Unsupervised Continuous Object Representation Networks for Novel View Synthesis

Nicolai Häni **Selim Engin** **Jun-Jee Chao** **Volkan Isler**
University of Minnesota
{haeni001, engin003, chao0107, isler}@umn.edu

Abstract

Novel View Synthesis (NVS) is concerned with the generation of novel views of a scene from one or more source images. NVS requires explicit reasoning about 3D object structure and unseen parts of the scene. As a result, current approaches rely on supervised training with either 3D models or multiple target images. We present Unsupervised Continuous Object Representation Networks (UniCORN), which encode the geometry and appearance of a 3D scene using a neural 3D representation. Our model is trained with only two source images per object, requiring no ground truth 3D models or target view supervision. Despite being unsupervised, UniCORN achieves comparable results to the state-of-the-art on challenging tasks, including novel view synthesis and single-view 3D reconstruction.

1 Introduction

In 1971, Shephard and Metzler [43] introduced the concept of mental rotation, the ability to rotate 3D objects mentally, and link the model to its projection, to the cognitive sciences. Novel View Synthesis (NVS) research seeks to replicate this capability in machines by generating images of a scene or an object from previously unseen viewpoints. This is a challenging problem, as it requires understanding the 3D scene structure or image semantics, and the ability to project the "mental" representation into a target viewpoint.

A common approach for NVS is to use a large collection of views to reconstruct a 3D scene [9, 42]. Recent methods have made progress in learning 3D object representations, such as voxel grids [59, 45, 51, 31, 30], point clouds [1, 60, 55], or meshes [53, 10, 47, 54]. However, the discrete nature of these representations limit the achievable resolution and induce significant memory overhead. Continuous representations [34, 24, 41, 46, 57, 6, 23, 26] address these challenges. However, proposed methods require either 3D ground truth or multi-view supervision, limiting the applicability of these approaches to domains where data is available.

We introduce Unsupervised Continuous Object Representation Networks (UniCORNs), a neural object representation that can be learned from as few as two images per object. The conditional formulation of UniCORN, combined with a differentiable neural renderer, enforces multi-view consistency of the learned 3D scenes, and naturally generalizes across different object shapes. The key idea of UniCORN is to use transformation chains and 3D feature consistency as self-supervision requiring $50\times$ fewer data during training than the current state-of-the-art models. We build on recent advances in continuous function representations, especially [34, 46, 26]. Instead of using a global object descriptor, UniCORN combines local and global features to learn detailed 3D structures. Furthermore, UniCORN uses an auto-encoder in contrast to the previously proposed auto-decoder approaches [34, 46]. Using an auto-encoder avoids latent code optimization at test time, which improves inference speed drastically.

In summary, our contributions are:

Preprint. Under review.

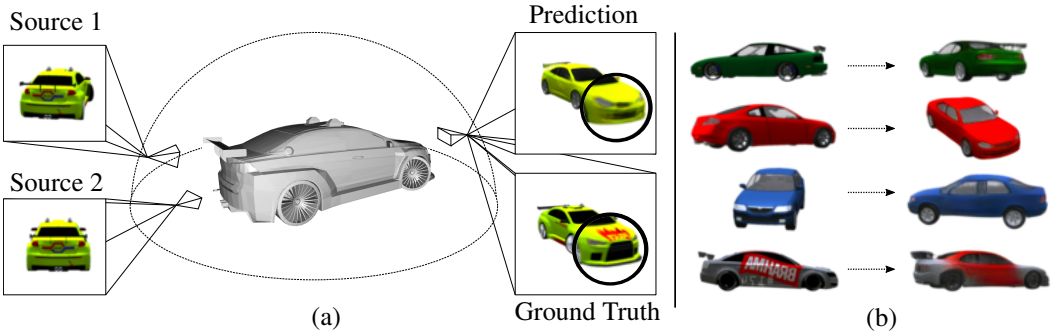


Figure 1: Our model learns to synthesize novel views using only two source images per object during training (a). For this instance, even though both source images are from the back of the car, our model can reconstruct unseen areas with a reasonable level of detail. During inference (b), our model predicts novel views from a single input image. It can accommodate drastically different source and target poses.

- We present UniCORN for view synthesis based on a novel representation that captures the scene's appearance and geometry at arbitrary spatial resolution. UniCORN is end-to-end trainable and uses only two images per object during training time, without any 3D space or 2D target view supervision.
- Despite being unsupervised, UniCORN performs competitively (up to within 2% of the best score) against approaches that use dozens of images per object and direct supervision on the target views.
- We present several applications of our method, including single-view 3D reconstruction and novel view synthesis from out-of-domain samples.

2 Related Work

Our goal is to learn a generative model for novel view synthesis, that extracts 3D scene content, without access to the 2D or 3D ground truth. As such, our work lies at the intersection of novel view synthesis, 3D object reconstruction, and generative modeling. In the following, we review related work.

Novel view synthesis. Novel view synthesis from single or few images is ill-posed, as infinitely many different 3D shapes may render the same images. Early work on novel view synthesis has sought to represent 3D knowledge implicitly; by directly regressing pixels in the target image [49, 48, 58], weakly disentangling view pose and appearance [65, 61, 20] or by learning appearance flow [63, 33, 48, 5]. Other prior work proposed to apply the view transformations in latent space [56] or learn a complete latent space [50] from which to sample. Generative Query Networks [8, 21, 29], a probabilistic reasoning framework, embeds the 3D structure of a scene in a neural network. Implicit 3D representations do not allow extraction of the scene's 3D structure. We propose a continuous object representation that models 3D scenes explicitly and generates new views with a state-of-the-art neural renderer.

3D scene representations. Novel view synthesis requires reasoning about the 3D world. If a large number of images from distant viewpoints are available, 3D scene content can be inferred and used to generate novel views with conventional methods. Traditionally, this was achieved via multi-view geometry [7, 9, 42]. If only a single or a few images are given, we can learn a representative 3D object representation from incomplete information. Different 3D representations exist. Discrete representations include: one dimensional latent vectors embeddings [8], voxels [59, 45, 51, 31, 30], meshes [53, 10, 47, 54] or point clouds [1, 60, 55]. Methods based on continuous function representations represent 3D scenes as learned binary classifiers [6, 23, 26] or continuous signed distance functions [34, 24, 41, 46, 57]. While these techniques are successful at modeling 3D geometry, they often require 3D supervision. When combined with a neural renderer, some approaches are supervised with 2D target images instead, relying on large image collections for training. Our proposed method encapsulates scene geometry and appearance from only two source

images per object and can be trained end-to-end via a learned neural rendering function through self supervision.

Generative models. Our work builds on recent advances in using deep neural networks to generate high-quality images. Generative Adversarial Networks (GAN) [11, 37, 3] and its conditional variants [28, 14, 64] generate these images from either a random latent code or in the conditional case, directly from input images. Some approaches synthesize new views by embedding the view transformations in the networks [13, 15, 59]. Another approach is to treat novel view synthesis as an inverse graphics problem [20, 61, 19, 52, 44, 62]. In contrast to recent 3D scene representation learning methods [46, 34] that learn by optimizing latent codes at train and inference time, we show that the conditional GAN formulation allows novel view synthesis at test time from a single input image, without additional overhead.

3 Method

In this section, we introduce UniCORN (Fig. 2) and describe how we overcome the main challenge of representing a 3D scene without direct 2D or 3D supervision. To learn 3D structure, we project the input images into a latent feature space from which we learn to generate a continuous 3D representation of the object. This 3D representation is projected to a new view using a neural renderer, and a refinement network synthesizes the target image. We frame the entire end-to-end system (Fig. 2) as a GAN and use cyclic consistency to supervise the model with only the input images.

For each object, UniCORN takes as input a pair of source images I_1, I_2, and their camera poses T_1, T_2. Our goal is to learn a continuous scene representation θ that explicitly represents the 3D scene-structure and allows the synthesis of novel views I_G at arbitrary camera positions T_G without direct supervision in either the 3D or 2D target domain.

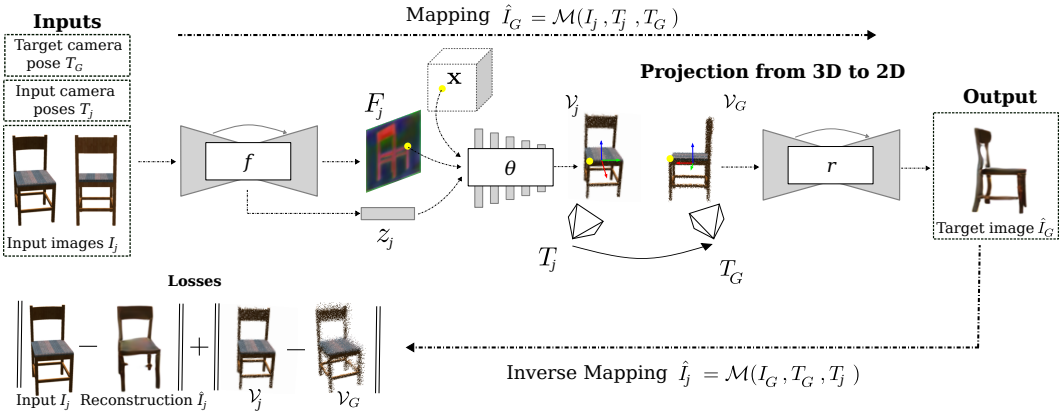


Figure 2: **Our proposed end-to-end model.** The system takes as input two source images I_j, together with their respective camera poses $T_j, j = 1, 2$, and a target pose T_G. The *feature predictor* function f learns to extract a global object descriptor z_j and a set of local features F_j (visualizing projected features with PCA). Local and global features along with a sampled 3D point $\mathbf{x}$ are input to the *continuous representation network* θ to generate 3D features. These 3D features are rendered using goal transformation T_G and passed through a *refinement network* r to generate the target prediction $\hat{I}_G$. We supervise the model by computing the inverse mapping $\hat{I}_j = \mathcal{M}(\hat{I}_G, T_G, T_j)$ and enforce source image reconstruction using a variety of photometric and discriminator losses. For the 3D visualizations the RGB colors are directly taken from images, for clarity.

3.1 Spatial and global object feature network

An asymmetric Fully Convolutional Network (FCN) [25] f maps the raw input images into higher dimensional global and local features. The global features represent high-level object semantics, whereas the local features express pixel-level information and high-frequency details. The design of the feature network follows standard architectures adapted for the task.

Global feature network. We use a pretrained ResNet-18 [12] network to extract a 128-dimensional global feature z_j.

Local feature network. The local feature network builds on the UNet [39] architecture. It takes the last two-dimensional feature map of the global feature extractor as input and uses four upsampling and skip-connection layers to extract a 64-dimensional feature F_j for each pixel of the input image.

3.2 Continuous scene representation

Our continuous scene representation network θ maps local and global features at a point $\mathbf{x} \in \mathbb{R}^3$ to a learned high dimensional 3D object representation. We sample 3D points $\mathbf{x}$ uniformly at random and extract local features by projecting the points to the feature map using the known camera pose. We follow a perspective pinhole camera model that is fully specified by its extrinsic $E = [R, t] \in \mathbb{R}^{3\times4}$ and intrinsic $K \in \mathbb{R}^{3\times3}$ matrices. The extrinsic camera matrix contains rotation matrix $R \in \mathbb{R}^{3\times3}$ and translation vector $t \in \mathbb{R}^3$. Given a 3D coordinate $\mathbf{x}$, the projection from world space to the camera is given by:

$$\mathbf{u} = [u \quad v \quad 1]^\top = K(R\,\mathbf{x} + t) \tag{1}$$

where u and v are the pixel coordinates in the image plane. We extract local features at location (u, v) using bilinear sampling. The continuous representation network θ takes concatenated features $\mathbf{v} \in \mathbb{R}^m$ (containing local features $F_j(u, v)$, global features z_j, and the sampled 3D point $\mathbf{x}$) as input and maps them to a higher dimensional feature $\mathbf{v}' \in \mathbb{R}^n$ at the given spatial coordinate $\theta : \mathbb{R}^m \to \mathbb{R}^n, \mathbf{v} \mapsto \mathbf{v}' = \theta(\mathbf{v})$. In our experiments, we predict a 64-dimensional feature for each spatial location. To increase spatial consistency, we also predict occupancy probabilities for each 3D point, which we render as binary masks. This continuous representation densely models the scene properties and can, in theory, be sampled with arbitrary resolution. In our work, we use a multi-layer perceptron (MLP) to approximate θ.

In contrast to most recent work on representing scenes as continuous functions [34, 46], which only use a global object descriptor to represent a scene, we use global and local features. In this, we compare to the recent work by Xu et al. [57] on single-view 3D model reconstruction, which has shown improved performance on modeling fine details using such an approach.

3.3 Point encoding

Instead of operating on xyz-coordinate inputs directly, we found that learning a higher dimensional embedding of the 3D space improves the overall performance. This is consistent with recent work [38, 27], which show that neural networks are biased towards learning low frequency functions. A higher dimensional encoding can increase the networks capability to correctly model high-frequency details. They also showed, that using deterministic trigonometric functions for the point encoding γ achieves similar performance as when γ is parametrized as an MLP. Following Mildenhall et al. [27] we parametrize γ as non-learned composition of trigonometric functions:

$$\gamma(\mathbf{x}) = \big(\sin(2^0\pi\mathbf{x}), \cos(2^0\pi\mathbf{x}), ..., \sin(2^{L-1}\pi\mathbf{x}), \cos(2^{L-1}\pi\mathbf{x})\big). \tag{2}$$

. In our experiments, we set $L = 10$ and apply γ to each point coordinate individually.

3.4 Neural renderer and refinement network

Given a neural scene representation θ and a targeted view pose T_G, we generate a rendering of the scene representation at the target viewpoint. We sample coordinates in a 3D bounding volume uniformly at random for k points $\mathbf{x}_1, ..., \mathbf{x}_k \in \mathbb{R}^3$ and evaluate the scene representation at these positions. The points are projected to the image plane at the target camera's transformation matrix T_G using a neural renderer based on [55]. The renderer uses soft-rasterization [22] and α-compositing to generate a projected feature map F'_j.

Even if the neural renderer projects features accurately, regions not seen in the input view will be empty in the target image. The refinement network *r* infers these missing features. To build the refinement network, we use a UNet [39] architecture with four down/upsampling blocks and skip connections and spectral normalization [38] following each convolution layer to regularize training.

3.5 Unsupervised learning

To discover a meaningful 3D scene representation without 3D or 2D target image supervision, we assume without loss of generality that for an object, there exists a unique 3D object representation in a canonical view frame. If we can discover such a representation, we can render arbitrary views. Our key insight is that we can use cyclic consistency in the form of transformation chains to learn this representation from only two source images in a self-supervised fashion.

Cyclic consistency. Given two source images I_1 and I_2 we define three transformation chains to regularize the 3D representation. The first two are transforming one source image to the target view pose and back to the second source image: $\hat{I}_2 = I_2 \leftarrow \hat{I}_G \leftarrow I_1; \hat{I}_1 = I_1 \leftarrow \hat{I}_G \leftarrow I_2$. To encourage consistency between the two source views, we also transform source image one into source image two and vice versa: $\hat{I}_1 \leftarrow I_2$ and $\hat{I}_2 \leftarrow I_1$. We use a combination of L_1 and perceptual losses [16] between the input source images and the source reconstructions to supervise our model with reconstruction loss $\mathcal{L}_{\text{rec}} = |I_j - \hat{I}_j|_1 + |I_j - \hat{I}_j|_{\text{vgg}}$.

3D feature consistency. Since we assume the object to be in a canonical view frame, we enforce 3D feature consistency among transformations. Our scene representation network θ should, therefore, learn a view independent, continuous object representation. Our 3D feature representation loss $\mathcal{L}_{\text{3d}}$ uses L_1 loss between the different 3D feature representations of the transformation chains as $\mathcal{L}_{\text{3d}} = \sum_{i \neq j} |\mathcal{V}_j - \mathcal{V}_i|_1$, where $\mathcal{V}$ denotes the set of features $\{\mathbf{v}'_i\}_{i=1}^k$ for each point $\mathbf{x}_i$.

Other losses. To encourage the network to generate spatially meaningful features in 3D space, we also use our object representation network θ to predict occupancy for the sampled 3D points $\mathbf{x}$. We supervise occupancy prediction using binary cross-entropy between the source masks (we segment the input image instead of using ground truth mask) and the predicted masks. Finally, to encourage the generator to synthesize realistic images, we use a GAN loss $\mathcal{L}_{\text{gan}}$ with a patch discriminator [14]. Our final objective function is:

$$\mathcal{L} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{3d}}\mathcal{L}_{\text{3d}} + \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{gan}}\mathcal{L}_{\text{gan}} \tag{3}$$

where the λ parameters are the respective weight of the loss terms.

Training details. The models are trained with the Adam optimizer, learning rate of 0.0002, and momentum parameters $(0, 0.99)$. Empirically, we found $\lambda_{\text{rec}} = 10, \lambda_{\text{3d}} = 1, \lambda_{\text{bce}} = 1,$ and $\lambda_{\text{gan}} = 0.1$ to lead to good convergence. The models are trained for 25 epochs (chair) and 5 epochs (car). We implemented our models in PyTorch [35]; they take 1-2 days to train on 4 Tesla V100 GPUs. For architectural details please see the supplementary material. Code, data, and model files will be made available upon publication of this work.

4 Experiments

We evaluate our approach on the task of novel view synthesis and validate our architecture design through ablations. We additionally evaluate our system on potential applications, namely single-view 3D reconstruction and out of distribution view synthesis on real data. For additional results, we refer the reader to the supplementary document.

4.1 Experimental setup

Dataset. For novel view synthesis, we perform experiments on the ShapeNet v2.0 [4] dataset. Similar to previous work, we use the car and chair categories and extract train/test data split from standard benchmarks [33, 48]. Our dataset contains 108 RGB images per object with a resolution of 128×128 pixels. The camera poses are sampled from a viewing hemisphere with equally spaced azimuth (ranging between $0^\circ - 360^\circ$) and elevation ($0^\circ - 20^\circ$) angles. During training, our model uses only two of the 108 images per object as input.

Evaluation metrics. We evaluate our method using two standard error metrics for image generation tasks: $a)$ L_1 distance and $b)$ structural similarity (SSIM) between the predicted and ground truth images. The L_1 distance (lower is better) measures the performance of the color estimation, whereas the SSIM index (higher is better) measures structural properties, such as edge consistency.

Baselines. As our method is the first unsupervised continuous object representation, we compare UniCORN on novel view synthesis against five recent methods that use 2D target view supervision.

In our first experiment, we compare our approach against [48, 5, 58, 31], which share a common evaluation protocol. At inference time, the models are presented with a single source image and have to generate a fixed target view. To compare our method against SRN [46], we use a single image to generate the 3D object representation. From this representation, we synthesize all 108 views and compare them against the ground truth. Note that all these baseline methods have access to the ground truth target images, while our method does not. For additional details on the baseline methods, please see the supplementary document.

4.2 Comparison with other methods

Results on single image view synthesis. We use 20,000 randomly generated test pairs for the standardized view synthesis evaluation. The task is to transform a single source-view into a target camera view. Table 1 and Fig. 3 show results on the ShapeNet v2.0 cars and chairs dataset. Despite being unsupervised, our method performs competitively (up to within 2% of the best score) against the supervised approaches, demonstrating UniCORNs ability to generate meaningful object representations from a fraction of the data and without direct supervision. The appearance flow-based methods fail if the viewpoint transformations are large (top row). Our model qualitatively preserves fine details and generates meaningful results for missing parts of the object without any supervision.

Table 1: **Quantitative novel view synthesis results.** We report mean and standard deviation of the L_1 loss (lower is better) and the structural similarity (SSIM) index (higher is better) for several supervised methods and our unsupervised one. Our model achieves competitive results, using $50\times$ less data and only two input images for self-supervision.

Methods	Car		Chair	
	$L_1(\downarrow)$	SSIM $(\uparrow)$	$L_1(\downarrow)$	SSIM $(\uparrow)$
M2NV [48]	0.0692 / 0.03	0.7761 / 0.06	0.0573 / 0.03	0.7709 / 0.07
CVS [5]	0.0456 / 0.02	0.8251 / 0.05	0.0618 / 0.03	0.7766 / 0.07
VIGAN [58]	**0.0261** / 0.01	**0.8891** / 0.04	0.0605 / 0.02	0.7828 / 0.05
TBN [31]	0.0288 / 0.01	0.8823 / 0.05	**0.0459** / 0.02	**0.8298** / 0.06
UniCORN	0.0310 / 0.01	0.8716 / 0.04	0.0645 / 0.02	0.7627 / 0.04

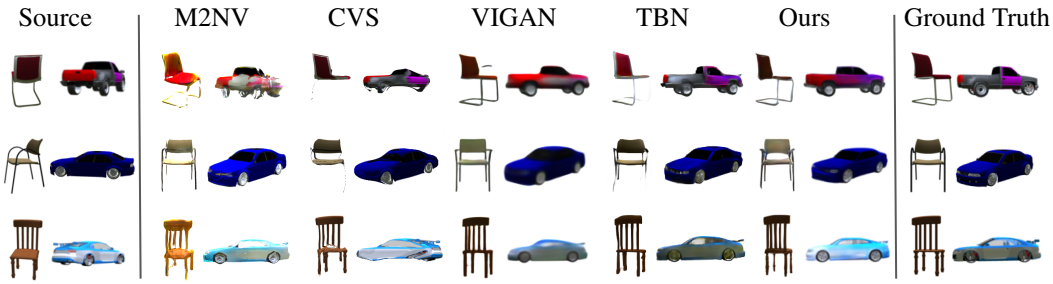


Figure 3: **Qualitative novel view synthesis results.** Our method generates detailed novel views, outperforming most of the baselines. Appearance flow methods fail to generate convincing images for large viewpoint transformations (e.g. top row of M2NV and CVS).

Our transformation chain and 3D feature supervision are shown to be important, as they improve the baseline's performance significantly (Tab. 2). Note that for the $\mathcal{L}_{\text{rec}}$ ablation, we removed only one of the transformation chains: $\hat{I}_1 \leftarrow I_2; \hat{I}_2 \leftarrow I_1$.

Results on continuous scene representation. We compare our method against SRN [46], a state-of-the-art continuous scene representation network. To allow a fair comparison, we use only objects at the intersection of the two evaluation protocols for evaluation. We use a single source image for both models to reconstruct the scene representation, before generating 108 views (ranging between azimuth $0^\circ - 360^\circ$ and elevation $0^\circ - 20^\circ$ angles).

Table 2: **Loss function ablation on the chair dataset.** We report the mean and standard deviation of the L_1 loss and SSIM index for each setup of our trained model. The ablations demonstrate the utility of different aspects of our model.

Setup	w/o $\mathcal{L}_{\text{gan}}$	w/o $\mathcal{L}_{\text{gan}}$, $\mathcal{L}_{\text{vgg}}$	w/o $\mathcal{L}_{\text{rec}}$	w/o $\mathcal{L}_{\text{3d}}$	Full model
$L_1(\downarrow)$	0.0692 / 0.020	0.0703 / 0.021	0.1053 / 0.033	0.0663 / 0.019	0.0645 / 0.02
SSIM ($\uparrow$)	0.7486 / 0.045	0.7489 / 0.044	0.7167 / 0.042	0.7590 / 0.046	0.7627 / 0.04

Table 3 and Fig. 4 show the results. Our method outperforms SRN, even without target view supervision. Our model does not require latent code optimization during inference, which dramatically reduces the inference time from about 5 minutes (SRN) to milliseconds (ours) per image.

Table 3: **Continuous representation results.** We evaluate the ability of SRN [46] and UniCORN to synthesize 108 novel views from a single image on car and chair objects. Our method outperforms SRN on both datasets, even without supervision on the target view.

Methods	Car		Chair	
	$L_1(\downarrow)$	SSIM ($\uparrow$)	$L_1(\downarrow)$	SSIM ($\uparrow$)
SRN [46]	0.04502 / 0.016	0.82804 / 0.049	0.08001 / 0.027	0.7392 / 0.055
UniCORN	**0.03381** / 0.013	**0.8632** / 0.048	**0.06636** / 0.015	**0.7568** / 0.036

Figure 4: **Quantitative results of view sequence generation.** We used a single image to predict 108 target views. Our model synthesizes images with a higher level of detail.

4.3 Applications

Single-view 3D reconstruction. One possible application of our method is to perform single-image 3D reconstruction. We synthesize N novel views on the viewing hemisphere from a single image. From these images, we sample k 3D points uniformly at random from a cubic volume in a similar procedure to the one described in Section 3.4. Our goal is to predict the occupancy of each of these k points. To accomplish this, we project each point onto the synthesized images and label it as occupied if it projects to the foreground mask. Fig. 5 shows the 3D reconstruction results from the input images. Our method captures the overall structure of the objects, and in most cases, their fine-level details. In our experiments we synthesize $N = 15$ images and sample $k = 10^5$ points in 3D space.

Real images. Results reported so far are on synthetic datasets where the input images are rendered from 3D CAD models. To test the generalization performance of UniCORN to real data, we evaluate our model trained with the ShapeNet car objects on the Cars [17] dataset. This dataset contains a

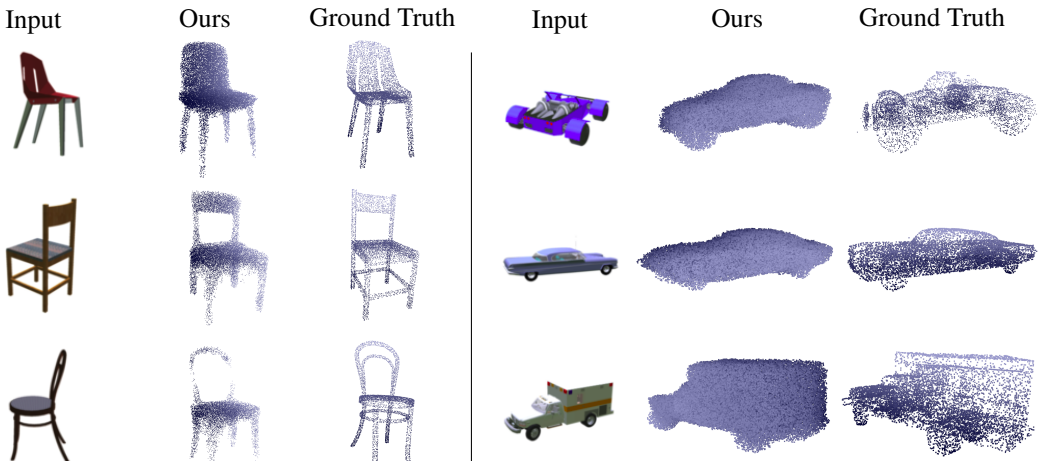


Figure 5: **Qualitative single view 3D reconstruction results.** The synthesized views can be used to produce high quality 3D reconstructions from a single input image.

wide variety of real car images taken from natural scenes. Note that we did not retrain our model on this dataset.

Fig. 6 shows the novel view synthesis of objects given real input images as input. Our method preserves local geometric and photometric features in this challenging setup. This experiment suggests that our model can be used to synthesize images from different datasets, indicating some level of domain transfer capability.

Figure 6: **Qualitative results of novel view synthesis on real data.** Our model generates high quality views of previously unseen data. We use the model trained on ShapeNet and evaluate on the Cars [17] without retraining.

5 Conclusion

We introduced UniCORN, a continuous neural representation for novel view synthesis, learned without any 3D object information or 2D target view supervision. The key component of our system is the use of transformation chains and 3D feature consistency to self-supervise the network. The resulting continuous representation network maps local and global features, extracted from a single input image, onto a spatial feature representation of the scene. Incorporating a differentiable neural renderer allows the synthesis of new images from arbitrary views in an end-to-end trainable fashion. UniCORN requires only two source images per object during training and achieves comparable results (within 2% of the state-of-the-art supervised models) with $50\times$ fewer data and without target view supervision. We demonstrate applications of our model for novel view synthesis, single image 3D object reconstruction, and out of distribution view synthesis on real images. The use of two input views during training to regularize the 3D representation by imposing consistency across the input training images is critical to our method's success. We intend to investigate whether such regularization can be achieved with only one training image per object or without access to the camera poses in future work. We also plan to test our method in real-world applications, including settings for robotic planning and manipulation.

Broader Impact

Our approach to learning novel view synthesis has the immediate possibility of enabling augmented and virtual reality applications [36, 18]. The limited requirement of only two images per object will make these techniques applicable beyond synthetic data. As with any generative model that provides tools for image manipulation, we too run the risk of producing fake visual content that can be exploited for malicious causes. While visual object rotation itself does not have direct negative consequences, the mere fact that such manipulations are possible can erode the public's trust in published images [32]. Image manipulation is not a new phenomenon, however, and there has been research trying to detect manipulated images [40, 2] automatically. Still, more work on and broader adoption of such techniques is needed to mitigate image manipulation's adverse effects.

Acknowledgments and Disclosure of Funding

Funding provided in direct support of this work came from the UMII Graduate Assistantship Award and LCCMR. The authors acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this paper. URL: `http://www.msi.umn.edu`

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning Representations and Generative Models for 3D Point Clouds. In *International Conference on Machine Learning*, pages 40–49, 2018.

[2] M. Barni, E. Nowroozi, B. Tondi, and B. Zhang. Effectiveness of random deep feature selection for securing image manipulation detectors against adversarial examples. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2977–2981. IEEE, 2020.

[3] A. Brock, J. Donahue, and K. Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations*, 2018.

[4] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, and others. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[5] X. Chen, J. Song, and O. Hilliges. Monocular neural image based rendering with continuous view control. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4090–4100, 2019.

[6] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[7] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, 1996.

[8] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, and others. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[9] A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63(2):141–151, 2005.

[10] G. Gkioxari, J. Malik, and J. Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795, 2019.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, and others. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[16] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[17] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[18] M. Krichenbauer, G. Yamamoto, T. Taketom, C. Sandor, and H. Kato. Augmented reality versus virtual reality for 3d object manipulation. *IEEE transactions on visualization and computer graphics*, 24(2):1038–1048, 2017.

[19] T. D. Kulkarni, P. Kohli, J. B. Tenenbaum, and V. Mansinghka. Picture: A Probabilistic Programming Language for Scene Perception. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4390–4399, 2015.

[20] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015.

[21] A. Kumar, S. Eslami, D. J. Rezende, M. Garnelo, F. Viola, E. Lockhart, and M. Shanahan. Consistent generative query networks. *arXiv preprint arXiv:1807.02033*, 2018.

[22] S. Liu, W. Chen, T. Li, and H. Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. *arXiv preprint arXiv:1901.05567*, 2019.

[23] S. Liu, S. Saito, W. Chen, and H. Li. Learning to infer implicit surfaces without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 8293–8304, 2019.

[24] S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, and Z. Cui. DIST: Rendering Deep Implicit Signed Distance Function with Differentiable Sphere Tracing. *arXiv preprint arXiv:1911.13225*, 2019.

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[26] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[27] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *arXiv preprint arXiv:2003.08934*, 2020.

[28] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[29] P. Nguyen-Ha, L. Huynh, E. Rahtu, and J. Heikkilä. Predicting Novel Views Using Generative Adversarial Query Network. In *Scandinavian Conference on Image Analysis*, pages 16–27. Springer, 2019.

[30] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7588–7597, 2019.

[31] K. Olszewski, S. Tulyakov, O. Woodford, H. Li, and L. Luo. Transformable bottleneck networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7648–7657, 2019.

[32] R. J. Oriez. *Do readers believe what they see?: reader acceptance of image manipulation*. PhD Thesis, University of Missouri–Columbia, 2009.

[33] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-Grounded Image Generation Network for Novel 3D View Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3500–3509, 2017.

[34] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[36] M. S. Pinho, D. A. Bowman, and C. M. Freitas. Cooperative object manipulation in immersive virtual environments: framework and techniques. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 171–178, 2002.

[37] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[38] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the Spectral Bias of Neural Networks. In *International Conference on Machine Learning*, pages 5301–5310, 2019.

[39] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[40] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3:1, 2019.

[41] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019.

[42] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

[43] R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

[44] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5541–5550, 2017.

[45] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer. DeepVoxels: Learning Persistent 3D Feature Embeddings. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019.

[46] V. Sitzmann, M. Zollhöfer, and G. Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1119–1130, 2019.

[47] E. Smith, S. Fujimoto, A. Romero, and D. Meger. GEOMetrics: Exploiting Geometric Structure for Graph-Encoded Objects. In *International Conference on Machine Learning*, pages 5866–5876, 2019.

[48] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim. Multi-view to Novel View: Synthesizing Novel Views with Self-Learned Confidence. In *European Conference on Computer Vision*, 2018.

[49] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016.

[50] Y. Tian, X. Peng, L. Zhao, S. Zhang, and D. N. Metaxas. CR-GAN: learning complete representations for multi-view generation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 942–948, 2018.

[51] H.-Y. F. Tung, R. Cheng, and K. Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2595–2603, 2019.

[52] H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4364–4372. IEEE, 2017.

[53] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.

[54] C. Wen, Y. Zhang, Z. Li, and Y. Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1042–1051, 2019.

[55] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson. SynSin: End-to-end View Synthesis from a Single Image. *arXiv preprint arXiv:1912.08804*, 2019.

[56] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Interpretable transformations with encoder-decoder networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5726–5735, 2017.

[57] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 490–500, 2019.

[58] X. Xu, Y.-C. Chen, and J. Jia. View Independent Generative Adversarial Network for Novel View Synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7791–7800, 2019.

[59] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in neural information processing systems*, pages 1696–1704, 2016.

[60] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4541–4550, 2019.

[61] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015.

[62] S. Yao, T. M. Hsu, J.-Y. Zhu, J. Wu, A. Torralba, B. Freeman, and J. Tenenbaum. 3D-aware scene manipulation via inverse graphics. In *Advances in neural information processing systems*, pages 1887–1898, 2018.

[63] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016.

[64] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[65] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Advances in Neural Information Processing Systems*, pages 217–225, 2014.

A Additional Results

In this section, we present additional qualitative results of UniCORN.

Figure 7: **Additional qualitative results.** Given a single source image and a target camera pose, our method generates fine level details even for unobserved parts of the object. We provide the ground truth view of the objects for reference.

Figure 8: **Additional rotation results.** We used a single image to predict 108 views. Compared to the ground truth image sequence (top row per model) our model (bottom row per model) generates target views with high detail.

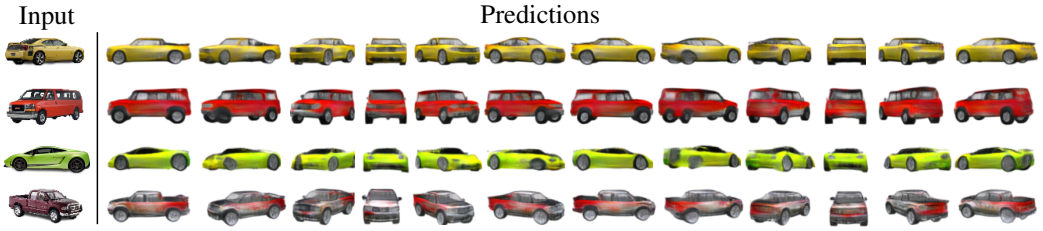


Figure 9: **Additional synthetic → real results.** Given a single image of a real car, we predict 108 novel views. We use our model trained on the synthetic ShapeNet dataset without retraining.

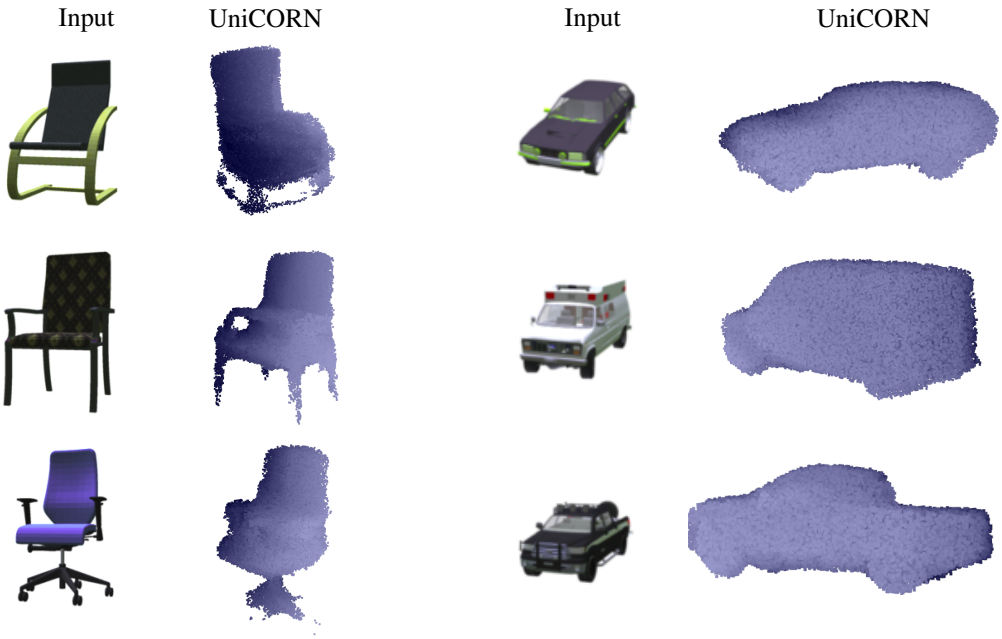


Figure 10: **Additional 3D reconstruction results.** Given a single input image our method can synthesize images from multiple viewpoints to generate the 3D reconstruction of an object.

B Additional architecture details

Here we give more information about the precise architecure used to build our model.

Spatial and global object feature network. We use a pretrained ResNet-18 for global and upsample blocks for the local feature extraction. In particular, we use the setup in Fig. 11a.

Continuous function network. We use an MLP for the continuous function network. In particular, we use the setup in Fig. 11b.

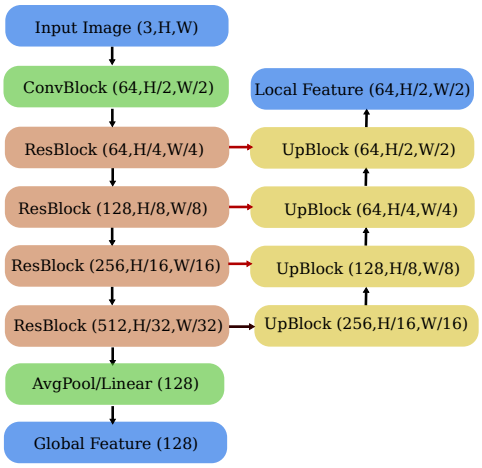
Neural renderer. Our neural renderer based on [55] uses disks of radius 2 pixels for splatting, and stores 16 points per pixel for z-buffering. Please refer to [55] for additional details.

Refinement network. We use a UNet for the refinement network, containing 4 down/upsample blocks with skip connections. In particular, we use the setup in Fig. 11c.

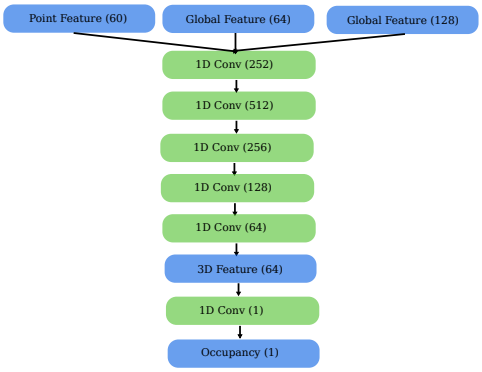
C Baselines

In this section we give additional details on the used baseline methods.

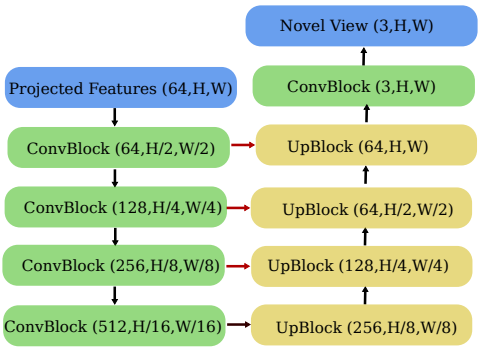
M2NV [48]: This baseline uses appearance flow and confidence estimates to combine values of multiple source images. We use trained weights, made available by the authors, and evaluate their method using a single source image.



(a) Feature extractor



(b) Continuous function network



(c) Refinement network

Figure 11: **UniCORN architecture: (a)** Our feature extractor uses a pretrained ResNet-18 for global feature extraction (left). For local feature extraction (right) we use four upsampling blocks with skip connections (red arrows). **(b)** Our continuous function network takes as input a position encoding, local features and global features and produces a 3D feature at the given spatial location and an occupancy probability. **(c)** Our refinement network contains a UNet with four down/upsampling blocks.

CVS [5]: Another appearance flow method, CVS, integrates a transforming network and estimates a depth map to improve performance. In [5], this method is evaluated for target views that are within $[-40°, 40°]$ rotation from the source image. We evaluate it using the provided models over the entire range of the dataset.

VIGAN [58]: VIGAN is using an implicit feature space, that does not require a 3D representation. We reimplemented their approach and modified their loss functions due to the non-convergence of the original model. We remove all loss functions, except reconstruction, cyclic consistency, pixel losses. The model was trained from scratch on our training dataset for 100,000 iterations (chairs) and 1,000,000 iterations (cars).

TBN [31]: This baseline uses a discrete voxel representation, learned from multiple 2D images. The model disentangles viewpoint transformations and the objects' appearance by introducing a transformable bottleneck layer. This bottleneck layer is transformed with arbitrary SO(3) rotations. We use the provided models for evaluation.

SRN [46]: Scene representation networks are closely related to our method, learning a continuous scene representation with multi-view consistency. We use existing models, trained on 50 source images, and optimize the latent vectors of objects in our test set until convergence.