Supplementary: Grid-guided Neural Radiance Fields for Large Urban Scenes

1. Overview

In this supplementary, we first 1) elaborate the problem background and general challenges for large-scale scene modeling, and then glimpse at more 2) dataset details. We then go into the framework details, and show 3) more results on different datasets and compare 4) various ablation results on module design. 5) Illustrations for verifying our model design effects are also analyzed. We also discuss 6) several techniques for practicing visual improvements.

Three demo videos are provided. 1) *Video A* shows the novel view renderings for traveling through the three scenes. 2) *Video B* includes the *Rubble* scene, showing the results exiting from the two branches. The *left frame* is rendered from the *grid branch*, and the *right frame* is from the *NeRF branch*. It can be seen that the NeRF branch provides better renderings in terms of visual smoothness and sharp details for long videos. 3) *Video C* shows a comparison with MegaNeRF's partition solution on *Campus* scene, where a NeRF based method struggles with model capacity issue when the scene scales up.

Figure 1. Demo video frame for two-branch outputs.

2. Background and Challenges.

Large-scale 3D scene reconstruction and rendering is a long-standing problem in computer vision and graphics, and also an exciting and important application in our daily life. A recent line of works has been proposed, targeting the *neural rendering* for urban scenes, motivated by the compact representation of these *implicit scene representations*. Multiple challenges for large urban scene modeling and rendering have been actively discussed. For example, BlockNeRF [13] learns multiple NeRFs from the autonomous driving data capturing San Francisco; MegaNeRF [15] also provide a division solution to partition large areas; Urban Radiance Fields [11] highlight the issue of sky modeling and exposure variation; BungeeNeRF [18] targets the learning from multi-scale images; Mip-NeRF 360 [2] extends from Mip-NeRF [1] and address the unbounded issue in scene modeling, propose a distortion loss for regularizing floating ray points, and additionally use a proposal network to guide the efficient sampling of NeRF model.

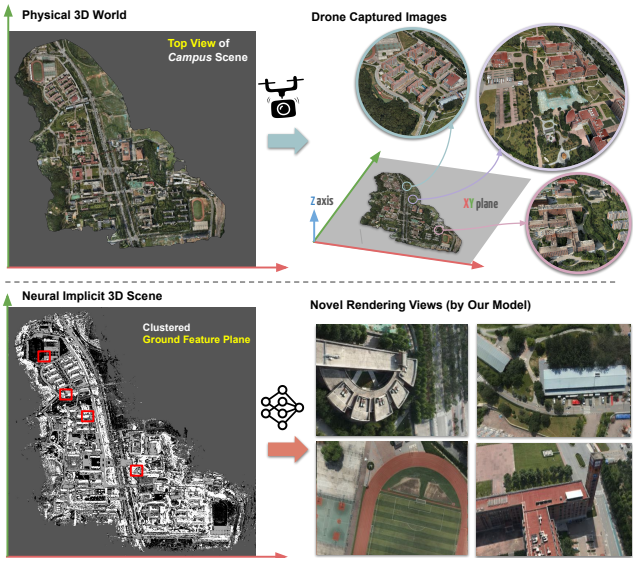

Figure 2. (1) Illustration of *Campus* scene, where common urban objects (*e.g.*, buildings, plants) are shown in the zoom-in pictures. The majority of scene contents are wide spread in the xy-plane compared to the limited scene height along the z-axis. The scene covers over $2.7km^2$ with over $5k$ images, where the average pixel footage only occupies a very small portion of the entire scene, leading to the large-scale nature of our target scenario. (2) Our methods embed such large physical world into a neural implicit representation, where we store the information in a set of ground feature planes, as shown on the left. With our representation, an MLP renderer is able to convert the latent feature into RGB and density, and provide novel view renderings for new camera poses within the scene. Unlike the traditional way of storing such scenes in textured meshes with extremely large number of vertices, neural based methods offer a compact and adaptive way to embed large 3D scenes, enabling photo-realistic rendering results.

A popular alternative for NeRF-based representations is to construct a feature grid for the target scene, where a render MLP is used to translate the hidden features for RGB

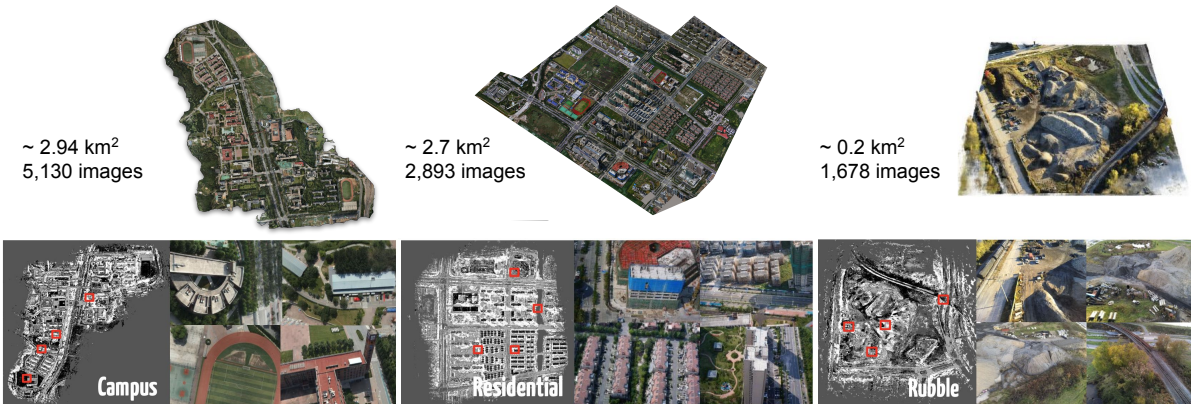

Figure 3. A glimpse of our experimental real-world scenes with self-learned ground feature maps. Our rendering results are supplied on the right with annotated red dots for the estimated locations..

and density, *i.e.*, using a hybrid representation. Existing methods (*e.g.*, TensoRF [4], Instant-NGP [10], DVGo [12]) assume when the grid resolution is high enough, grid features can faithfully encode scene geometry and texture and a small MLP renderer is sufficient to translate grid features into density and color, leading to superior efficiency compared to pure MLP-based models. However, such an assumption is challenged by large urban scenes, which are 1) large-scale, 2) have rich and complex details, and 3) have limited drone views. While grid-based approaches can increase the resolution to match the scale, they often suffer from severe degradation in quality due to several factors. First, each grid is independently optimized and thus lacks the inherent continuity that comes with MLP's information-sharing nature. Second, the inadequate non-linearity within a grid unit via interpolated grid values also contributes to degradation. Third, a small renderer MLP may struggle to interpret the large feature space from the high-resolution feature grid with limited capacity

Fig. 2 illustrates the application scenario of our method. We aim to embed a large urban scene in physical world into a compact neural implicit scene representation. Targeting the aforementioned challenges in modeling large urban scenes, we seek a *model-level solution* to resolve the challenges orthogonal to the commonly adopted partitioning techniques [13, 15]. Our two-branch model unifies the two representations (NeRF-based & Grid-based) by taking advantage of 1) the fast learning of a coarse scene with explicit grid features, and 2) the inductive bias of large MLPs with high-frequency PE inputs for learning a globally smoother and locally more accurate scene representation. The NeRF-branch here learns to accurately represent the scene by referring to and refining grid features, more than a simple MLP renderer as in previous works. It regularizes and improves the grid features to encode more within-grid variations that can better supplement PE inputs in modeling scene details, being more effective than direct RGB supervision. Consequently, the regularized grid features also benefit the small MLP renderer to interpret a more compact feature space with sufficient capacity to translate an increased amount of grid values.

3. Dataset

To demonstrate the effectiveness of our method, experiments are mainly conducted on real-world scenes that are challenging for both NeRF and grid-based methods. Our two-branch design is applicable for standard NeRF datasets [9] and image/shape fitting settings, and is found more advantageous in complex large urban scenes.

The main experiments reported in the paper are three real-world urban scene datasets, each containing thousands of high-resolution images captured by UAV. A typical drone capturing path is shown in Fig 4. These scenes are generally bounded, as shown in Fig. 4. Since they are captured by a drone under a single scale, the difference is slight between NeRF/Mip-NeRF's results. The three scenes depict diverse urban environments, including rural rubble sites [15] (*Rubble*), university campus (*Campus*), and residential complexes (*Residential*). The camera poses are obtained from the off-the-shelf commercial photogrammetry software ContextCapture, which is widely used for creating an engineering-ready 3D model from oblique photography in capturing city-wide data. On some of our data, we find COLMAP/PixSFM failed to extract poses, possibly due to texture-less areas and sparse views, whereas ContextCapture can still estimate poses with specified control points and parameters. Moreover, its extracted camera poses are automatically in ENU coordinates that better fit the ground plane representation by aligning the expanded xy-plane with the physical ground, making it easier to verify the learned feature plane and more suitable for urban analysis. The camera poses are further normalized to be within a unit ball in pre-processing. Since the original scene size is large, it is therefore necessary to use high-frequency Fourier

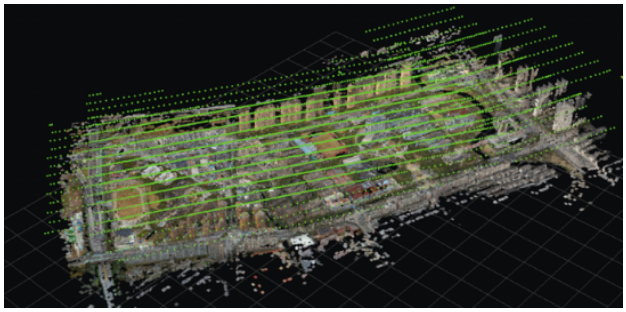

Figure 4. A typical drone flying pattern (camera poses) for a five-lens oblique camera when capturing an urban scene shows difficulty on modeling with limited viewing angles.

encoding (*e.g.* 2^{15} as used in our experiments) to reflect the finest pixel variation in imagery.

As demonstrated in Fig. 3, the *Campus* dataset spans a ground area of $\sim 2.94km^2$, with a total of 5,130 images. The *Residential* dataset spans a ground area of $\sim 2.7km^2$ with a total of 2,893 images. The images were captured by a five-lens oblique camera. The *Mill-Rubble* is contributed from MegaNeRF [15] and covers a nearby construction area full of debris, consisting of 1,678 images. The difficulty of learning such scenes with implicit neural scene representation *varies*. Experimentally, we found it relatively easy to model *Rubble* where the scene contents lies on the grounds, with little variation in height and sharp boundaries. In contrast, the *Residential* scene is orders of challenging to achieve high metric scores, considering its high complexity and diversity of scene contents (*e.g.*, the tall buildings, the thin scaffolds, the near-ground lake, etc). To ensure an efficient sampling on image pixels with allowable storage and training time, in experiments, we downsample the raw image resolution to moderate resolutions accordingly. MegaNeRF trained on full resolution requires over 30 hours on 8 A100 GPUs. We use downsampled resolution (*e.g.*, 1K) for training and validation to avoid OOM issues, as also suggested in their code repository, and accordingly adopt a smaller model configuration of MegaNeRF baseline, which is sufficient to demonstrate our ideas on a single A100 GPU.

4. Framework Clarifications

Grid factorization with ground plane: Our one-dimensional factorization (*i.e.*, representing the major scene with a ground feature plane) is particularly designed for large scenes with widespread content in xy-plane. For such scenes, using xy-plane can already capture the large amount of information in 3D, with comparable performance compared to the one using full three-dimensional factorization. For example, the *Residential* scene has a large amount of high-res buildings, but it can still be well handled via our factorization, since urban scenes usually have compact

Figure 5. Typical artifacts due to noisy grid features are commonly found in modeling real-world scenes with Instant-NGP [10] with no additional regularizations applied(*e.g.*, distortion loss [2], total variation loss [4]). The issue becomes exaggerated when applying to large-scale scenes with severely downgraded rendering quality.

Figure 6. Grid artifacts of Instant-NGP [10] and TensoRF [4]. Instant-NGP (with implementation [14]) appears noticeable artifacts with discrete boundaries. The artifact maintains when we increase the resolution and hashtable size to 65536 and 2^{23}; while TensoRF appears blurry and wavy textures within and across grids.

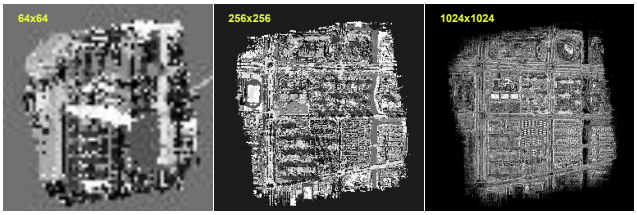

Figure 7. Multiresolutuion feature planes capture the scene contents with different granularities. We found such design greatly improve the stability of learning for large scenes and can generally improve the rendering quality without noticeable artifacts caused by the discrete feature grids.

and repeated information along the z-axis. We do not expect the pre-trained grid to recover all the details since it will be further rendered by the NeRF-branch, as verified in Fig. 15. The performance difference was minor when using full mode but requires more parameters. Moreover, the shared z-axis encourages learning a more informative and reliable ground plane. This compactness if useful when we use sparse view images for training, where full 3D grid-based methods can easily overfit to images with floating points, as shown in Fig. 5. It also encourages distinct feature grids covering similar 3D contents to take closer grid values, which can be useful for clustering categories from the feature plane for object discovery, as discussed in Fig. 16.

Multi-resolution feature grids. It is designed to capture scene contents at multiple granularities, which is demonstrative strong representation power [10], and is suitable for urban scene modeling with objects of different scales, as visualized in Fig. 7. A single-resolution one obtained by progressively upsample [4] is inferior in representation power and behave sensitive to the choice of upsample steps. A single grid resolution (*e.g.*, TensoRF [4]) can lead to noticeable grid artifacts in regions across boundary. Instant-NGP [10] with multiresolution grids only, leads to severe discrete grid artifacts in rendering, as shown in Fig. 6. Additionally, 3) the separate factorization of density and RGB fields allows the disentanglement of concrete density and appearance variations (*e.g.*, casted shadows), as demonstrated in Fig. 17.

Grid branch vs. NeRF branch. Grid branch contains a 2-layer small MLP renderer directly translating the grid features to density and colors. NeRF-branch is a 4-layer MLP, which takes both PE and grid features as inputs and has a reasonable capacity to encode finer scene details based on the grid features. Both branches' outputs are supervised with the ground truth pixel value. Intuitively, this encourages the inputs to the two branches to maximize their information for predicting the 3D scene content.

PE appended to Grid branch. Naively enlarging the MLP renderer and/or injecting high-frequency PE (like DVGo) only leads to minor improvement (PSNR 0.1-0.3dB). We conjecture that the grid features originally used for encoding scene contents are now entangled with PE inputs. The two-branch supervision is thus necessary to maintain informative grid features while letting PE capture high-frequency details. A comparative ablation result showing the *rectified feature plane* and the need for *grid branch supervision* is shown in Fig. 8.

Two Stage vs. Joint Train from Scratch. While joint training from scratch can deliver similar effects for small-scale scenes or objects, the training is much slower as we initially wanted to utilize the simplicity and speed of grid branch. Meanwhile, NeRF alone is already hard to train on *large-scale* scenes, let alone concatenating with not well-trained features from grid branch at the beginning. While NeRF is stronger with deeper MLPs, its rendering speed is much slower than the Grid module.

Fast rendering with Grid Branch. After the joint training, the rendering quality of the grid branch is improved whilst maintaining its rendering efficiency. The NeRF branch can be optionally excluded at inference time to realize fast rendering of high-quality results (10x speed-up). This is usually enough for per-image rendering. However, for high-quality video rendering, NeRF branch is still preferred.

5. Visual Quality Improvements

To further bring high-fidelity rendering results targeting on modeling large urban scenes, a series of special 2D image-level techniques can be considered here.

Perceptual Losses: It has been vastly observed in generative tasks where reconstruction loss on RGB is not powerful enough to obtain photorealistic results. To further boost the visual quality when rendering the entire images, we consider imposing a perceptual loss [19] on rendered patches in later training stages. For datasets depicting natural scenes, such perceptual losses can help reveal sharper details compared to pure MSE losses based on per-pixel rendering procedure, as shown in Fig. 9. For each iteration, we randomly select a batch of image patches from a sampled image. We find that a large patch size is preferred (*e.g.*, 64×64 or 128×128) to cover larger areas. For example, roads and stripes can get better visual quality by displaying straight lines that may otherwise suffer from little distorsion. GAN-based [6] losses have also experimented. The results are aligned with our expectations where regular patterns such as windows and patterned facades gain generated texture, as shown in Fig. 10. Note that these losses are applied only when the MSE loss is relative low, *i.e.*, the scene has been reconstructed properly. Otherwise, such losses may easily disrupt the training process. Moreover, as perceptual loss evaluates on the feature space, the reconstructed color may appear slightly deviate from the ground truth. Therefore, a smaller loss weighted ratio between MSE and perceptual loss is also needed.

Super-resolution Module. As mentioned earlier, the real data captured by a drone (*e.g.*, DJI) could have over 10000 pixels on one side. We experimented with the super-resolution module in EG3D [3] with $\times2$ and $\times4$ settings. Note that both these image-based techniques cannot guarantee strict 3D consistency, which is not ideal for rendering long videos, but can generally improve the visual quality.

6. More Results and Additional Analysis

More visualized rendering results on our main scenes are shown in Fig. 12. The rendered images with planed path

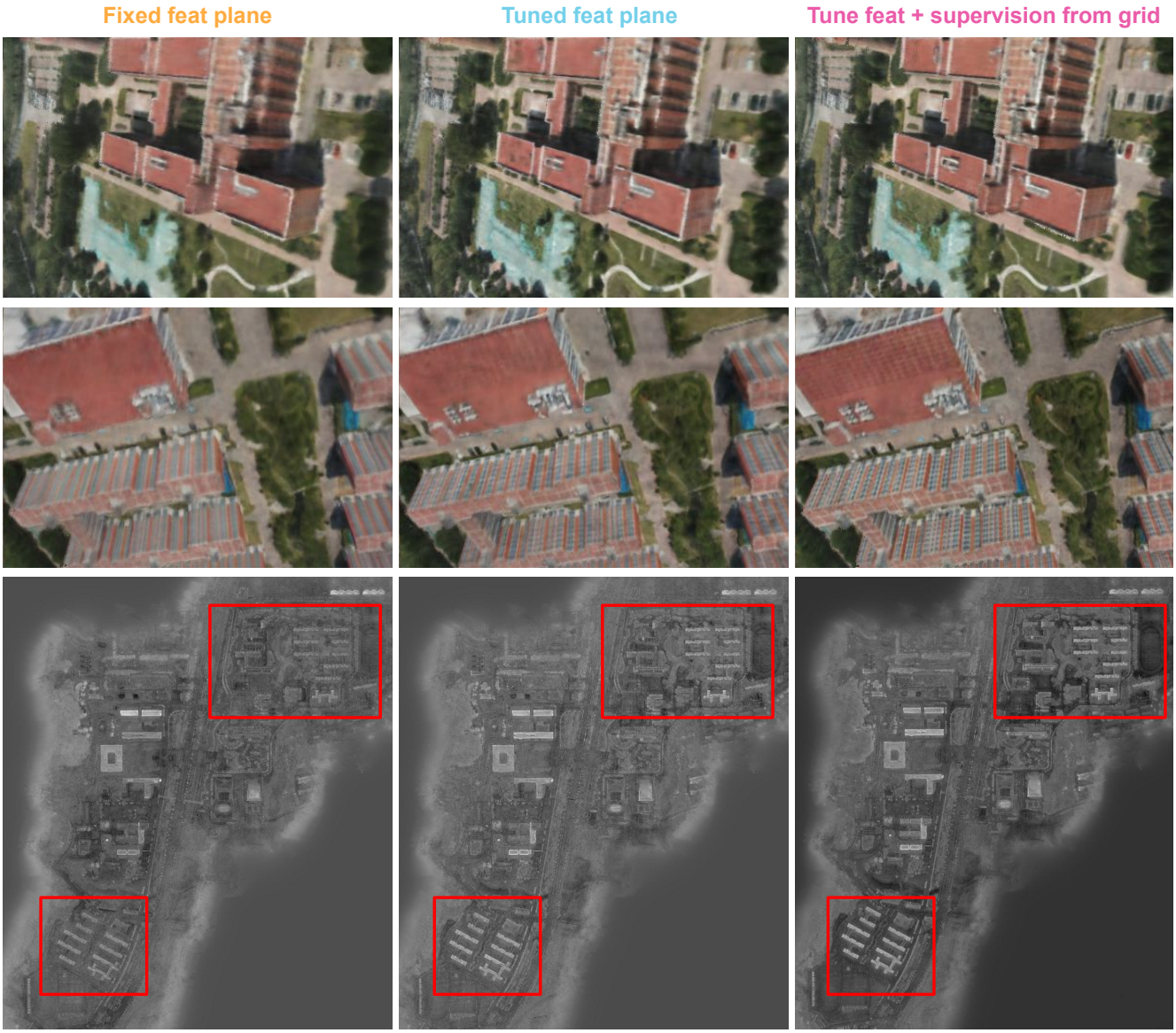

Figure 8. Grid branch rendering. The first column (fixed feat plane) represent a baseline treatment of DVGO [12]. We show on the later two columns of the effectiveness of allowing the finetuning of grid branch during the second stage joint training, and the direct supervision from the output head of grid branch.

give an immersive navigation experience in the large scene.

Small-scale Scenes. We also test our methods on Mip-NeRF 360 data, achieving realistic rendering results with our proposed representation and training design. As these scenes are generally small-scale, where a vanilla NeRF or existing grid-based methods can efficiently handle, our methods' use may seem unnecessary, and the advantage of supplying two-branch is relatively minor. Still, we can verify the effects of adding NeRF module to supply the grid features. As shown in Fig. 11, the grid branch shows a homogeneous level-of-details rendering across the scene, while NeRF branch builds upon the grid features further recover the fine details on the delicate areas like a stump, leaves, bottle, and object textures. Compared to NeRF-based methods as adopted in Mip-NeRF 360 [2], the training of our methods is much faster as the grid pre-train stage can fastly capture most scene details with fewer iterations. As our methods are orthogonal to the techniques (*e.g.*, proposal network, space warping, distortion loss, *etc.*), further improvements can be expected with additional inclusion.

Frequency Comparison. We visualize the activation of frequency channels in the input positional encoding of

Figure 9. Perceptual losses (*e.g.*, LPIPs [19]) can enhance overall image quality, providing more natural looking images matched with humen perception.

Figure 10. Applying super-resolution module (on feature space) or GAN- losses can add in more fine details for the final rendering, especially for regions with geometric structures like buildings.

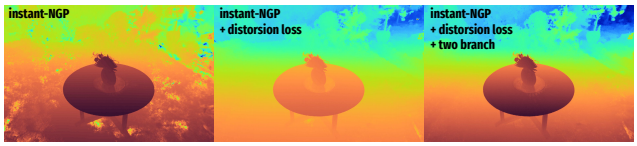

Figure 11. Applying our two-branch structure to Instant-NGP [10] based representation. Without any regularization on the feature values, the learned geometry gets slightly noisy. We show that with the additional NeRF branch and our joint training scheme, together with the useful distorsion loss [2], resulting in the clean and sharp geometry compared to its vanilla version.

NeRF in Fig. 15 and the frequency domain comparison in Fig 14. It shows that the vanilla NeRF suffers from the heavy learning burden where the low-to-high frequencies are responsible for learning all coarse-to-fine details in the scene, and fails to activate higher frequency channels. However, our approach encourages it to utilize such information to model scene details. It is also noteworthy that such a response is stronger among the z-axis encoding, indicating that NeRF put more effort on making up the missing content along the z-axis.

7. Discussions and Future Works

Applicability. Our method can be intuitively integrated with other grid-based methods and their variants (e.g. TensoRF [4], DVGo [12], Instant-NGP [10]). Fig. 11 show the depth map rendered from the Instant-NGP with improved quality after the two-branch training, which exhibits more accurate details and less floats. Common rulebased regularizations, e.g. TV loss [4], distortion loss [2], are suggested to be used in combination for better quality

Camera Poses. Currently, we rely on poses estimated by a commercial professional software (ContextCapture) without further adjustment. However, as pointed out by MegaNeRF [15], the camera pose's accuracy may significantly impact the final results. This effect has been observed when we try to model buildings with sharp edges. The inaccurate camera poses can prevent us from getting accurate boundaries for the buildings and introduce unwanted flickering when rendering around such props. Therefore, it is best to be combined with camera parameters optimization, as introduced in [7, 17] with relative good initial poses.

Another interesting application is to allow the multi-view image fusion with the neural rendering pipeline. While the views captured by a drone can provide the overall scene context, a dive-in into street-level views is highly desired to achieve a more immersive experience. How to provide accurate camera poses from these two sources of images in a consistent world coordinate is also challenging. While this could be easily done in a virtual presence, it is generally challenging for large-scale capturing in the real-world.

Transient and Dynamic Objects. It is also noticed that urban scenes are generally dynamic, with moving vehicles and constantly varying lighting conditions. Without modeling these dynamics, resulting views can be flickering with inconsistent content. One solution is to associate each view with a latent appearance code that will be jointly learned, as has been vastly adopted in previous works [5, 8, 13]. Note that these lines of methods penalize transient objects by assigning a smaller weight on the self-learned instance masks, which may cause ambiguity with those image regions with relatively higher MSE errors with hard textures (*e.g.*, high-frequency details). Alternative solutions like using off-the-shelf detection models or self-learned depth priors could be promising directions. Note that the displayed examples in our paper did not incorporate lighting variation control. We have experimented with latent codes, which are able to capture light variations and enable smooth interpolation between different lighting conditions.

Unbounded scene. The drone data for large-scale city scene reconstruction are typically in grid-pattern with 45 to 90 look-down angle. For unbounded scenes, we can deploy the space contraction technique from MipNeRF-360 [2] to account for extremely far-away sky, or using a spherical background grid to account for far-field background.

Scene Editing. Note that, the adoption of our ground feature plane reveals the potential of direct performing scene editing on 2D feature planes. As feature grid values reflects

Figure 12. Selected Rendering results from 3 scenes with our methods.

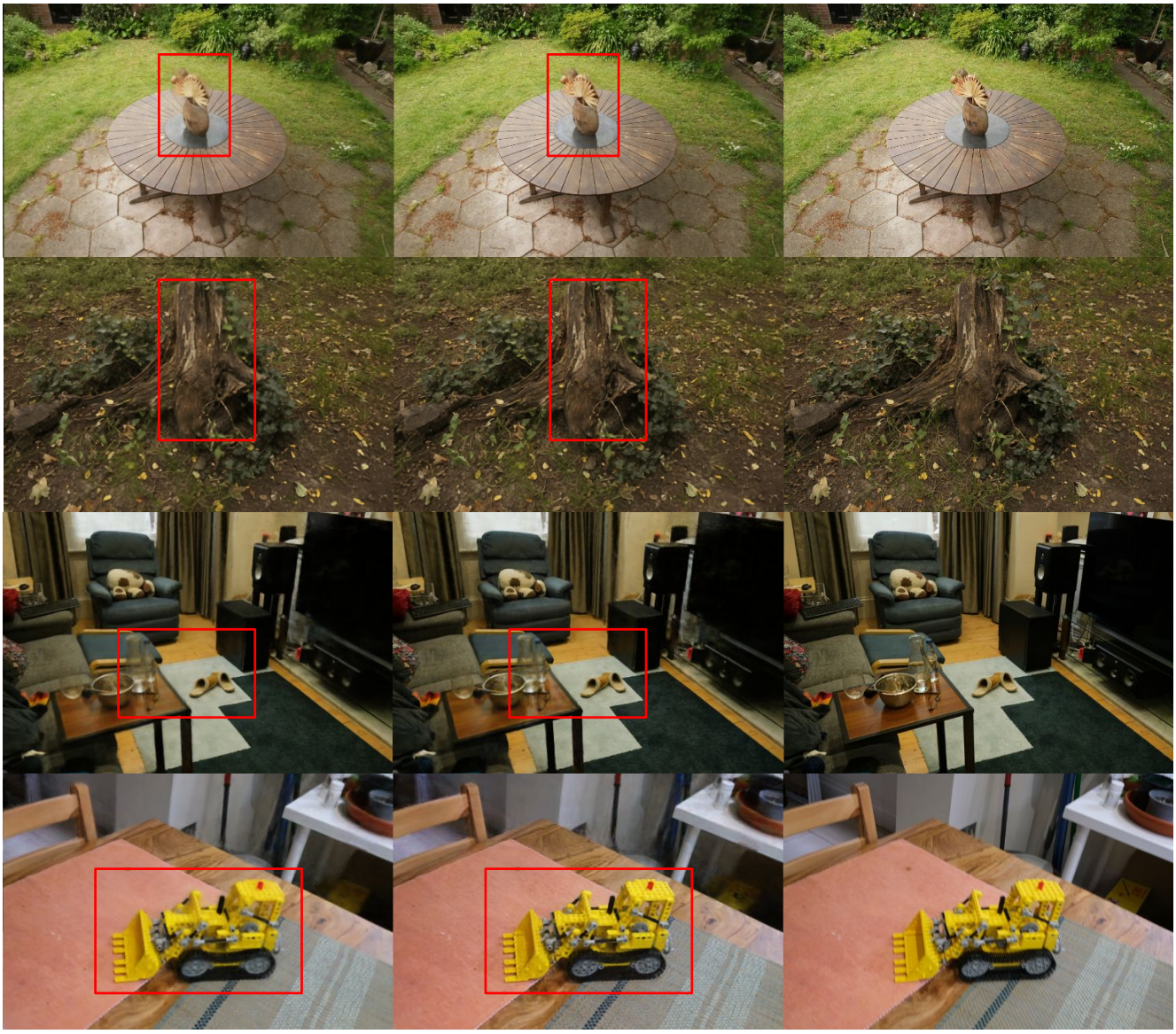

Figure 13. Results on MipNeRF-360 [2]. For these small scenes, we use a moderate resolution for feature grids here, in order to verify the effectiveness of resorting to NeRF's positional embedding to deliver the high-frequency details. Though the feature grids are good at capturing local contents even for a large-scale environment, the detailed textures are better recovered with NeRF branch, as shown above.

the local scene contents by storing features on the grid vertices, the manipulation of local areas will not affect other regions, which is hard to achievable with NeRF-based methods. The integration of real-time rendering speed with grid-based methods may reach the demand for interactive editing applications on large scenes.

Scalability. Scalability can still be our potential limitation when facing larger scenes. As our approach still conditions on the pre-trained feature grid, in general we expect a moderate to high grid resolution to capture sufficient detail for NeRF to refine. Although we have relaxed such requirement compared to pure feature grid-based approaches, it can still be a bottleneck that restrict neural rendering on large-scale scenes. A feasible solution is to further combine our method with geographical division as adopted by BlockNeRF [13] and MegaNeRF [15], while assuring the rendering of each sub-region can be as photorealistic as possible.

Another critical question to address is the scale-up of **image numbers** and **resolutions**, where a single image can be as large as 151 megapixels. Training with the image reconstruction process can make these practices hard to scale

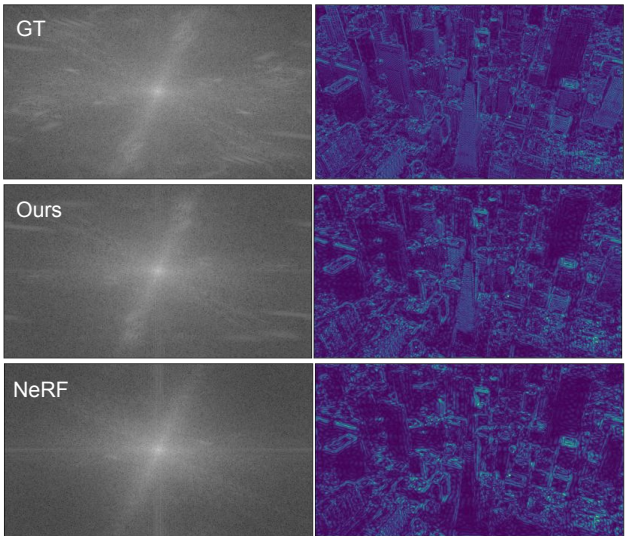

Figure 14. Fourier transformed rendering results demonstrate the effiency of our methods in revealing fine-details for large-scale scene rendering. NeRF-based method fail to capture the high frequency details efficiently.

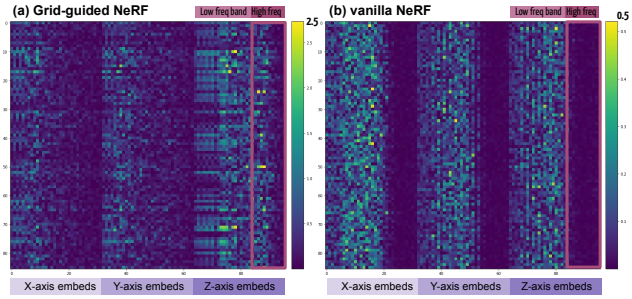

Figure 15. Weight matrix analysis. We show the weights associate to positional encoding channels ranging from 2^0 to 2^{15} frequencies (in total 96 channels), where the 3 block for each matrix correspond to x, y, z dimensions. The weight pattern of (a) our NeRF branch under guidance of grid features has relative easier learning burden on the x, y dimensions which is largely captured by the multi-resolution feature planes. Meanwhile, the high-frequency channels along z-axis get well exploited to provide complementary information. In contrast, (b) the vanilla NeRF suffers from large learning burden where all coarse-to-fine details in the scene are required to be reasoned. As indicated in the value bars, all the parameters have relative small activation values compared to (a), and the high-frequency components are rarely activated even after a long training process.

up, especially when the real-world captured data are from extremely high-resolution video captures. Generally, we can consider applying techniques used in [2, 18] by starting with the downsampled version and applying progressively high-resolution images to add in details. Additional 2D image techniques such as adopting a super-resolution

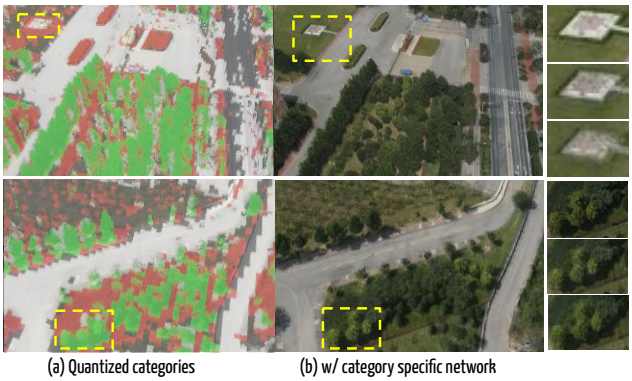

Figure 16. We exploit the learned 2D feature plane for clues of scene contents with similar color and geometry, which can be treated as an unsupervised discovery of semantic categories. The clustered feature grids via Vector Quantization (VQ) [16] show clear clue on places for trees, bushes, and roads, which indicate the compact latent feature spaces capturing objects with similar appearances and geometries with close feature values. The rightmost column shows the comparison between an experimental VQ-based NeRF (middle) and a single NeRF (bottom) for our NeRF branch, where the ground truth patch is shown on the top. The cluster-aware sub-NeRFs bring more object-aware renderings even on difficult areas like bushes with clearer edges and shapes, where a global NeRF may have difficulty to distinguish these semantics.

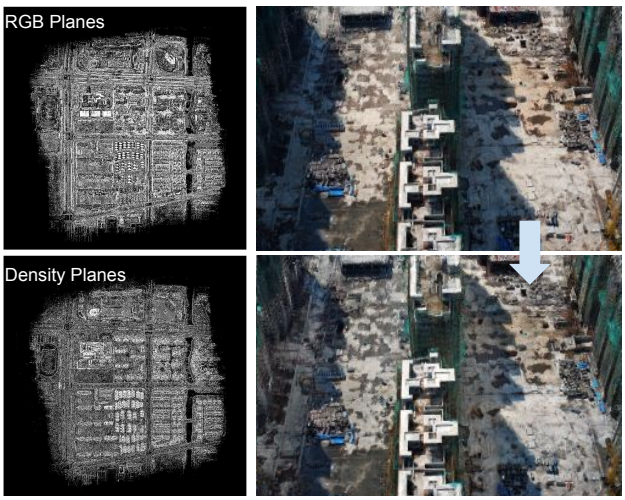

Figure 17. A separate modeling of RGB and density planes can accommodate cases with appearance variations, *e.g.*, the casted shadow of building can be reflected from the RGB plane. In such cases, we can manipulate on the RGB plane only to change the appearance without affecting the object density. The right column shows the affecting of zeroing out one feature component in RGB plane make the ground shadows fade out.

module may also be considered at the current stage. A large model configuration and equipment requirement for a large number of image training can be found in BlockNeRF [13],

which shows a workable strategy to perform training on that data scale. The efficient training pipeline of neural radiance fields, either in NeRF-based or hybrid based formats, still leaves it an open question to explore in the future.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021. 1

[2] 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. 1, 3, 5, 6, 8, 9

[3] Eric Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, S. Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. *ArXiv*, abs/2112.07945, 2021. 4

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022. 2, 3, 4, 6

[5] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12952, 2022. 6

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 4

[7] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. *arXiv preprint arXiv:2104.06405*, 2021. 6

[8] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *CVPR*, 2021. 6

[9] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 2

[10] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv:2201.05989*, Jan. 2022. 2, 3, 4, 6

[11] 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

[12] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *arXiv preprint arXiv:2111.11215*, 2021. 2, 5, 6

[13] 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. *arXiv preprint arXiv:2202.05263*, 2022. 1, 2, 6, 8, 9

[14] Jiaxiang Tang. Torch-ngp: a pytorch implementation of instant-ngp, 2022. https://github.com/ashawkey/torch-ngp. 3

[15] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. *arXiv preprint arXiv:2112.10703*, 2021. 1, 2, 3, 6, 8

[16] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 9

[17] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 6

[18] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Citynerf: Building nerf at city scale. *arXiv preprint arXiv:2112.05504*, 2021. 1, 9

[19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4, 6