DistillNeRF: Distilling Neural Radiance Fields into Sparse Voxels for Generalizable Scene Representations

Letian Wang[1,2], Seung Wook Kim[1], Jiawei Yang[3], Cunjun Yu[1,4], Boris Ivanovic[1], Steven Waslander[2], Yue Wang[1,3], Sanja Fidler[1,2], Marco Pavone[1,5], Peter Karkus[1]

pkarkus@nvidia.com

1. Introduction

High-fidelity simulation is crucial for developing autonomous vehicle (AV) technologies and ensuring their safety even in rare situations. An important active area of research is on representing and reconstructing 3D driving scenes given unlabelled sensor streams. Scene reconstruction may enable re-simulating a large number of real-world scenarios collected from a fleet of AVs. Neural scene representations, such as Neural Radience Fields (NeRFs) [15, 17] and 3D Gaussian Splatting (3DGS) [10], have brought unprecedented success in estimating the 3D geometry and appearance of objects or scenes from camera image-pose pairs. They have also been successfully extended to challenging outdoor driving scenes [25], even when densely populated with dynamic agents [30, 32]. However, towards large scale simulations, a key limitation is that these methods train a new representation for each scene, which requires significant compute typically in the order of hours on a modern GPU. In addition to simulation, neural representations are also promising as a self-supervised scene representation in online data-driven AV stacks [7, 9]; however, it is unclear if per-scene training can ever be accelerated sufficiently to run real time using onboard compute.

In this work, we introduce DistillNeRF, a novel model architecture and training strategy for generalizable scene representation prediction of driving scenes without the need for per-scene training at inference time. Our model uses a sparse hierarchical voxel representation with volumetric rendering and a Lift-Splat-Shoot [19] type architecture for lifting and fusing features from multi-view 2-D camera images into 3-D voxels. To train our model we propose to distill information from per-scene optimized offline NeRFs with static-dynamic decomposition, such as EmerNeRF [30]. Specifically, we experiment with dense depth supervision, and discuss alternative strategies such as "virtual" cameras and direct 3D feature regularization.

Preliminary results on the NuScenes dataset [16] suggest that our model is capable of reconstructing and rendering novel views of driving scenes on par with SOTA offline methods that require per-scene training, and it outperforms alternative generalizable scene prediction models. By reducing the compute requirements by orders of magnitude, we believe our work opens up new opportunities for neural representations in re-simulating large number of real-world scenarios, as well as adoption in online AV stacks. Demos and code will be available on the anonymous project page.

[1]NVIDIA Research [2]University of Toronto [3]University of Southern California [4]National University of Singapore [5]Stanford University

2. Related work

Neural scene representations, like NeRFs [15, 17] and 3DGS [10], have brought unprecedented success in learning powerful representations of 3D scenes, and have also been successfully applied to challenging driving scenes populated with dynamic objects [4, 18, 21, 25, 27, 28]. However, these methods typically require expensive training for each scene, ranging from hours to days.

Generalizable neural representation models, such as PixelNerf [33], IBRNet [26], NeuRay [13], and others, learn to predict a neural field representation in a model forward pass. Most of them focus on objects or small indoor scenes. Few recent works explored driving scenes, including NeuralFieldLDM [11], SelfOcc [8], and UniPAD [6]. We show that our model outperforms prior works in reconstructing driving scenes, thanks to our proposed sparse hierarchical voxel representation and distillation strategy.

The idea of distilling larger models to smaller, cheaper models appears in the literature in various forms [5]. Offline trained NeRFs have also been distilled into, e.g., Generative Adversarial Networks in [22], and feed-forward models for temporal object shape prediction in [24]. However, these works mainly focus on *static* object-centric or indoor scenes, leaving challenging dynamic outdoor scenes much unexplored. To the best of our knowledge, we are the first to propose distilling a statically-dynamically decomposed 4D NeRF for training a generalizable neural representation model for driving scenes.

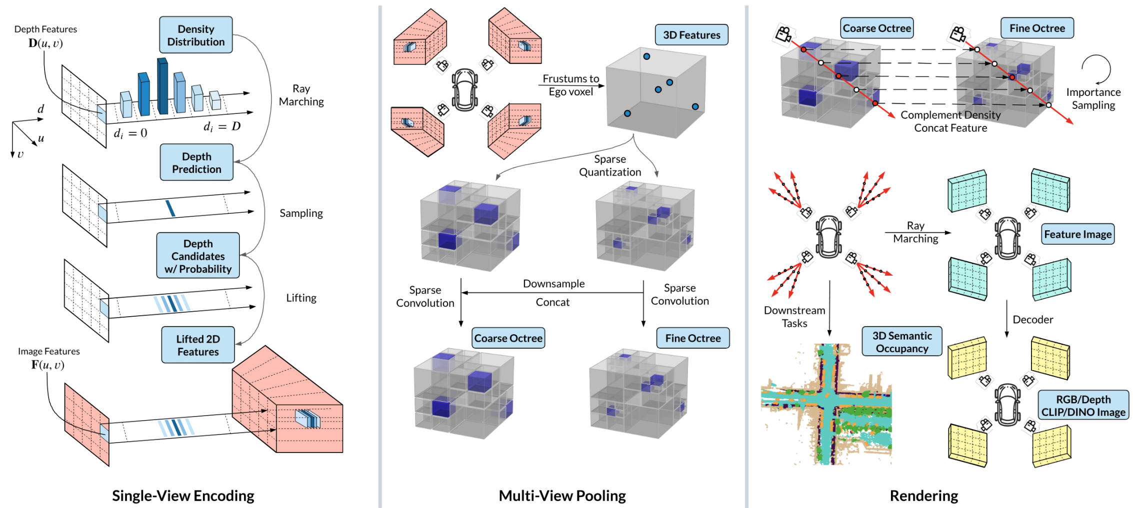

Figure 1. DistillNeRF model architecture. (Left) single view encoding with discretized probabilistic depth prediction; (center) multi-view pooling into a sparse hierarchical voxel representation using Octrees; (right) volumetric rendering from sparse hierarchical voxels.

3. Method

DistillNeRF predicts a generalizable scene representation in the form of sparse hierarchical voxels from multi-view RGB image inputs, and is trained to reconstruct RGB images through volumetric rendering, as well as depth images supervised with per-scene optimized NeRFs.

The model architecture is shown in Fig. 1. The inputs are N posed RGB camera images $\{I_i\}_{i=1}^N$. We use a 2D backbone to extract N feature images $\{X_i\}_{i=1}^N$. We then lift the 2D features to a 3D voxel-based neural field $\mathcal{V} \in R^{H \times W \times D \times C}$ using the corresponding intrinsics and extrinsics, and apply sparse quantization and convolution to fuse features from multiple views. To account for unbounded scenes we use a parameterized neural field with fixed-scale inner voxels, and varying-scale outer voxels contracting the infinite range. Volumetric rendering is performed to supervise the reconstruction of the scene. For better guidance on scene geometry, we "distill" knowledge from offline-optimized EmerNeRFs, using rendered dense depth images.

3.1. Sparse Hierarchical Voxel Model

Single-View Lifting. For each of the N camera image inputs, we follow a similar procedure as Lift-Splat-Shoot (LSS) [19] to lift the 2D image features to the 3D neural field. Specifically, we feed each image to a 2D image backbone to predict a depth feature map. We employ FPN [12] to fuse multi-scale features, and then concatenate them with prior depth features from [31] as the final depth feature map. According to the camera intrinsics, the depth feature map is further embedded as a discrete frustum of size $H \times W \times D$, where D denotes categorical depths. Unlike LSS and variant works [11, 19, 20] which adopts a one-stage depth prediction strategy, we propose a two-stage strategy to capture more nuanced depth. To this end, we first aggregate the frustum to predict a raw depth for each pixel, and then sample fine-grained depth candidates centered around the raw depth. Specifically, inspired by volume rendering equation [15], the frustum is designed to contain the density value for each pre-defined depth. That is, the $d'th$ channel of the frustum at pixel (h, w) represents the density value $\sigma_{h,w,d}$ of the frustum entry at (h, w, d). The occupancy weight of entry (h, w, d) is then calculated as:

$$\text{O(h,w,d)} = \exp(-\textstyle\sum_{j=1}^{d-1} \delta_j \sigma_{h,w,j})(1 - exp(-\delta_d \sigma_{h,w,d})) \quad (1)$$

where $\delta_d = t_{d+1} - t_d$ is the distance between each pre-defined depth t in the frustum. The raw depth for pixel (h, w) is aggregated by:

$$\text{D(h,w)} = \textstyle\sum_{d=1}^{D} \mathbb{O}(h, w, d) t_d \quad (2)$$

Around the raw depth, we sample D' depth candidates t', whose densities σ' are predicted by embedding on the concatenation of the depth feature map and embedded depth candidate features. The occupancy weights $\mathbb{O}'$ of the depth candidates are predicted similarly by running Eq 1. We then run an FPN to get 2D image features ϕ, and assign the 2D image features to the 3D frustum. Specifically, for pixel (h, w), its image feature $\phi_{h,w}$ is distributed to each depth candidates t'_d by $[\mathbb{O}'_{h,w,d}\phi_{h,w}, \sigma'_{h,w,d}]$, where we scale the pixel image feature $\phi_{h,w}$ with occupancy $\mathbb{O}'_{h,w,d}$ and concatenate it with density $\sigma'_{h,w,d}$.

Multi-View Fusion. After constructing the frustum for each view, we transform the frustums to the world coordinates according to camera extrinsic, and fuse them into a shared 3D voxel-based neural field $\mathcal{V}$, where each voxel represents a region in the world coordinates and carries both

Method	Single Timestep Input Frames	No Test-Time Per-Scene Opt	Reconstruction		Novel View Prev Pose		Novel View Next Pose	
			PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
EmerNerf [30]	✗	✗	30.8826	0.8798	-	-	-	-
Single-Frame EmerNerf	✔	✗	31.9675	0.9272	21.1244	0.6087	21.1401	0.6056
SelfOcc [8]	✔	✔	20.6743	0.5561	18.2222	0.4625	18.2223	0.4641
UniPad [29]	✔	✔	19.4449	0.4972	15.6609	0.3337	16.4552	0.3751
DistillNeRF (no distillation)	✔	✔	29.3502	0.8931	19.1657	0.4964	19.2959	0.5034
DistillNeRF (w/ param space)	✔	✔	28.4257	0.8792	20.0754	0.5644	20.0670	0.5653
DistillNeRF (ours)	✔	✔	30.1165	0.9172	19.9534	0.5575	20.2759	0.5678

Table 1. Reconstruction and novel-view synthesis results on the NuScenes validation set. DistillNeRF is on par with the per-scene optimized EmerNerfs and outperforms SOTA generalizable methods.

densities and features. Unlike previous works [11, 23, 29] utilizing dense voxels that uniformly quantize the neural field and spare unnecessary computations/memory on dominating empty spaces, we apply sparse quantization on the neural field for efficient computation. Specifically, we follow the octree representation [14] to recursively divide the neural field according to the 3D positions of the lifted 2D features. While an octree with many levels can capture more accurate 3D positions of lifted features, overly fine-grained octrees can lead to difficulty in querying features during rendering especially when high-resolution sampling is not feasible. To this end, we generate two octrees with different quantization granularities, one fine octree and one coarse octree. Sparse convolutions [3] are then applied to both octrees to encode the relationships and interactions among voxels. The features in the fine octree are also downsampled and concatenated with the coarse octree to enhance the details in the coarse octree.

Neural Field Parameterization. Unlike prior works that consider a neural field covering a fixed range [8, 11, 29], our work aims at accounting for the unbounded-scene settings in the driving scenes by proposing a parameterized neural field. The motivation is that, the neural field should keep the inner-range voxels at the real scale and high resolution for the interest of AV tasks (e.g. occupancy prediction, object detection in the range of [-50m, -50m, -2m, 50m, 50m, 16m]), while contracting the scenes up to the infinite distance into the outer range of the voxels at a lower resolution for rendering with low memory/computation consumption (e.g. sky, far-away buildings). Inspired by [1, 34], we propose a transform function that maps a 3D point in the world coordinates $p = (x, y, z)$ to the coordinates in the parameterized neural field:

$$\text{f(p)} = \begin{cases} \alpha \frac{p}{p_{inner}} & \text{if } |p| \leq p_{inner} \\ \left(1 - \frac{p_{inner}}{|p|}(1-\alpha)\right) \frac{p}{|p|} & \text{if } |p| > p_{inner} \end{cases} \tag{3}$$

The transformed coordinates $f(p)$ will be always within $[0, 1]$, where p_{inner} denotes the range of the inner voxel (region of interest) and varies in x, y, z directions, and $\alpha \in [0, 1]$ denotes the proportion of the inner range in the parameterized neural field. Consistent parameterizations are enforced for both the single-view lifting process (on the depth space) and the multi-view fusion process (on the 3D coordinate space).

Volume Rendering. Finally, we perform volume rendering to project the neural field onto 2D feature maps. Specifically, for each pixel of each camera, we shoot a ray originating from the camera to the neural field according to the camera poses, and sample points along the ray. For each sample point, we query both the fine and coarse octree to get the density and features of the corresponding voxel that the point lies in. Specifically, to capture both high-level information and fine-grained details, the features from both octrees are concatenated as the final feature. Regarding the density, while the fine octree captures more accurate 3D positioning, the sample points could be easily within empty voxels and thus query no information especially in faraway regions, since the fine octree voxel only covers a small region. To this end, for each sample point, we first query the fine octree to get the fine density. If the fine density is zero, we then query the coarse octree to complement the density. Following [1] we sample points for each ray in two phases: first we sample a set of points uniformly, then we sample another set of points with importance sampling given densities for the first set of points, so to enhance surface details in the scene. With the densities and features of the sampled points we do volumetric rendering using Eq 2 to get the 2D feature map for each camera. The 2D feature maps are then fed into a CNN decoder to upsample the final rendered RGB image without increasing the volume rendering cost. Note that from the volume rendering process, we can also get the expected depth for each pixel [21].

3.2. Distillation from Offline NeRFs

While our model can be trained by simply reconstructing input RGB images, it remains challenging to reconstruct scene geometry from only single time-step camera image inputs. The challenge is especially pronounced with typical AV setups where mounted cameras have limited view overlap, making multi-view reconstruction degrade to the

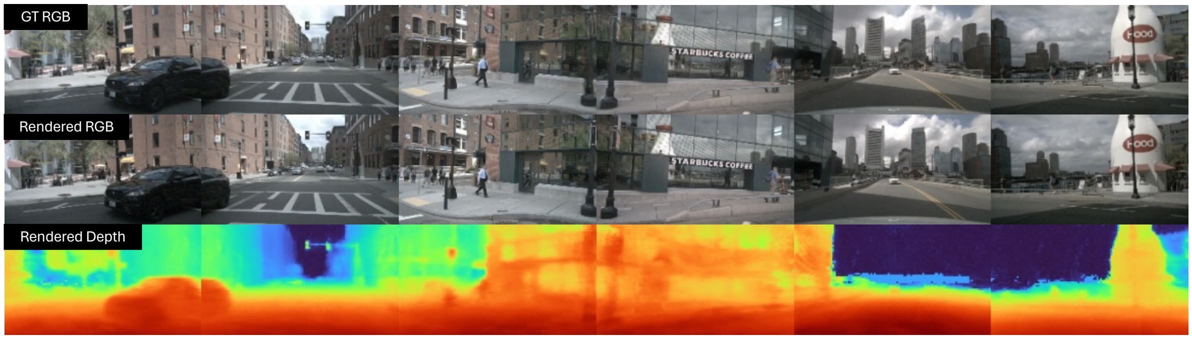

Figure 2. DistillNeRF reconstructs photo-realistic scenes without test-time per-scene optimization. See Appendix for more visualizations.

monocular setting and aggravating depth ambiguity. A natural idea is to use images from multiple time steps, however, driving scenes typically contain many dynamic objects that move between time steps, introducing noise to the reconstruction objective. Instead, we propose to use per-scene optimized NeRFs with static-dynamic decomposition, such as EmerNeRF [30], that aggregate information from a full sensor stream, and decomopses the scene into static and dynamic components. We propose three different ways to distill knowledge from per-scene optimized NeRFs.

- **Dense 2D depth.** Depth supervision from LiDAR point clouds, L_{depth}, is commonly used to facilitate 3D geometry learning, however, point clouds are typically sparse and only provide depth labels for a limited range. We can use offline optimized NeRFs as depth auto-labeling tool, specifically, for each training target image we render a dense depth map from the offline NeRF, and use it as an additional depth supervision, $L_{depth'}$.
- **Virtual cameras.** We can leverage temporally decomposed NeRFs by rendering "virtual cameras", i.e., novel views, while keeping the time dimension frozen. In this manner the number of target images and the view overlap between cameras can be artificially increased. The virtual RGB or depth images can be then used as a reconstruction target, or as a photometric loss for consistent depth prediction following [2, 35].
- **3D voxel regularization.** We can directly regularize the learned scene representation with features extracted from per-scene optimized NeRFs in 3D. Specifically, we predefine or sample a set of 3D points in the scene, and regularize queried features from our predicted neural field, to be similar to those from the offline optimized NeRFs.

In this paper we experiment with dense 2D depth distillation, and leave virtual cameras and 3D voxel regularization to future work. Specifically we use the loss $L = \underbrace{L_{rgb} + L_{depth} + L_{density}}_{\text{rendering}} + \underbrace{L_{depth'}}_{\text{distillation}}$ where L_{rgb} and L_{depth} denote the rendering of RGB and depth, and $L_{density}$ denotes a density entropy loss from [1] to encourage compact rays and sharp object surface.

4. Experiments

We compare the rendering performance of DistillNeRF against both SOTA offline and generalizable NeRFs, specifically EmerNerf [30] trained separately for each scene using all frames from the scene (EmerNerf) or using only frames from a single timestep (Single-Frame EmerNerf); and SelfOcc [8] and UniPad [29] generalizable NeRFs that predict a neural scene representation online similarly to our model.

We report rendering performance for reconstruction, where target frames are the same as input frames, and novel-view synthesis where target frames are defined as the frame at the previous/next timestep to the input frame. We use two common metrics for rendering quality: structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR).

We train our model and the generalizable alternatives on 700 scenes from the NuScenes training split, and evaluate on 150 scenes from the validation split. Due to high computation cost of per-scene optimized NeRFs, we evaluate on only one frame from each scene in Table 1. We observed semantically matching results on the full validation set.

As in Table 1, our method achieves comparable rendering performance to offline NeRFs even without test-time per-scene optimizations, and significantly outperforms prior generalizable NeRF methods in all metrics and evaluation settings. Further, distilling dense 2d depth from per-scene optimized EmerNeRFs helps. The parameterized space slightly reduces the rendering metrics but generates more reasonable unbounded depth as in Fig 2. See Fig 3 and Fig 4 in the appendix for more comparisons.

Conclusions. We proposed a novel model architecture and distillation strategy for generalizable scene representation prediction from multi-view cameras. While we show encouraging preliminary results for scene reconstruction, we expect the quality of novel-view synthesis and 3D geometry could be further improved, e.g., via the discussed additional distillation strategies. Inspired by EmerNeRF, future work may extend our model to render foundation model features alongside RGB, or decompose static/dynamic scene components, and thus beyond efficient 3D simulation, also enable next-generation self-supervised autonomy stacks.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 3, 4

[2] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019. 4

[3] Benjamin Graham and Laurens Van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 3

[4] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023. 1

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[6] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1

[7] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1

[8] Yuanhui Huang, Wenzhao Zheng, Borui Zhang, Jie Zhou, and Jiwen Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. *arXiv preprint arXiv:2311.12754*, 2023. 1, 3, 4

[9] Peter Karkus, Boris Ivanovic, Shie Mannor, and Marco Pavone. Diffstack: A differentiable and modular control stack for autonomous vehicles. In *6th Annual Conference on Robot Learning*, 2022. 1

[10] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 1

[11] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8496–8506, 2023. 1, 2, 3

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[13] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022. 1

[14] Donald Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147, 1982. 3

[15] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[16] Motional. nuScenes Prediction Challenge, 2020. Available at https://www.nuscenes.org/prediction?externalData=all&mapData=all&modalities=Any. 1

[17] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 1

[18] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 1

[19] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 1, 2

[20] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8555–8564, 2021. 2

[21] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022. 1, 3

[22] Mohamad Shahbazi, Evangelos Ntavelis, Alessio Tonioni, Edo Collins, Danda Pani Paudel, Martin Danelljan, and Luc Van Gool. Nerf-gan distillation for efficient 3d-aware generation with convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2888–2898, 2023. 1

[23] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 3

[24] Jeff Tan, Gengshan Yang, and Deva Ramanan. Distilling neural fields for real-time articulated shape reconstruction.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4692–4701, 2023. 1

[25] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 1

[26] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 1

[27] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8370–8380, 2023. 1

[28] TH Wu, FC Zhong, A Tagliasacchi, F Cole, and C Oztireli. D2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. In *NeurIPS*, 2022. 1

[29] Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, et al. Unipad: A universal pre-training paradigm for autonomous driving. *arXiv preprint arXiv:2310.08370*, 2023. 3, 4

[30] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 1, 3, 4

[31] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024. 2

[32] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. 1

[33] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 1

[34] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 3

[35] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 4

DistillNeRF: Distilling Neural Radiance Fields into Sparse Voxels for Generalizable Scene Representations

Supplementary Material

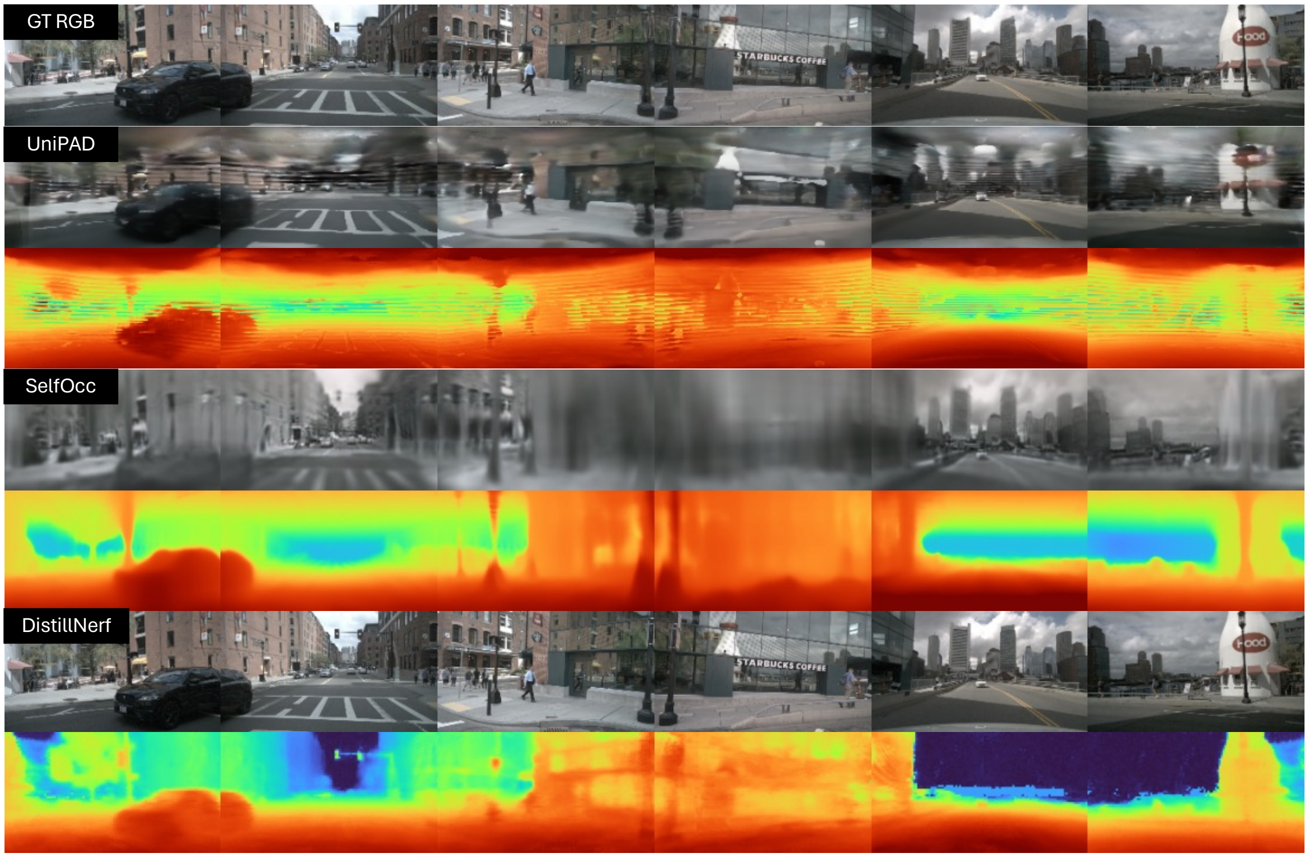

Figure 3. RGB image and depth image rendering comparison with SOTA generalizable NeRFs methods. Our DistillNerf can generate high-quality scene reconstructions and depths.

Method	Venue	No Test-Time Per-Scene Opt	Reconstruction		Novel View Prev Pose		Novel View Next Pose	
			SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑
EmerNerf	ICLR 2024	✗	30.9927	0.8802	-	-	-	-
SelfOcc	CVPR 2024	✔	20.7220	0.5564	18.1539	0.4591	18.1673	0.4580
UniPad	CVPR 2024	✔	19.4970	0.4965	15.5741	0.3229	16.4667	0.3663
Ours		✔	29.3446	0.8922	18.8458	0.4746	18.989	0.4810

Table 2. Rendering comparison with per-scene NeRF optimized on the full validation set.

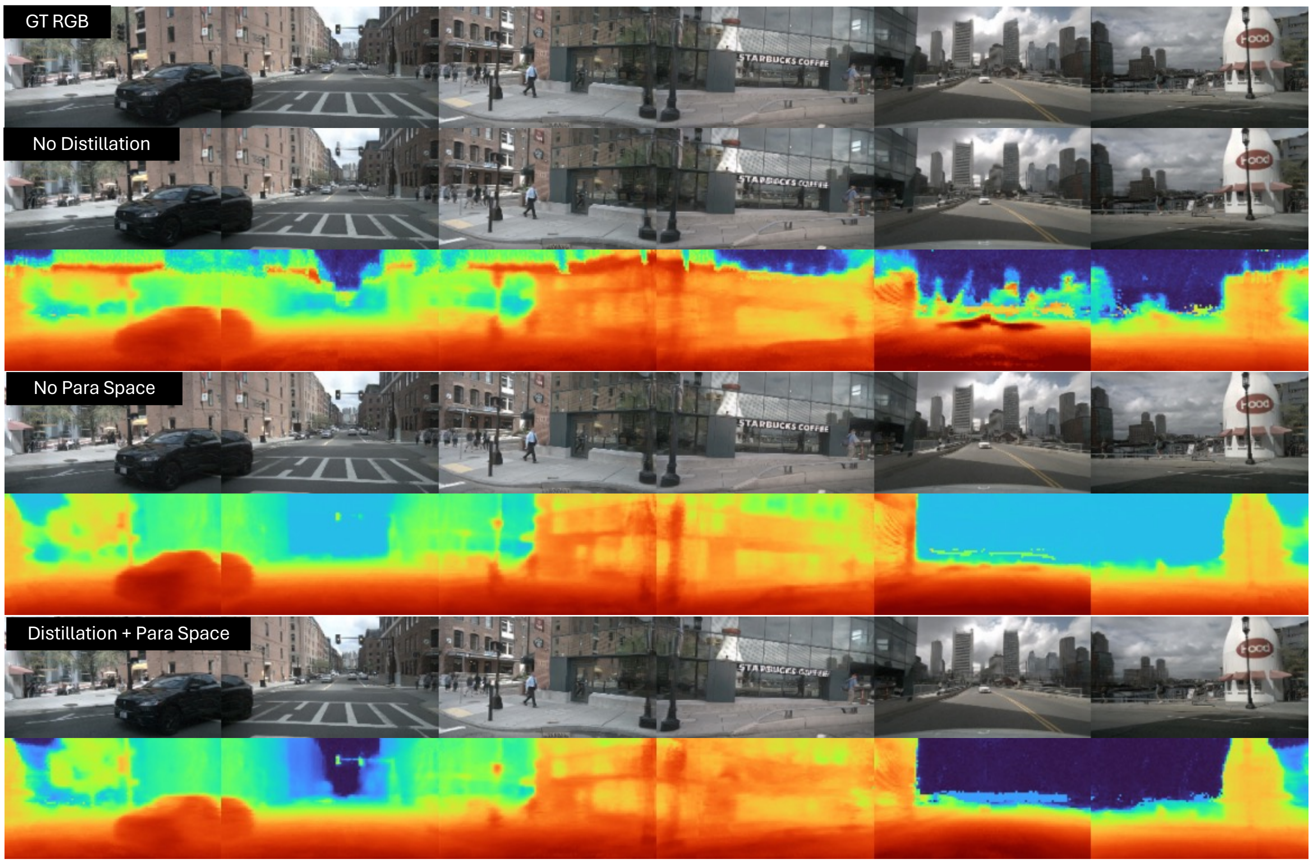

Figure 4. RGB image and depth image rendering ablations on the distillation loss and parameterized space. Our model can predict high-quality depths including far-away regions.