

Closing the Visual Sim-to-Real Gap with Object-Composable NeRFs

Nikhil Mishra^{*,1,2}, Maximilian Sieb³, Pieter Abbeel^{1,2}, Xi Chen¹

Abstract—Deep learning methods for perception are the cornerstone of many robotic systems. Despite their potential for impressive performance, obtaining real-world training data is expensive, and can be impractically difficult for some tasks. Sim-to-real transfer with domain randomization offers a potential workaround, but often requires extensive manual tuning and results in models that are brittle to distribution shift between sim and real. In this work, we introduce Composable Object Volume NeRF (COV-NeRF), an object-composable NeRF model that is the centerpiece of a real-to-sim pipeline for synthesizing training data targeted to scenes and objects from the real world. COV-NeRF extracts objects from real images and composes them into new scenes, generating photorealistic renderings and many types of 2D and 3D supervision, including depth maps, segmentation masks, and meshes. We show that COV-NeRF matches the rendering quality of modern NeRF methods, and can be used to rapidly close the sim-to-real gap across a variety of perceptual modalities.

I. INTRODUCTION

Nearly all applications in robotics require perception of the physical world, and deep learning is the method of choice for nearly all tasks in computer vision. As neural network architectures and training recipes have matured, limits on training data have become the bottleneck for these methods. Even if a large dataset is available, deployed systems may encounter scenarios that their original training did not prepare them for. Moreover, while some tasks can be supervised via human annotation (for example, object detection or semantic segmentation), other tasks, like depth estimation or shape completion, are best supervised in simulation.

However, when models trained in simulation are transferred to the real-world, their performance often degrades because the input distribution has shifted (a well-documented phenomenon known as the sim-to-real gap). Domain randomization [14] is typically used to improve robustness to out-of-distribution inputs: if a model is trained to generalize to different parameters of the simulator, such as viewpoint, lighting, or material properties, then it may also generalize to the real world as simply another randomization. However, this requires the training distribution to be both diverse enough to enable generalization, as well as realistic enough that the real world is a plausible sample from this distribution. This can be difficult to achieve in practice, often requiring manual asset creation (meshes, textures, materials), scene construction, and parameter tuning. Moreover, if a large sim-to-real gap is observed, it can be difficult to know exactly how to improve the simulator, which may require specialized expertise (e.g. graphics for visual tasks) or domain/task-specific tuning.

*Correspondence to: nmishra@berkeley.edu.

¹Covariant.ai, ²UC Berkeley, ³Work done at Covariant.ai

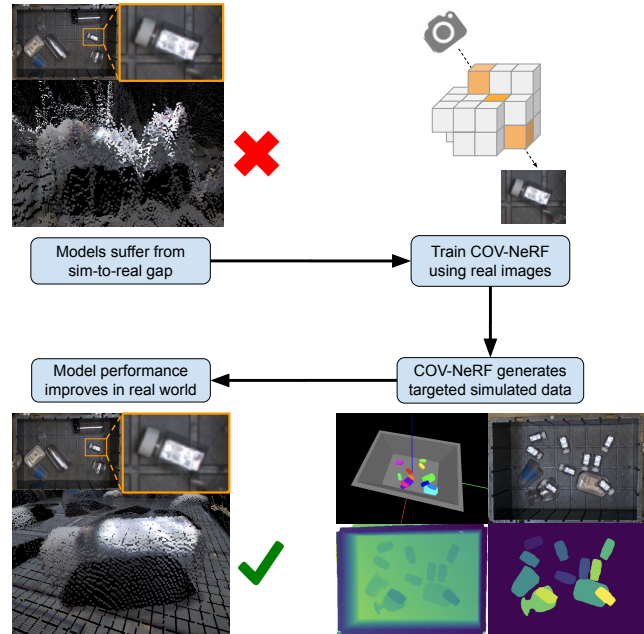


Fig. 1. Our proposed object-centric neural renderer, COV-NeRF, can be used to generate targeted supervision for other models that are brittle to sim-to-real distribution shift. After learning explicit neural representations of real objects, COV-NeRF can compose those representations into photorealistic synthetic scenes and generate many modalities of downstream supervision, including depth maps, segmentation masks, and instance meshes.

In this work, we explore neural rendering methods as a mechanism to close the sim-to-real gap for perception tasks. The seminal NeRF [1] uses real images to build a neural representation of a scene, which can then be rendered from new viewpoints through a ray-marching process based on physical models of light transport. However, these methods typically require extensive test-time optimization (TTO), taking many GPU-hours to reconstruct a single scene. In this work, we introduce Composable Object Volume NeRF, or COV-NeRF, a novel neural renderer that is uniquely suited to remedy sim-to-real mismatch. Unlike prior NeRF methods, COV-NeRF explicitly represents objects while also generalizing across scenes without TTO. COV-NeRF performs real-to-sim learning of structured neural representations that can subsequently be used for novel scene synthesis, rendering photorealistic images along with the corresponding supervision for many perception tasks. Existing methods for domain adaptation simply re-skin simulated scenes [15] [2]; we synthesize new scenes *de novo*, can produce both RGB images and many types of 2D and 3D supervision, and automatically enjoy geometric and semantic consistency across viewpoints and between the rendered images and labels.

Our key contributions are as follows:

- 1) We introduce COV-NeRF, a novel NeRF architecture that both explicitly represents objects and generalizes across scenes. Although these properties have been explored individually in prior work; COV-NeRF is the first to exhibit both simultaneously.
- 2) We demonstrate COV-NeRF’s ability to generate targeted synthetic data based on real-world scenes and objects. Without artificial consistency constraints, we show that COV-NeRF can generate supervision for a variety of perceptual tasks, including depth estimation, object detection, instance segmentation, and shape completion, and that training on COV-NeRF’s generations improve real-world performance.
- 3) Finally, we apply the entire pipeline to a real world bin-picking application. We identify challenging scenarios where state-of-the-art perception models and simulated datasets face a large sim-to-real gap, and show that COV-NeRF’s real-to-sim capabilities can rapidly close the gap to achieve application-level improvement.

II. RELATED WORK

Neural Rendering: NeRF [1] was a breakthrough in physically-plausible differentiable rendering, yielding better visual quality and easier optimization than existing methods for inverse rendering. However, it required 50-100 source images and several GPU-days of TTO, since it had to optimize new neural networks for each scene. More recent methods, such as PixelNeRF [3], MVS-NeRF [4], and NerFormer [5] use more specialized architectures that can be trained simultaneously on many scenes and generalize to new scenes without TTO. Since they learn priors that transfer across scenes, they only need 3-5 source views during inference.

Object-NeRF [7] and Object Scattering Functions (OSF) [6] stayed within the single-scene paradigm, but explored object-centric decompositions that allowed scene editing, where objects can be explicitly added, re-positioned, or scaled. OSF’s edited scenes are the most realistic, but it requires the lighting to be specified parametrically (which can be difficult outside of controlled settings), and an order of magnitude more compute during rendering. Subsequent work [8] [9] adds bells-and-whistles to Object-NeRF; COV-NeRF extends these methods to not require TTO.

In robotics, single-scene NeRFs have been explored for online geometry estimation, but were bottlenecked by the computational cost of TTO [10] [11]. NeRF-Supervised Deep Stereo [12] used single-scene NeRFs to generate labels for stereo depth estimation. This pipeline generated high-quality supervision, but required extensive TTO and many source views per scene. GraspNeRF [13] trained NeRF rendering in simulation as an auxiliary loss for grasp generation, but did not explore or evaluate its rendering quality, let alone its ability to mitigate sim-to-real mismatch. GraspNeRF’s architecture is similar to MVS-NeRF; as such, we expect similar rendering performance, and it does not share COV-NeRF’s object-composable properties.

Sim-to-Real Transfer: Domain randomization has been explored for sim-to-real transfer of both visual and physical tasks. Early visual sim-to-real work [14] focused on diversity rather than realism: these methods achieved promising results on simple tasks and environments, but could not scale to more challenging applications without manual tuning. Simple heuristics like cut-and-paste were explored for segmentation, but produce visually and geometrically implausible scenes, and cannot generate 3D supervision [31] [32].

Domain adaptation methods employed generative models to automatically improve the realism of simulated images. CyCADA [15] trained GANs to translate simulated images into real ones, but required a cycle consistency objective so that the original semantic segmentations could still serve as supervision for the translated images. Subsequent work applied this idea using object detection [2] and Q-values [16]. However, finding such a bijective mapping is an ill-posed task, and its difficulty scales poorly as more modalities are considered (e.g. if we require consistency for depth maps as well as for semantic segmentation). We take a fundamentally different approach: COV-NeRF creates datasets *de novo* that are more realistic and more meaningfully diverse than the adaptations produced by these methods, and achieves multi-view consistency for many modalities for free. We evaluate domain adaptation methods in Section IV-B.

Inspired by recent advances in vision-language alignment, CACTI [17], ROSIE [18], and GenAug [19] used text-conditioned diffusion models [30] to automatically apply augmentations, such as changing backgrounds or adding distractor objects. These methods improved the robustness of end-to-end policies, but ultimately suffer from the same limitations as the GAN-based approaches.

III. GENERALIZABLE NEURAL RENDERING WITH COMPOSABLE OBJECT VOLUMES

A. Preliminaries

NeRF methods use differentiable volumetric rendering to train models for novel view synthesis. Given a collection of source images $\{\mathcal{I}^{(b)}\}_{b=1}^B$ of a scene and their corresponding camera intrinsic/extrinsic matrices, their goal is to render an image \mathcal{I}^* of the scene as seen from a new viewpoint.

Recall that a ray \mathbf{r} is parametrized as $\mathbf{r}(t) = o + t \cdot d, t \geq 0$, where $o, d \in \mathbb{R}^3$ are the ray’s origin and direction. Each pixel $(u, v) \in \mathcal{I}^*$ corresponds to a particular ray $\mathbf{r}_{u,v}(t)$; the intrinsics and extrinsics determine o and d . To render a ray, points $\mathbf{r}(t_i), t_1 < \dots < t_N$ are sampled along it, and the density σ_i and radiance c_i are determined for each, then accumulated to compute the ray’s rendered color $\hat{C}(\mathbf{r})$:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N T_i \alpha_i c_i \quad (1)$$

where $\delta_i = t_{i+1} - t_i$, $\alpha_i = 1 - e^{-\delta_i \sigma_i}$, $T_i = e^{-\sum_{j<i} \delta_j \sigma_j}$. The rendered color of pixel (u, v) is simply $\hat{C}(\mathbf{r}_{u,v}(t))$.

Early NeRFs used fully-connected neural networks to predict σ_i and c_i from the coordinate $\mathbf{r}_{u,v}(t_i)$ and the ray direction d . However, since they lacked direct access to the

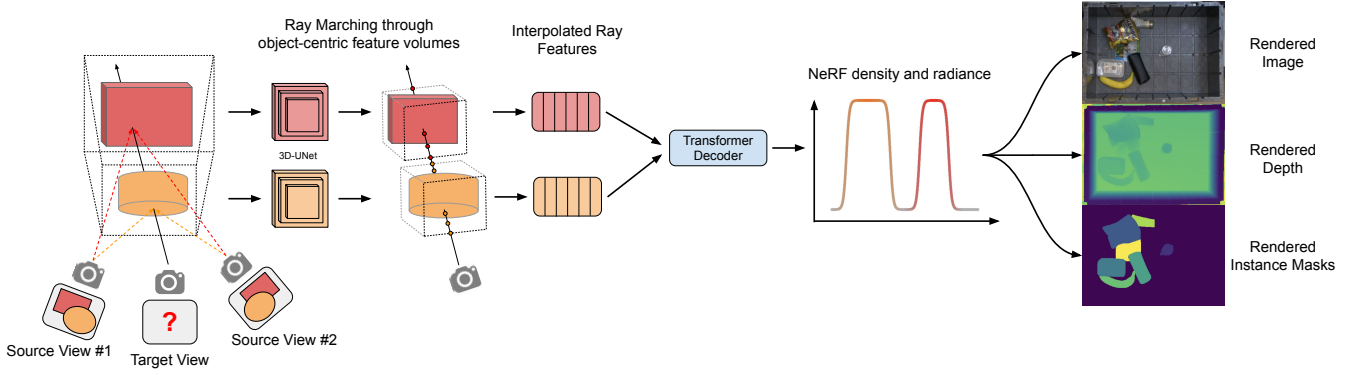


Fig. 2. An overview of COV-NeRF’s object-centric rendering process. Visual features from the source views are projected into a feature volume for each object in the scene. For each pixel to be rendered, features are interpolated along the corresponding ray from each volume that the ray intersects with. A Transformer decodes the interpolated features into the NeRF density and radiance, which are composited into RGB colors, depths and segmentation masks.

source images during inference, they had to distill visual details from the source images into the network weights via optimization, which prevented them from representing multiple scenes. Moreover, since the networks were trained from scratch for each scene, a large $B \approx 100$ was required.

In the scene-generalizable extensions discussed in Section II, the dependence on $\{\mathcal{I}^{(b)}\}$ was made explicit by making these images inputs to the network during rendering. More powerful architectures, such as 3D-CNNs and Transformers, were introduced to better leverage the additional information. Instead of memorizing visual details, the networks learned to robustly aggregate information from the source images, and could work with substantially fewer source views ($B \approx 3$).

B. Composable Object Volumes

For synthetic data generation, we seek a NeRF method that both generalizes across scenes and explicitly represents objects. These properties ensure that we can generate supervision for different perceptual tasks, that the generated supervision is consistent with the rendered images, and that we can scalably generate diverse datasets with many objects in a variety of configurations.

COV-NeRF represents a scene as a collection of objects $\{o_k\}_{k=1}^K$ and a background. Each object is defined by a frustum, which is discretized into a $D \times H \times W$ voxel grid. A learned feature vector is associated with each voxel, resulting in a feature volume $V^{(k)} \in \mathbb{R}^{C \times D \times H \times W}$. The object $o^{(k)}$ is fully specified by its pose $p^{(k)} \in \mathcal{SO}^3$ and volume $V^{(k)}$.

Each $V^{(k)}$ is computed by projecting the RGB values from the source views, resulting in an initial volume of shape $B \times 3 \times D \times H \times W$. We aggregate features across source views via attention over the B dimension, reducing the volume to $C \times D \times H \times W$, and then refine it with a 3D-UNet to produce $V^{(k)}$. MVS-NeRF [4] uses a similar plane-sweep scheme, but constructs one volume for the entire scene, while we construct one for each object.

To render a ray, we gather contributions from each object and the background. For the background, we sample points $r(t_i^{(0)}), t_1^{(0)}, \dots, t_N^{(0)}$ throughout the entire scene, and predict $\sigma_i^{(0)}, c_i^{(0)}$ using a simplified NerFormer [5], which interpolates features from each source view and processes them with a Transformer. For each object $o^{(k)}$, if the ray intersects with

its frustum, we also sample points $r(t_i^{(k)}), t_1^{(k)}, \dots, t_N^{(k)}$. We transform $r(t_i^{(k)})$ and d into the object frame $p^{(k)}$, trilinearly interpolate feature vectors from $V^{(k)}$, and use a Transformer to decode them into $\sigma_i^{(k)}$ and $c_i^{(k)}$ following [5] [26].

To accumulate these values into $\hat{C}(\mathbf{r})$, we sort all the $\{t_i^{(k)}\}_{i=1, k=0}^{i=N, k=K}$ and apply Equation 1. This natively handles occlusions: even if $\sigma_i^{(k)}$ is large, $T_i^{(k)}$ may be small if the ray passes through other objects before hitting object o_k .

Following prior work, the product $\alpha_i T_i$ can be interpreted as the probability that the ray terminates at distance t_i . Then the expected distance $\hat{\tau}(\mathbf{r})$ that the ray travels is:

$$\hat{\tau}(\mathbf{r}) = \sum_{i=1}^N T_i \alpha_i t_i \quad (2)$$

$\hat{\tau}(\mathbf{r})$ is the point-to-point distance from the camera center, but can be trivially converted into a depth value in the camera frame. COV-NeRF renders depth maps using this scheme, considering all objects and the background.

This probabilistic interpretation can also be leveraged to render instance masks. In addition to standard (modal) instance masks, which indicate the pixels where an object is visible, COV-NeRF can also render amodal masks, which include both visible and occluded portions of each object.

Let $M_{u,v}^{(k)} \in [0, 1]$ be the probability that pixel (u, v) belongs to object o_k ’s modal mask, and let $\bar{M}_{u,v}^{(k)}$ be the corresponding probability for its amodal mask. $M_{u,v}^{(k)}$ is simply the probability that $\mathbf{r}_{u,v}$ terminates inside o_k , and $\bar{M}_{u,v}^{(k)}$ is the probability that $\mathbf{r}_{u,v}$ would terminate inside $o^{(k)}$ in the absence of other objects:

$$M_{u,v}^{(k)} = \sum_{i=1}^N T_i^{(k)} \alpha_i^{(k)} \quad (3)$$

$$\bar{M}_{u,v}^{(k)} = \sum_{i=1}^N \bar{T}_i^{(k)} \alpha_i^{(k)}, \text{ where } \bar{T}_i^{(k)} = \exp\left(-\sum_{j<i} \delta_j^{(k)} \sigma_j^{(k)}\right) \quad (4)$$

For more details about COV-NeRF’s architecture and rendering, see [our implementation](#).

C. Training COV-NeRF

Like other scene-generalizable NeRF methods, COV-NeRF learns visual and geometric priors that enable inference when only a few source images are available. To best learn these priors, we train COV-NeRF on a mix of simulated and real data: simulation helps bootstrap its understanding of 3D geometry, and real data exposes it to realistic textures, materials, and lighting. This gives us the best of both worlds: COV-NeRF can leverage dense geometric supervision when available, but is also robust to sim-to-real mismatch, since it can be trained on any real data that it does not initially generalize to. We jointly train the following losses:

- View synthesis: the rendered color (Equation 1) is trained to match the true color using an L2-loss.
- Depth estimation: the rendered depth (Equation 2) is trained to match the true depth using an L1 loss.
- Instance segmentation: the rendered masks (Equation 3) are trained to match the ground-truth masks using a cross-entropy loss.
- Voxel occupancy: we use $V^{(k)}$ to predict the occupancy of each voxel in that object’s feature volume, following [25] [20]. This facilitates novel scene synthesis (see Section III-D) since it lets us automatically pose objects into physically plausible configurations.

In practice, we found that an efficient scheme is to pre-train all of the above losses in simulation, and add real data when it becomes available. We pre-train for 100 epochs on COB-3D-v2 [20], a small but high-quality simulated dataset that contains all of the necessary supervision modalities, which takes about 1 day using 8 NVIDIA RTX A5000s. We used a batch size of 32 scenes and the Adam optimizer with default parameters ($\alpha = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$).

D. Scene Generation with COV-NeRF

Single-scene NeRF methods (see Section II) anecdotally explore scene-editing as a benefit of object-centric rendering. We extend this capability to perform novel scene synthesis, procedurally generating new scenes at scale, for use in a downstream application. When sim-to-real models struggle in the real world, COV-NeRF can synthesize new training data targeting specific real-world scenes and objects.

Given a pre-trained COV-NeRF and collection of real scenes, we first perform real-to-sim finetuning using the captured images and any additional supervision that may be available (such as instance mask annotations). Although this step is not strictly required, the ability to use weak 2D supervision as learning signal for the underlying 3D semantics and geometry (for which direct real-world supervision usually cannot be obtained) is a strength unique to COV-NeRF, and is only possible because of its object- and ray-centric structure. Next, we extract feature volumes and meshes for objects from the real scenes. The meshes are extracted from the voxel occupancy predictions via Marching Cubes. To compose a new scene, we sample objects, pose them in geometrically realistic configurations using their meshes and a physics simulator (we use Mujoco [27]), and then render the scene using COV-NeRF. The process is described in Algorithm 1.

Algorithm 1: Scene Generation with COV-NeRF

```

Input: Desired number of scenes  $N$ 
         Feature volumes  $\mathcal{V} = \{V^{(1)} \dots, V^{(K)}\}$ 
         Background source images  $\{\mathcal{I}^{(1)}, \dots, \mathcal{I}^{(B)}\}$ 
for  $i = 1, \dots, N$  do
    Sample camera viewpoints
    Sample a number of objects  $K^{(i)}$ 
    Initialize physics simulator (e.g. Mujoco)
    for  $k = 1, \dots, K^{(i)}$  do
        Sample object  $V^{(k)} \sim \mathcal{V}$  and corresponding mesh
        Sample initial pose  $p_0^{(k)}$ 
        Add mesh to simulator at pose  $p_0^{(k)}$ 
    end
    Advance physics for  $T$  timesteps (e.g. until all objects settle)
    Render images and supervision with  $V^{(k)}, p_T^{(k)}, k = 1, \dots, K^{(i)}$  and  $\{\mathcal{I}^{(1)}, \dots, \mathcal{I}^{(B)}\}$ 
end

```

IV. EXPERIMENTS

We conducted experiments in both simulation and the real world to answer the following questions:

- 1) How does COV-NeRF compare to other NeRF methods, in terms of the visual quality of its renderings?
- 2) How effective is synthetic data generated by COV-NeRF for training perception models relevant to robotic applications?
- 3) How effective is COV-NeRF at reducing the sim-to-real gap in challenging scenarios?

A. View Synthesis

To evaluate COV-NeRF’s rendering capabilities, we compare it to the following methods:

- MVS-NeRF [4] is a state-of-the-art scene-generalizable NeRF method. It shares architectural similarities to COV-NeRF, but is not object-centric.
- Object-NeRF [7] is an object-centric renderer like COV-NeRF, but must be retrained for each scene. It requires many sources views and mask supervision in each.

In Table I, we evaluate MVS-NeRF and COV-NeRF on the COB-3D-v2 validation set (~ 600 scenes). Object-NeRF is excluded due to computational constraints. In Table II, we evaluate all methods on scenes from the real-world environment from Section IV-B. In Figure 3, we show qualitative examples from these scenes. COV-NeRF strictly outperforms MVS-NeRF, and, without TTO, matches or exceeds the performance of single-scene methods like Object-NeRF.

TABLE I
SCENE-GENERALIZABLE VIEW SYNTHESIS ON COB-3D-V2

Method	TTO?	Object-Centric?	PSNR (\uparrow)	SSIM (\uparrow)	LPIPS (\downarrow)
MVS-NeRF	N	N	25.03	0.867	0.102
COV-NeRF	N	Y	28.32	0.915	0.082

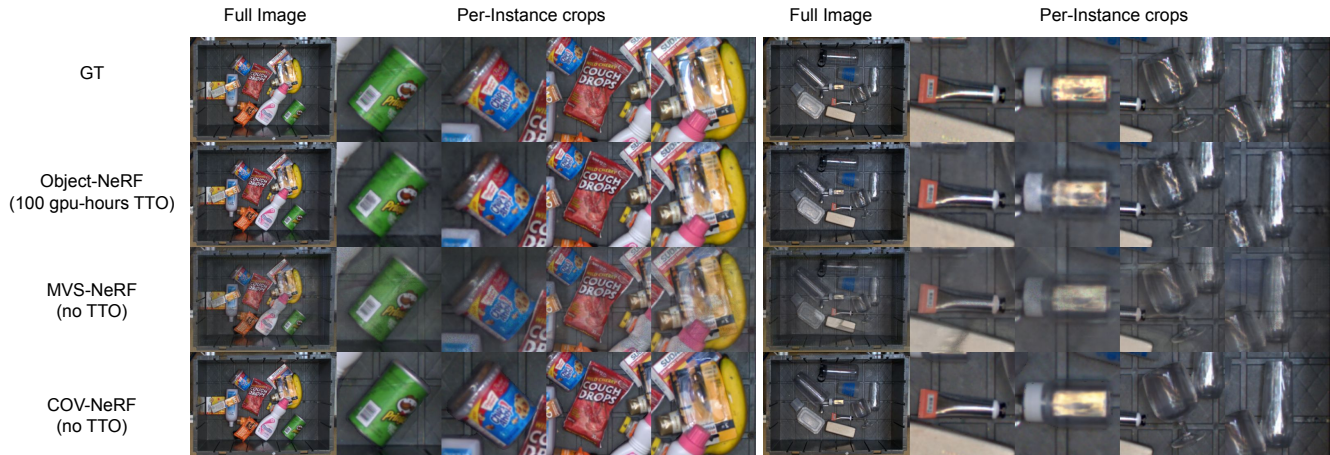


Fig. 3. Qualitative view synthesis results on two real scenes. For each method, an image is rendered from a novel viewpoint using 4 source views (not pictured). The ground-truth image from the novel viewpoint is shown in the top row. COV-NeRF matches the performance of object-centric methods that require expensive, per-scene TTO, and outperforms other scene-generalizable methods.

TABLE II
VIEW SYNTHESIS, REAL

Method	TTO?	Object-Centric?	PSNR (\uparrow)	SSIM (\uparrow)	LPIPS (\downarrow)
Object-NeRF	Y	Y	23.01	0.841	0.118
MVS-NeRF	N	N	18.97	0.830	0.203
COV-NeRF	N	Y	25.62	0.905	0.089

B. Sim-to-real Perception

In this section, we study COV-NeRF’s effectiveness at improving visual sim-to-real transfer. Using a real-world bin picking system, we construct challenging configurations that exhibit a sim-to-real gap and evaluate COV-NeRF’s effectiveness at reducing the gap. Our system consists of an ABB 1200, a 6-cup suction gripper, and 6 RGB cameras mounted over a bin. The robot must grasp objects from the bin one at a time, and transport them to an adjacent bin.

We construct a state-of-the-art bin picking system, using the following components as a representative sample of fundamental perception capabilities for robotic applications:

- Instance segmentation: we modify a SOTA method, MaskDINO [23], to predict both modal and amodal masks. We train it on COB-3D-v2, which has similar scene composition to our real environment.
- Depth estimation: we train a SOTA model for multi-view stereo, MVS-Former [24], on COB-3D-v2.
- Grasping: we use a fully-convolutional grasp-quality CNN (FC-GQCNN) [22] in the style of DexNet 3.0 [21]. It is trained in simulation to place suction cups on flat surfaces and near the object’s center of mass. For each unoccluded object in the scene (as predicted by MaskDINO), we sample grasps based on a crop of the depth map predicted by MVS-Former.

We consider a few different scenarios:

- 1) Mixed-Clutter: generic household objects are arranged chaotically inside the bin. The clutter is generally challenging for both segmentation and depth; it can be especially difficult to reason about occlusions.
- 2) Hard-Specular: we curate a set of challenging transparent and reflective objects. These non-Lambertian

surfaces cannot be accurately sensed by standard depth cameras, necessitating the use of learned methods like MVS-Former.

For each scenario, we first evaluate the baseline system. As expected, there is substantial room for improvement because it has not seen any real data, let alone these particular objects and environmental conditions. We then benchmark the following strategies for reducing the sim-to-real gap:

- COV-NeRF: We generate a synthetic dataset (following Section III-D) and use it to finetune MaskDINO and MVS-Former. Although COV-NeRF generalizes well for view synthesis (as evidenced by Figure 3), we found that even tiny amounts of mask supervision vastly improve the synthetic dataset, especially the sharpness of the rendered masks. We finetune COV-NeRF using 100 real scenes encountered by the baselines system, of which 5 have mask supervision, and use the extracted objects to generate 1000 synthetic scenes.
- CyCADA [15]: This well-established domain adaptation method uses GANs to translate images between sim and real, with cycle consistency in both RGB and semantic segmentation. Deriving semantic segmentation from instance masks using three classes,

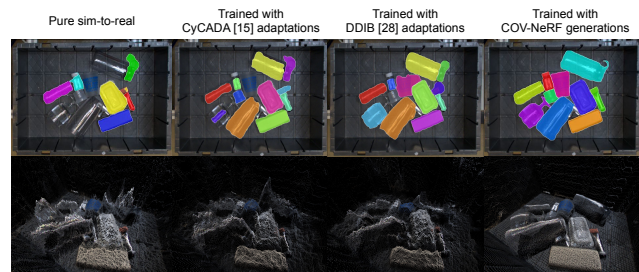


Fig. 4. Sample instance segmentation predictions from MaskDINO (top row) and stereo depth predictions from MVS-Former (bottom row) resulting from the sim-to-real methods evaluated in Table III. COV-NeRF enables substantial improvement in both modalities.

TABLE III
SIM-TO-REAL IMPROVEMENT

Method	Real Supervision	Grasp Success Rate (\uparrow)		Mask AP, modal / amodal (\uparrow)	
		Mixed-Clutter	Hard-Specular	Mixed-Clutter	Hard-Specular
Pure sim-to-real	None	0.700 \pm 0.028	0.463 \pm 0.035	28.5 / 27.3	13.8 / 12.4
Finetune segm	100 scenes, segm	0.742 \pm 0.035	0.567 \pm 0.036	59.4 / 57.3	66.5 / 65.3
CyCADA [15]	100 scenes, rgb	0.725 \pm 0.035	0.453 \pm 0.037	31.2 / 29.2	28.1 / 28.0
DDIB [28]	100 scenes, segm	0.727 \pm 0.031	0.598 \pm 0.037	36.1 / 35.1	31.7 / 31.3
COV-NeRF (ours)	100 scenes (5 segm / 95 rgb only)	0.929 \pm 0.021	0.807 \pm 0.022	62.7 / 61.8	72.9 / 72.4

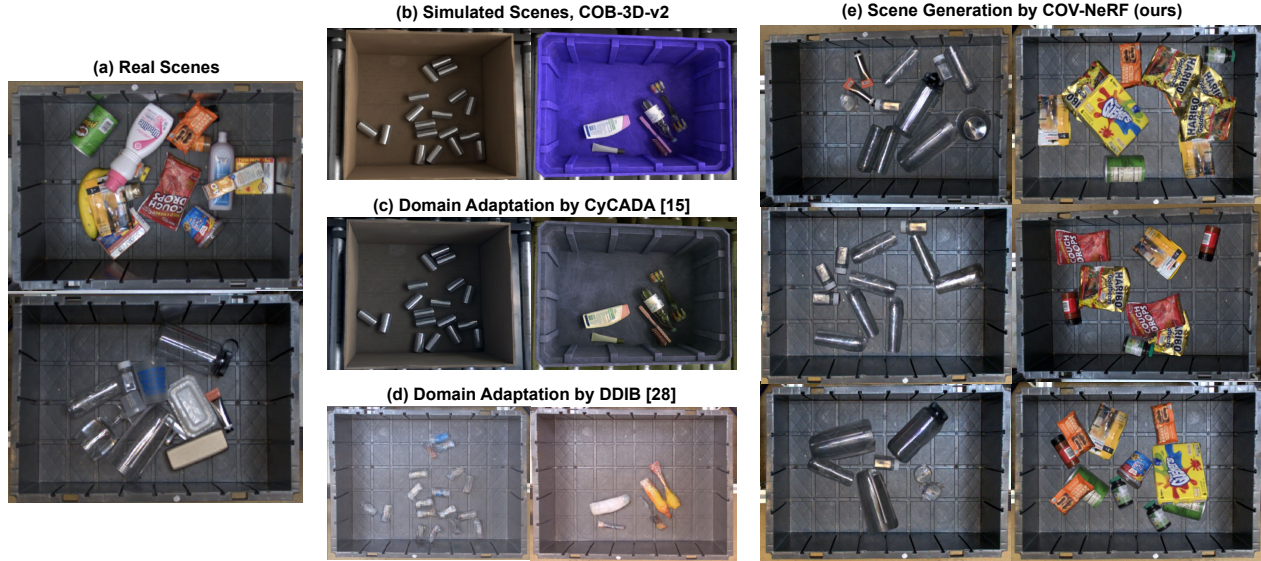


Fig. 5. (a) Representative real images from the Mixed-Clutter (top) and Hard-Specular (bottom) scenarios. (b) Sample simulated scenes from COB-3D-v2. (c) CyCADA [15] adapts the scenes from (b) to more closely resemble samples from the real world, but its cycle consistency objectives result in only a mild re-styling of the sim scenes. (d) DDIB [28] produces more visually realistic adaptations of (b), but violates the original scene semantics. (e) Instead of adapting simulated scenes, COV-NeRF composes new scenes using explicit object representations extracted from (a).

$\{\text{background, object, edge}\}$, we train CyCADA using COB-3D-v2 and 100 real scenes from each scenario. We then use the trained GANs to augment the training of MaskDINO and MVS-Former: for each simulated scene in COB-3D-v2, we use the original labels as supervision for the adapted images produced by CyCADA.

- Dual Diffusion Implicit Bridges [28]: DDIB performs image-to-image translation using diffusion models, which are the modern method of choice for image generation. Inspired by conditional diffusion models (such as StableDiffusion’s img2img and depth2img modes [29]), we extend DDIB to be mask-conditioned, and train it to translate between COB-3D-v2 and our real scenarios. We then use the same procedure as for CyCADA to train MaskDINO and MVS-Former.
- Segmentation Finetuning: As a baseline, we finetune MaskDINO on real scenes using mask supervision. Note that this does not affect MVS-Former.

In Figure 5, we visualize the outputs of each method, and in Table III, we quantitatively evaluate grasp success and instance segmentation (mask AP). Depth estimation can only be evaluated indirectly through grasp success, since we cannot obtain ground-truth in the real world (especially for Hard-Specular), but we study qualitative examples in Figure 4. With as little as 5 scenes of mask supervision, COV-NeRF enables substantial improvement in both grasp success

and mask AP. CyCADA has almost no effect on the real system: the cycle consistency objective and the challenges of GAN optimization prevent it from substantially altering the simulated images. DDIB’s adaptations have more realistic low-level textural details, but are still vastly less effective than the scenes synthesized by COV-NeRF.

V. CONCLUSION

We presented COV-NeRF, an object-composable and scene-generalizable neural renderer that can the close sim-to-real gap for a variety of perception modalities, including ones where direct supervision cannot be obtained in the real world. After validating COV-NeRF against existing NeRF methods, we explored its effectiveness at combating sim-to-real mismatch in challenging bin-picking scenarios. We showed that COV-NeRF can generate targeted synthetic data that is effective at improving state-of-the-art perception methods, translating to significant improvements in end-to-end application performance.

Despite its demonstrated results, COV-NeRF’s rendering model suffers from a few limitations. It does not account for higher-order visual effects like reflections or variations in scene lighting, and it must imagine the appearance of the occluded portions of objects. We hope to address this in future work, as it would further enhance the diversity and realism of the scenes that COV-NeRF can generate.

REFERENCES

- [1] Mildenhall, Ben, et al. "NeRF: Representing scenes as neural radiance fields for view synthesis." *European Conference on Computer Vision (ECCV)*, 2020.
- [2] Ho, Daniel, et al. "RetinaGAN: An object-aware approach to sim-to-real transfer." *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [3] Yu, Alex, et al. "PixelNeRF: Neural radiance fields from one or few images." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [4] Chen, Anpei, et al. "MvsNeRF: Fast generalizable radiance field reconstruction from multi-view stereo." *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [5] Reizenstein, Jeremy, et al. "Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction." *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [6] Yu, Hong-Xing, et al. "Learning object-centric neural scattering functions for free-viewpoint relighting and scene composition." *Transactions on Machine Learning Research (TMLR)*, 2023.
- [7] Yang, Bangbang, et al. "Learning object-compositional neural radiance field for editable scene rendering." *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [8] Kundu, Abhijit, et al. "Panoptic neural fields: A semantic object-aware neural scene representation." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [9] Siddiqui, Yawar, et al. "Panoptic lifting for 3d scene understanding with neural fields." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [10] Ichnowski, Jeffrey, et al. "Dex-NeRF: Using a neural radiance field to grasp transparent objects." *Conference on Robot Learning (CoRL)*, 2021.
- [11] Kerr, Justin, et al. "Evo-NeRF: Evolving nerf for sequential robot grasping of transparent objects." *Conference on Robot Learning (CoRL)*, 2022.
- [12] Tosi, Fabio, et al. "NeRF-Supervised Deep Stereo." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [13] Dai, Qiyu, et al. "GraspNeRF: multiview-based 6-DoF grasp detection for transparent and specular objects using generalizable NeRF." *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [14] Tobin, Josh, et al. "Domain randomization for transferring deep neural networks from simulation to the real world." *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [15] Hoffman, Judy, et al. "CyCADA: Cycle-consistent Adversarial Domain Adaptation." *International Conference on Machine Learning (ICML)*, 2018.
- [16] Rao, Kanishka, et al. "RL-CycleGAN: Reinforcement learning aware simulation-to-real." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [17] Mandi, Zhao, et al. "CACTI: A framework for scalable multi-task multi-scene visual imitation learning." *arXiv preprint arXiv:2212.05711*, 2022.
- [18] Yu, Tianhe, et al. "Scaling robot learning with semantically imagined experience." *arXiv preprint arXiv:2302.11550*, 2023.
- [19] Chen, Zoey, et al. "GenAUG: Retargeting behaviors to unseen situations via generative augmentation." *Robotics: Science and Systems (RSS)*, 2023.
- [20] Mishra, Nikhil, et al. "Convolutional Occupancy Models for Dense Packing of Complex, Novel Objects." *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.
- [21] Mahler, Jeffrey, et al. "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning." *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [22] Satish, Vishal, Jeffrey Mahler, and Ken Goldberg. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.
- [23] Li, Feng, et al. "Mask dino: Towards a unified transformer-based framework for object detection and segmentation." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [24] Cao, Chenjie, Xinlin Ren, and Yanwei Fu. "MVSFormer: Multi-View Stereo by Learning Robust Image Features and Temperature-based Depth." *Transactions on Machine Learning Research (TMLR)*, 2022.
- [25] Gkioxari, Georgia, Jitendra Malik, and Justin Johnson. "Mesh R-CNN." *IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [26] Liu, Yuan, et al. "Neural rays for occlusion-aware image-based rendering." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [27] Todorov, Emanuel and Erez, Tom and Tassa, Yuval. "MuJoCo: A physics engine for model-based control." *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [28] Su, Xuan and Song, Jiaming and Meng, Chenlin and Ermon, Stefano. "Dual Diffusion Implicit Bridges for Image-to-Image Translation." *International Conference on Learning Representations (ICLR)*, 2023.
- [29] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [30] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [31] Ghiasi, Golnaz, et al. "Simple copy-paste is a strong data augmentation method for instance segmentation." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [32] Sieb, Maximilian, and Katerina Fragkiadaki. "Data dreaming for object detection: Learning object-centric state representations for visual imitation." *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2018.