ is the distance between adjacent samples on the camera ray.

**Losses and Training.** We train our model on multi-view video, where multi-view information is used to learn 3D structure, while motion is used to disentangle the static and dynamic components in the scene. During training, we sample two time-steps per video. For each time-step, we sample multiple images from different camera views; some of the views are used as input to the method while others are used to compute the loss function. We use the input images to infer static and dynamic ground plans, and use them to render out per-frame query views. Our per-frame loss consists of an image reconstruction term, a hard surface constraint, and a sparsity term.

$$L = \frac{||R - I||^2_2}{\mathcal{L}_{\text{img}}} + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}}(R, I) - \lambda_{\text{surface}} \sum_i \log(\mathbb{P}(w_i)) + \lambda_{\text{sparse}} \sum_i |\sigma_i^D|. \quad (1)$$

$\mathcal{L}_{\text{img}}$ measures the difference between the rendered and ground truth images, $R$ and $I$ respectively, using a combination of $\ell_2$ and patch-based LPIPS perceptual loss. $\mathcal{L}_{\text{surface}}$ encourages both static and dynamic weight values (the weight for each sample in the rendering equation) $w_i$ for all samples along the rendered rays to be either 0 or 1, encouraging hard surfaces [79]. Here, $\mathbb{P}(w_i) = \exp(-w_i) + \exp(-1 - w_i)$. The sparsity term $\mathcal{L}_{\text{sparsity}}$ takes as input densities decoded from the dynamic ground plan from all rendered rays, and encourages the values to be sparse. This forces the model to explain as much non-empty 3D structure as possible via the static ground plan, leading to reliable...
static-dynamic disentanglement. Without this loss, the model could explain the entire scene with just the dynamic component. The loss functions are weighed using the hyperparameters $\lambda_{LPIPS}$, $\lambda_{surface}$ and $\lambda_{sparse}$. While we describe the loss functions for a single sample of ground-truth and rendered image, in practice, we construct mini-batches by randomly choosing multiple views of a scene at different time steps, and evaluate the loss function on each sample.

Unsupervised object detection and extracting object-centric 3D representations. Our formulation yields a model that maps a single image to two radiance fields, parameterizing static and dynamic 3D regions respectively. Please see Fig. 1 for an example. We now perform a simple search for connected components in the dynamic neural ground plan, performing unsupervised 3D instance segmentation, monocular 3D bounding box prediction, and the extraction of object-centric 3D representations. Specifically, given a dynamic ground plan, we first sample points in a 3D grid around the ground plan origin and decode their densities. We now perform conventional connected-component labeling in the ground plane space using accumulated density values, identifying disconnected dynamic objects. Using volume rendering, we can perform screen-space instance segmentation, see Fig. 1 for an example. 3D bounding boxes for each instance can be extracted by recovering the smallest box containing some percentage of the total density of the connected component. Finally, we may simply crop tiles of the dynamic ground plan that belong to a given object instance, yielding object-centric 3D representations, enabling editing of 3D scenes such as deletion, insertion, and rigid-body transformation of objects. This approach is not limited to a fixed number of objects during training or at test time. As we will show, this simple method is at-par with the state of the art on self-supervised learning of object-centric 3D representations, uORF [4]. However, this simple heuristic is not capable of segmenting objects that are in physical contact. In self-driving scenarios, this is of limited concern, but in general 3D scenes, this assumption is regularly violated. Note that our approach is entirely compatible with prior work leveraging slot attention [1, 3] and other inference modules, which can simply be run on the disentangled dynamic ground plan, which guarantees a structured and sparse semantic space that drastically simplifies instance segmentation.
Table 1: Quantitative baseline comparison of novel view synthesis results. We outperform PixelNeRF [22] and uORF [4] in terms of PSNR, SSIM, and LPIPS on both CLEVR and CoSY datasets.

<table>
<thead>
<tr>
<th></th>
<th>CLEVR (1 view)</th>
<th>CoSY (1 view)</th>
<th>CoSY (5 views)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR ▲</td>
<td>SSIM ▲</td>
<td>LPIPS ▼</td>
</tr>
<tr>
<td>Ours</td>
<td>34.5 0.956 0.15</td>
<td>15.71 0.43 0.53</td>
<td>18.29 0.57 0.43</td>
</tr>
<tr>
<td>PixelNeRF</td>
<td>33.98 0.945 0.200</td>
<td>14.61 0.34 0.64</td>
<td>17.31 0.49 0.50</td>
</tr>
<tr>
<td>uORF</td>
<td>29.35 0.898 0.151</td>
<td>— — —</td>
<td>— — —</td>
</tr>
</tbody>
</table>

Figure 6: **Qualitative comparisons.** Left two columns: Novel-view synthesis comparison with uORF [4] and PixelNeRF [22]. Right: Instance segmentation comparison with uORF [4].

## 5 Results

We demonstrate that our method can reconstruct street-scale 3D scenes using just a single image observation, disentangling dynamic foreground objects and static background. Using the proposed heuristic, we show that this can be leveraged for unsupervised instance segmentation and 3D bounding box detection. Our method achieves state-of-the-art performance both in terms of novel view synthesis as well as unsupervised 3D object discovery. We also demonstrate high-quality editing of the scene via manipulation of the individual objects. We encourage readers to refer to the supplemental material for more results, including video results.

### Datasets

Our method is trained on multi-view observations of dynamic scenes. To this end, we leverage CoSY [80], a procedural generator of street scenes, and render multi-view observations of 9000 scenes with moving cars, sampled using 15 background city models and 95 car models. Due to the procedural generation of the city with finite assets, different city instances use the same building geometry and textures but in different combinations. We train on 8000 scenes, and evenly split the rest into validation and test sets. We use the CLEVR dataset [4] for self-supervised object discovery benchmarks. Datasets and code will be made public.

### Baselines

For single-shot 3D reconstruction and novel view synthesis, we compare against PixelNeRF [22], a state-of-the-art single-image 3D reconstruction method. For unsupervised object-centric 3D reconstruction, we compare against the state-of-the-art method uORF [4]. We train PixelNeRF models on our datasets using publicly available code. We finetune the uORF model pretrained on CLEVR on our CLEVR renderings, and train it from scratch on CoSY using publicly available code.

### Single-image Static and Dynamic 3D Reconstruction

Fig. 4 shows results on single-image reconstruction of static and dynamic scene elements. Given only a single image, our method reliably segments the scene into static and dynamic parts. It plausibly completes parts of the scene that are unobserved in the context image but sufficiently constrained, such as the back-side of cars or the unobserved patch of road beneath a car. As expected from a non-generative method, regions that are entirely unconstrained such as occluded parts of the background are blurry. Fig. 6 provides a qualitative comparison to PixelNeRF and uORF in terms of single-image 3D novel view synthesis on both CoSY and CLEVR. While PixelNeRF succeeds in novel view synthesis on CLEVR, renderings on the complex CoSY dataset show significant artifacts, which may be due to the linear sampling employed by PixelNeRF.
Input Reconstruction Individual Objects Deletion Addition Rearrangement

Figure 7: Object-centric representations and scene editing. The proposed static-dynamic neural ground plans enables object discovery using a simple heuristic of connectedness of objects (left). This enables straight-forward scene editing such as deletion, addition, or rigid-body transformation via directly editing the neural ground plan (right).

uORF does not synthesize realistic images when trained on CoSY (see supplemental for results). On CLEVR, uORF generally produces high-quality renderings, but lacks high-frequency detail. In contrast to these methods, our method reliably synthesizes novel views with high-frequency detail for both datasets, while also disentangling the scene into movable and immovable scene parts. Quantitatively, we outperform both methods on novel-view synthesis in terms of PSNR, SSIM and LPIPS metrics on both datasets, please see Table 1 for these results. Our method supports monocular 3D reconstruction, as well as multi-view reconstruction at test time. We demonstrate in Table 1 that our monocular results, while slightly lower-quality, are still comparable to reconstructions computed using 5 input views.

Seeing 3D Objects. Fig. 5 demonstrates the results of the proposed 3D object discovery heuristic based on connected dynamic component discovery in the birds-eye view, instance segmentation, and 3D bounding box prediction. Note that our method enables high-quality novel view synthesis from a birds-eye view. Fig. 6 provides a qualitative comparison of object discovery with uORF. While uORF succeeds at segmenting CLEVR scenes with fidelity comparable to ours, it fails to provide reconstruction and instance segmentation for our diverse and visually complex street-scale CoSY dataset. Our method reliably segments separate car instances and predicts 3D bounding boxes, including for cars that are only partially observed. Table 2 quantitatively compares the computed segmentation maps on CLEVR to uORF. We use the Adjusted Rand Index (ARI) metrics following uORF. We evaluate this metric in the input view (ARI), as well as in a novel view (NV-ARI). We perform at-par with uORF on both of these metrics, demonstrating that our 3D ground plan representation reaches state of the art results with simple heuristics. In addition, as mentioned before, we achieve higher-quality novel-view synthesis results, and also achieve significantly better results on the challenging CoSY dataset. Our method is more efficient than uORF, both for training and inference. Please see the supplemental material for details.

3D Scene Editing. Fig. 7 provides editing results of our method. The instance-level segmentation, dynamic-static disentanglement, and 3D bounding boxes enable straight-forward 3D editing, such as

<table>
<thead>
<tr>
<th>ARI↑</th>
<th>NV-ARI↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.84</td>
</tr>
<tr>
<td>uORF</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Table 2: Quantitative segmentation accuracy evaluation on CLEVR using ARI and NV-ARI metrics.
translation, rotation, deletion, and insertion of individual objects in the scene. Note that such editing is difficult with methods that lack a persistent 3D representation, such as PixelNeRF.

6 Discussion

Limitations and Future Work. Although our method achieves high-quality novel view synthesis from a single image, generated views are not photorealistic, and unobserved scene parts are blurry commensurate with the amount of uncertainty. Future work may explore plausible hallucinations of unobserved scene parts. As mentioned earlier, the heuristic employed to discover objects from the ground plan cannot identify separate objects that are in contact with each other; a limitation that could be addressed by running existing object-centric encoders [1] directly on the object ground plan. Since our model is trained on a finite set of cars and city configurations, our model does not show strong generalization to cars with out-of-distribution geometry (e.g. bus, truck) and texture (e.g. police car).

Ethical considerations. Our method could reduce labeling cost for tracking and detection, and poses a threat to be misused for surveillance.

Conclusion. Our paper demonstrates self-supervised learning of 3D scene representations that are disentangled into movable and immovable scene elements. While our method is trained on multi-view videos, at test time, we can reconstruct disentangled 3D representations from a single image observation. We show that neural ground plans serve as a rich representation that enable data-efficient solutions to downstream object-centric tasks such as instance segmentation, 3D bounding box prediction, and 3D scene editing. We hope that our paper will inspire future work on the use of self-supervised static-dynamic neural scene representations for general scene understanding tasks.

7 Acknowledgements and Disclosure of Funding

This work is in part supported by DARPA under CW3031624 (Transfer, Augmentation and Automatic Learning with Less Labels) and the Machine Common Sense program, Singapore DSTA under DST000ECI20300823 (New Representations for Vision), NSF CCRI # 2120095, Stanford Institute for Human-Centered Artificial Intelligence (HAI), Stanford Center for Integrated Facility Engineering (CIFE), Qualcomm Innovation Fellowship (QIF), Samsung, Ford, Amazon, and Meta. The Toyota Research Institute also partially supported this work. This article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.
References


