XScale-NVS: Cross-Scale Novel View Synthesis with Hash Featurized Manifold

Supplementary Material

1. Additional Dataset Details

Scene	Area (m^2)	#Images	Resolution
TW-Pavilion (Day)	1.3×10^4	10,999	5K/8K
TW-Pavilion (Night)	1.3×10^4	1,687	5K/8K
Lanes & Alleys	4.5×10^5	18,206	5K/8K
The Great Wall (T3)	3×10^6	7,441	5K/8K
The Great Wall (T2)	2×10^6	6,372	5K/8K
The Five Old Peaks	1.5×10^6	3,960	5K/8K
Sandie Spring	3×10^6	5,474	5K/8K

Table 1. Statistics of the GigaNVS dataset.

Our dataset is captured using two drones and a handheld DSLR. The drones we use include a DJI Matrice 300 RTK equipped with a fixed 35mm camera, whose FOV is 63.5° and the image resolution is 5460×8192. The other drone is a DJI Mavic 3 mounted with a fixed 24mm camera, whose FOV is 84° and the image resolution is 3956×5280. The ground photography is captured by a Canon EOS R5 DSLR camera with a 15-30mm lens, and the image resolution is 5464×8192. Since we focus on in-the-wild ultra-large-scale scenes and aim to collect high-quality multi-view imagery of varying scales, it is difficult to finish capturing within a short period to maintain a constant illumination condition. We therefore manually adjust the ISO, white balance, shutter speed, and aperture size of the cameras according to the illumination condition to prioritize the imaging quality.

A detailed statistics of our dataset is listed in Table 1. The scenes we capture typically span areas of several square kilometers. For each scene, we capture between 1,687 and 18,206 images taking several days. We use shared intrinsic parameters for images captured from the same camera, and we use Agisoft Metashape [1] to compute the camera poses, undistort the images, and reconstruct the mesh. We take every 7th photo as our test set for each scene, which approximates a uniform coverage of all view points and scales.

2. Additional Implementation Details

In this section, we further detail the implementations of our model and the training strategies we use, and then present details regarding the comparative evaluations.

Additional Model Details. We use $L = 16$ levels of multi-resolution hash encoding with a coarsest resolution of 16 and a finest resolution of $2^{13} \sim 2^{19}$ depending on the scene scale and the image resolution. We use a maximum number of $N_h = 2^{22}$ hash entries with $Z = 8$ feature channels, following [11]. The mesh is reconstructed using Agisoft Metashape [1] and we set the multisampling rate γ as 2 during rasterization. The only loss function we use is the L1 photometric error. The decoder MLP $\mathcal{M}$ and the manifold deformation MLP ξ consist of 4 and 2 layers respectively, where each layer contains 64 hidden units. The concatenated multi-resolution hash features are consumed through FiLM-conditioning [5, 14]. To compensate for the camera response variations of the real-world captures, we follow [4, 11] to use appearance embedding to modulate the intermediate features of the decoder $\mathcal{M}$.

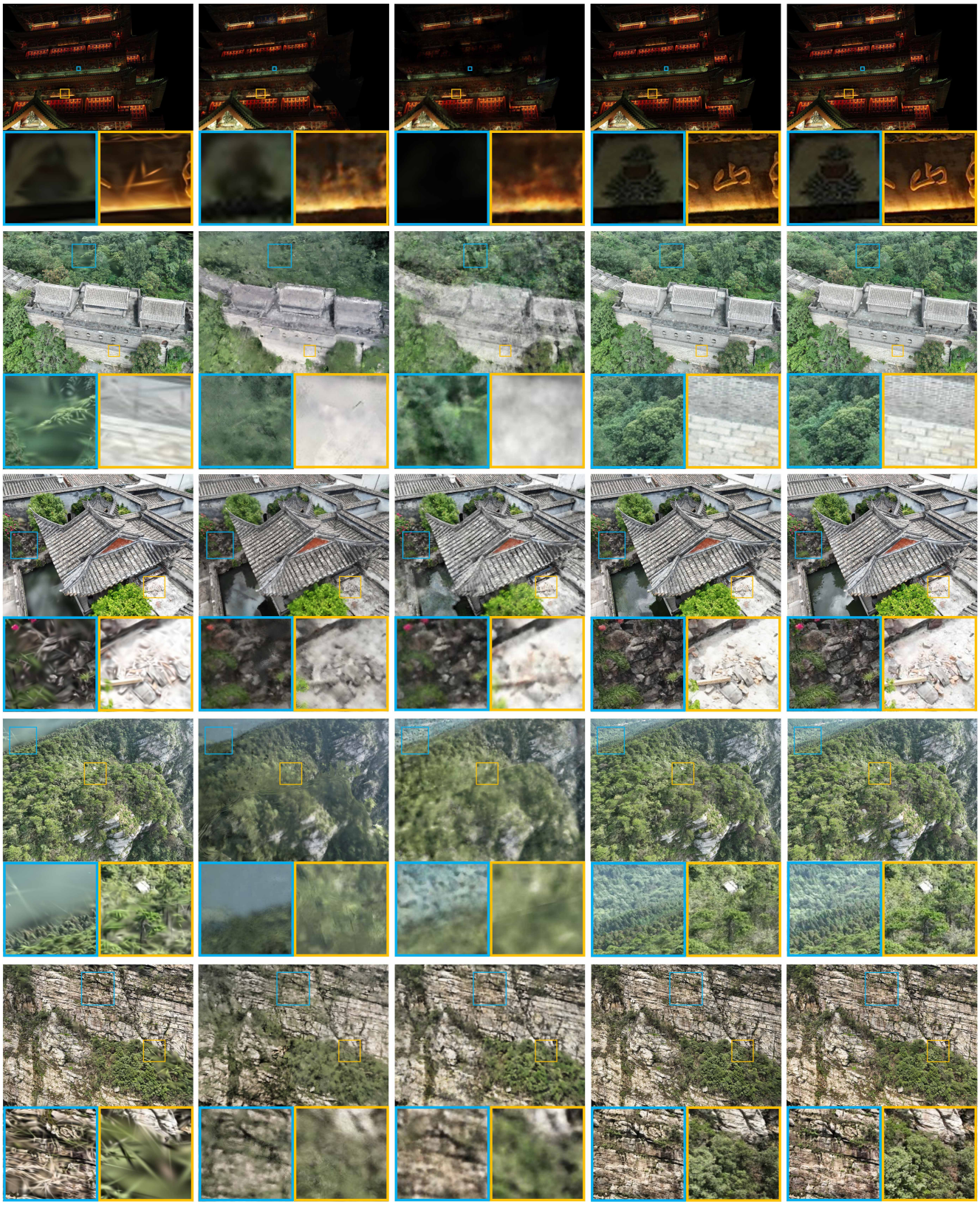
Additional Training Details. To enable a large batch size of camera views during training, we pre-cache the rasterized z-buffer of the reconstructed mesh for each training view. At each training iteration, we sample a random set of camera views and then a random batch of pixel rays from the caches for each sampled view. We use a batch size of 16 for the random views and a batch size of 16384 for the random rays. We train our model using the Adam optimizer [9] with a learning rate of 10^{-4} for 1000 epochs, randomly iterating through all training views to complete a single epoch. The training can be done with a single NVIDIA RTX3090 GPU.

Additional Evaluation Details. Similar to [18], we only focus on the reconstruction of foreground regions. For a fair comparison to other baselines [8, 11, 13], we leverage the mesh to explicitly benefit their performance by providing not only the geometric guidance but also foreground mask priors. Specifically, we project the mesh to fill the background regions of all input images as pure white, i.e., $[255, 255, 255]$, and set the background colour in [8, 11, 13] explicitly as the same white colour during training. We find this simple modification effectively bypasses the difficult background modelling task and significantly boosts the foreground rendering quality of these methods. We also use the rasterized foreground mask to calculate an additional BCE loss as done in [19] to further regularize the foreground geometry of [8, 11, 13]. Before computing the evaluation metrics, we apply the same rasterized mask to the renderings of all competing methods and the corresponding ground truth image. We fairly evaluate our method and all other baselines using the same NVIDIA RTX3090 GPU.

3. Additional Results on GigaNVS

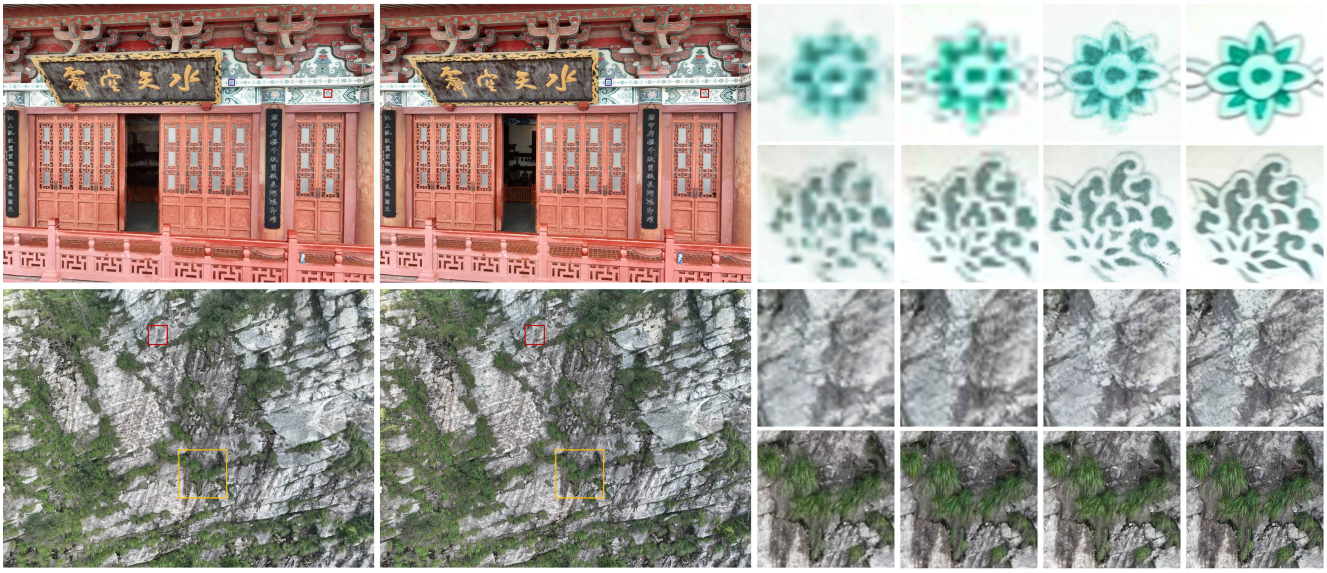
In this section, we present additional qualitative results on the challenging GigaNVS dataset.

1K-resolution Novel View Synthesis. We first compare against prior state-of-the-art approaches [8, 11, 13] on novel view synthesis with 1K resolution images as inputs and



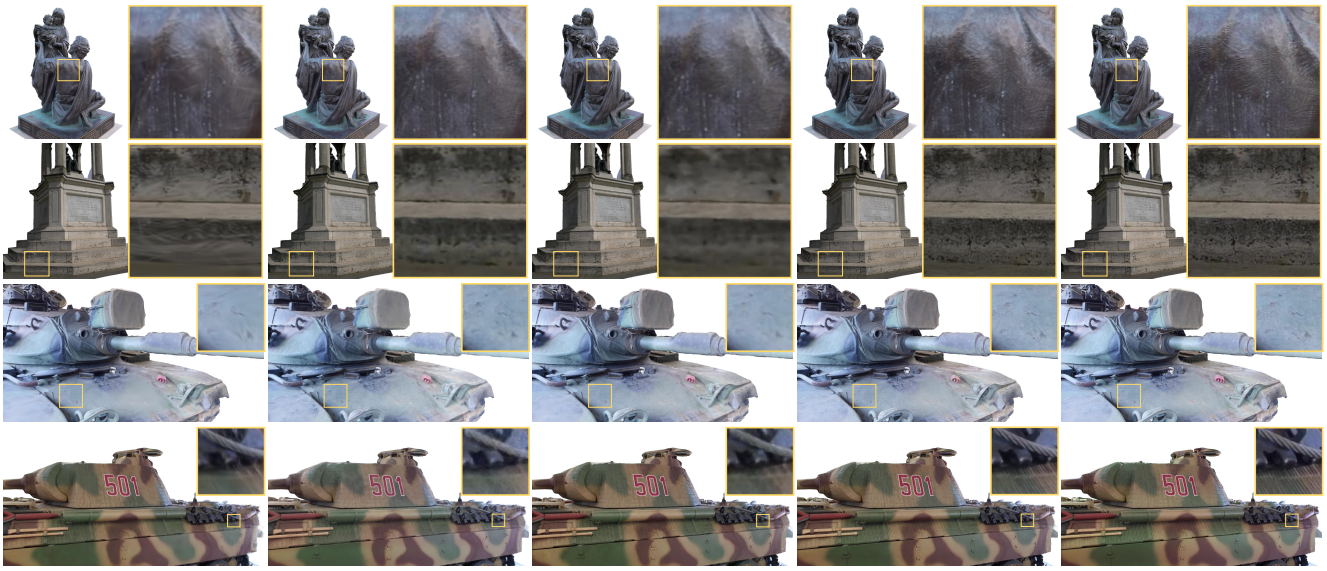
(a) 3D Gaussian Splatting [8] (b) Neuralangelo [11] (c) iNGP [13] (d) **Ours** (e) Ground Truth Image

Figure 2. Qualitative comparisons of 1K-resolution novel view synthesis on the `GigaNVS` dataset. Compared to [8, 11, 13], our hash featurized manifold represents cross-scale contents with superior fidelity and effectively preserves the rich details of the original imagery.



Ours (5K) Ground Truth Image (5K) Ours (1K) GT (1K) Ours (5K) GT (5K)

Figure 3. Qualitative results of 5K-resolution novel view synthesis on the GigaNVS dataset. Our method fully exploits the input resolution and demonstrates strong scalability towards high-resolution neural rendering. Please zoom-in to see the high-resolution details.



(a) 3D Gaussian Splatting [8] (b) Neuralangelo [11] (c) iNGP [13] (d) **Ours** (e) Ground Truth Image

Figure 4. Visual comparisons on the Tanks&Temples dataset [10]. Our method excels at recovering local intricate details yet generally performs on par with state-of-the-art neural rendering methods [8, 11, 13] on small-scale scenes.

outputs. The corresponding qualitative results are shown in Fig. 2. Our hash featurized manifold synthesizes novel views with significantly superior fidelity compared to other baselines, effectively recovering realistic cross-scale details of the relievos, plants, tiles, and stones, reflecting unprecedentedly rich contents in real-world large-scale scenes.

5K-resolution Novel View Synthesis. To take a step further, our method can operate effectively on high-resolution inputs of 5K or 8K due to the expressivity and efficiency, whereas existing baselines are not scalable towards this high-resolution setup, since they still struggle to render the details at 1K resolution (see Fig. 2 (a), (b), (c)) and suffer from memory issues. In Fig. 3, we provide our 5K novel view synthesis results where our model is trained using the original 5K resolution imagery. The results suggest that our method can successfully exploit the rich information brought by the high-resolution imagery and represent scenes using approximately the input-level resolution.

Scene	Meta representation[18]			3D Gaussian Splatting[8]			Neuralangelo[11]			iNGP[13]			**Ours**		
	PSNR ↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM ↑	LPIPS↓
Family	27.92	0.913	0.035	28.54	0.920	0.035	28.02	0.922	0.033	28.49	0.919	0.036	28.15	0.915	0.032
Francis	33.11	0.958	0.048	35.77	0.969	0.050	30.54	0.951	0.056	31.85	0.950	0.067	35.32	0.973	0.024
Horse	35.22	0.979	0.027	34.88	0.986	0.017	34.62	0.984	0.016	35.16	0.984	0.018	36.07	0.984	0.017
M60	27.36	0.891	0.153	29.41	0.918	0.133	27.03	0.886	0.154	28.82	0.907	0.132	28.07	0.899	0.115
Lighthouse	23.92	0.811	0.201	22.49	0.825	0.207	23.96	0.836	0.191	24.47	0.842	0.187	26.47	0.866	0.127
Panther	28.57	0.889	0.162	26.83	0.869	0.210	27.87	0.901	0.155	29.70	0.908	0.146	29.05	0.903	0.112
Train	22.12	0.803	0.207	22.44	0.834	0.206	19.85	0.806	0.195	22.21	0.826	0.184	24.50	0.863	0.124
Mean	28.32	0.892	0.119	28.62	0.903	0.123	27.41	0.898	0.114	28.67	0.905	0.110	29.66	0.914	0.079

Table 2. Quantitative comparisons on seven scenes from the Tanks&Temples dataset.

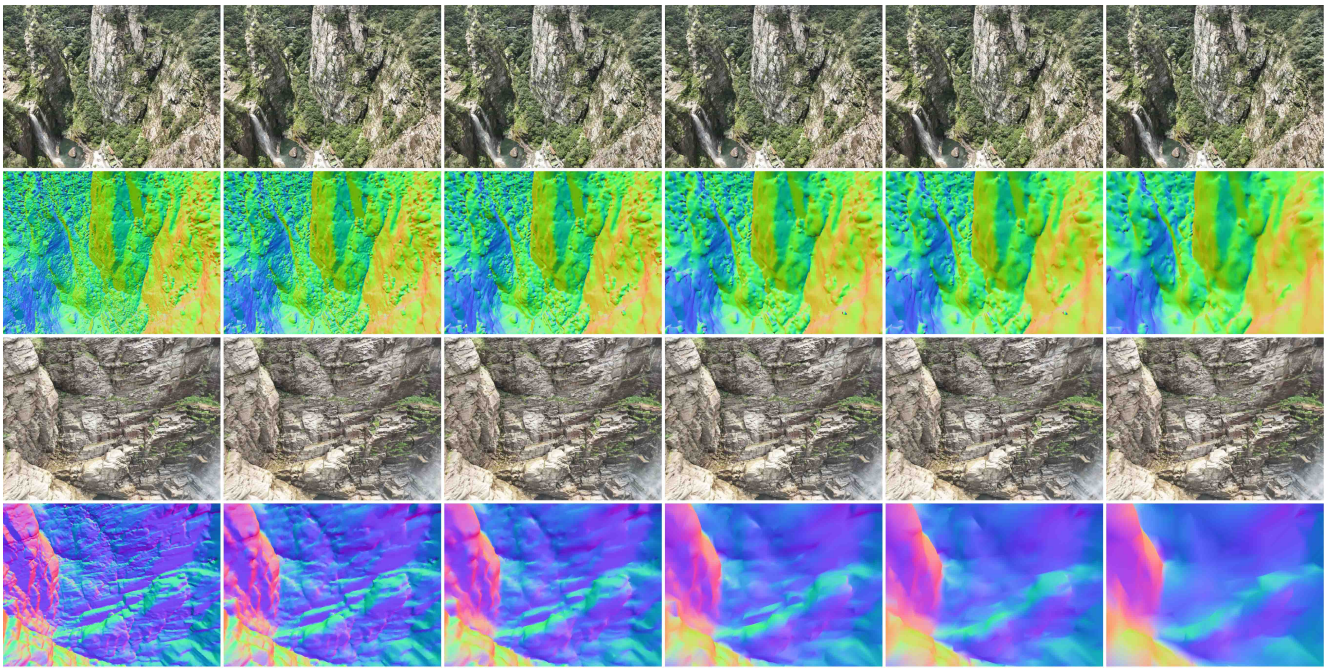


Figure 5. Qualitative results with different mesh resolutions on the Sandie Spring scene of GigaNVS dataset, where our novel view RGB renderings and the corresponding mesh normals are visualized. The first two rows are macro-scale renderings from a distance and the last two rows are micro-scale renderings at the close-up. Our method robustly recovers scene contents at drastically different scales when down-sampling the mesh to lower resolutions.

As observed, our 5K rendering delivers significantly richer details compared to the ground truth image of 1K resolution, indicating a continuously better perceptual quality proportional to the input resolution, which hash rarely been verified in existing in-the-wild neural rendering methods. Therefore, the proposed hash featurized manifold representation shows great potential to bridge the resolution gap between sensation and reconstruction.

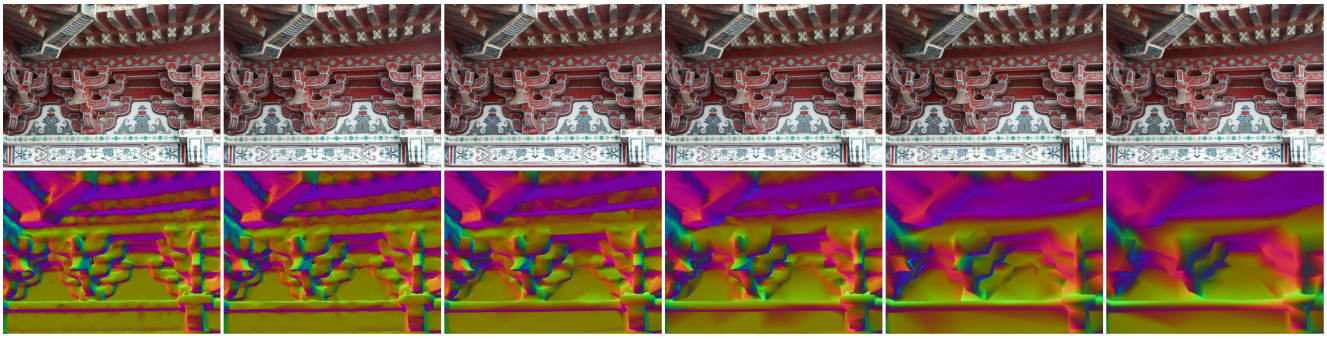
4. Additional Results on Tanks&Temples

We now present more detailed results on the public Tanks&Temples dataset [10]. Following [18], we select seven real-world outdoor scenes from this dataset to benchmark our performance on small-scale scenarios. In Fig. 4, we show visual comparisons of the competing methods on four representative scenes. In general, our method performs on par with state-of-the-art baselines [8, 11, 13] while only showing slight improvements at fine-grained details, with the micro-scale intricate textures on the sculpture and tanks better recovered. The quantitative evaluations are reported in Table 2. Our method yields a ∼1dB improvement in PSNR and a 28% lower LPIPS relative to the second best method iNGP [13]. Therefore, hash featurized manifold can serve as a general purpose neural scene representation offering superior expressivity and efficiency.

5. Additional Ablations

In this section, we provide additional ablations to conduct a more comprehensive demonstration of our design.

Featurization Enhancements. The surface multisampling and manifold deformation serve as optional enhancements upon the hash featurized manifold representation, which



Ori. (10M) 5M 2M 1M 0.5M 0.2M

Figure 6. Qualitative results with different mesh resolutions on the TW-Pavilion (Day) scene of GigaNVS dataset. First row: our novel view RGB renderings; Second row: the corresponding rendered normals of the input mesh.

can help to better describe the cross-scale scene details. As shown in Fig. 7, our method can subsequently render sharper details after introducing these designs.

Full w/o MD w/ M w/ MD (Full) GT

Figure 7. Qualitative comparisons of novel view synthesis with different levels of featurization enhancements. Sharp details can be effectively recovered with surface multisampling (M) and manifold deformation (D).

Method	#Primitives	PSNR↑	SSIM↑	LPIPS↓
Ours	16M	17.61	0.724	0.206
Ours	8M	17.58	0.722	0.210
Ours	4M	17.52	0.714	0.218
Ours	2M	17.47	0.704	0.226
Ours	1.6M	17.44	0.696	0.233
Ours	1M	17.35	0.680	0.242
Ours	0.5M	17.14	0.645	0.264
3D Gaussian Splatting [8]	3M	17.25	0.670	0.380

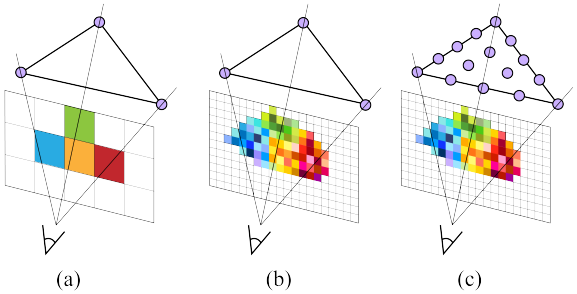
Table 3. Mesh resolution ablation on the Sandie Spring scene.

Method	#Primitives	PSNR↑	SSIM↑	LPIPS↓
Ours	10M	23.02	0.786	0.113
Ours	5M	22.94	0.785	0.112
Ours	2M	22.87	0.783	0.112
Ours	1M	22.84	0.782	0.114
Ours	0.5M	22.44	0.767	0.125
Ours	0.2M	22.07	0.749	0.140
3D Gaussian Splatting [8]	3M	21.52	0.752	0.225

Table 4. Mesh resolution ablation on the TW-Pavilion (Day) scene.

Mesh Resolution. We now present further demonstrations regarding the robustness of our representation w.r.t. the input mesh resolution. The ablations are performed on two challenging scenes (Sandie Spring and TW-Pavilion (Day)) from the GigaNVS dataset. The quantitative evaluations on the two scenes are reported in Table 3 and Table 4 respectively, where #Primitives denote the amount of triangle facets of our input mesh or the amount of gaussians used in [8], and 'M' denotes a million. Remarkably, our representation shows strong robustness w.r.t. the mesh resolution and outperforms 3DGS [8] with much lower discretization resolution. Unlike [8], our hash featurized manifold is essentially a neural-based representation which does not explicitly store per-primitive feature descriptors (and also the gradients during optimization), potentially being expressive enough independent of the discretization resolution. On the other hand, our method can leverage a higher resolution mesh in a memory efficient way, where the mesh resolution only affects the memory for rasterization. For example, when rendering a 1K image, 3DGS [8] consumes 18G memory with 3 million gaussians, whereas our method only takes up 15G memory using a mesh with 10 million facets.

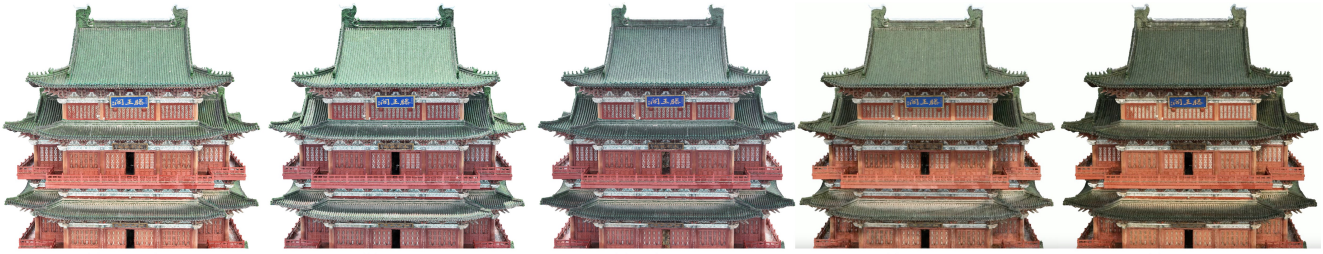
6. Necessity for Improving Feature Resolution



(a) (b) (c)

Figure 8. Illustration of the featuremetric resolution issue.

In contrast to early neural representations [12, 19] that use a large global MLP to encode the entire scene, recent works [6, 11, 13, 16] tend to use a hybrid representation, i.e., a combination of the explicit volumetric featurization with a light-weight MLP decoder. In this way, the repre-



(a) Morning (b) Noon (c) Afternoon (d) Sunset1 (e) Sunset2

Figure 9. Qualitative relighting results on the TW-Pavilion (Day) scene when varying the lighting embedding.

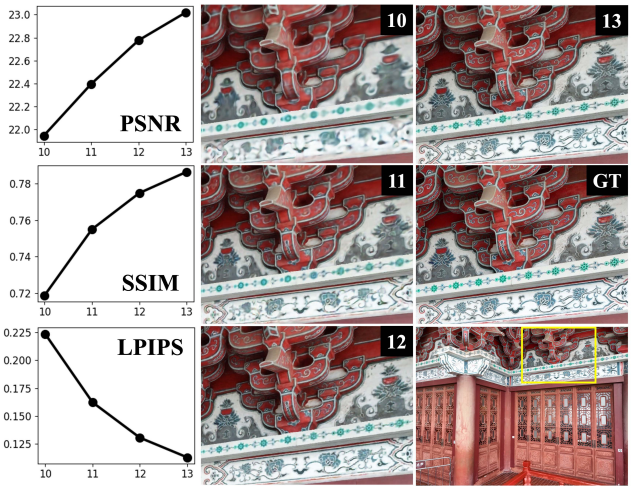


Figure 10. Quantitative and qualitative evaluations with different featuremetric resolutions ($2^{10} \sim 2^{13}$).

sentational power can be effectively shifted from the MLP to local spatial features, which enables superior reconstruction quality and efficiency.

In light of this, we hold that the spatial resolution of the underlying local features is strongly correlated to the expressivity of the representation, and it is essential to allocate local features with a sufficient resolution according to the scene complexity, scale variation, and the image resolution. We now give an intuitive illustration of this observation in Fig. 8. Assuming we have a coarse mesh with per-vertex learnable features, and we first consider small-scale simple scenes which are often captured at a medium distance with relatively low-resolution. As shown in Fig. 8 (a), a single surface primitive, i.e., a triangle facet in this case, only contains the information of very few pixels on the captured image. In this case, it is easy to accurately express the information of the pixels using the three feature descriptors at the vertices. However, for large-scale complex scenes with high-resolution or close-up captures, as illustrated in Fig. 8 (b), the pixel information covered by the same surface primitive is significantly enriched, making it difficult to fully describe the high-frequency pixel contents only by the vertex-defined features. In this case, the representation will produce blurry renderings. Note that representations based on other types of surface primitives like UV texels [17, 18], points [2, 15], or Gaussians [8] also share the same intuition. Therefore, to tackle this issue, we should increase the featuremetric resolution to match the richness of pixels, which may significantly exceed the resolution of surface primitives, as shown in Fig. 8 (c).

To verify the above intuition, we conduct ablations by varying the conceptually expressed finest resolution of hash encoding while keeping the hash table size fixed. The experiments are performed on the TW-Pavilion (Day) scene with 1K resolution. As shown in Fig. 10, increasing the finest resolution from 2^{10} to 2^{13} improves the quantitative metrics and enables more detailed view synthesis.

7. In-the-wild Relighting on `GigaNVS`

Our method models the view-dependent colour $\boldsymbol{c} \in \mathbb{R}^3$ with an MLP-based implicit function $\mathcal{M} : \mathbb{R}^{LZ} \times \mathbb{R}^3 \mapsto \mathbb{R}^3$, which takes as inputs the queried hash feature $\mathcal{F}_s^{\mathcal{H}}(\boldsymbol{x}_s) \in \mathbb{R}^{LZ}$ and the view direction vector $\boldsymbol{d} \in \mathbb{R}^3$, i.e.,

$$\boldsymbol{c} = \mathcal{M}(\mathcal{F}_s^{\mathcal{H}}(\boldsymbol{x}_s), \boldsymbol{d}). \tag{1}$$

We can easily extend this formulation to implicitly account for the global illumination by introducing an additional lighting embedding $\boldsymbol{e}_l^{(k)} \in \mathbb{R}^E$:

$$\boldsymbol{c} = \mathcal{M}(\mathcal{F}_s^{\mathcal{H}}(\boldsymbol{x}_s), \boldsymbol{d}, \boldsymbol{e}_l^{(k)}). \tag{2}$$

Specifically, we leverage the timestamps from the metadata to partition the collected imagery into K subsets, where each subset has a very different illumination, and images within a subset share nearly the same illumination condition. We then assign a learnable lighting embedding $\boldsymbol{e}_l^{(\cdot)}$ for each subset, and so we have K lighting embeddings $\{\boldsymbol{e}_l^{(k)}\}_{k=1}^K$ in total, representing K different scene illuminations. Given a specific training view from the k-th subset, we only back-propagate the gradients to the related lighting embedding $\boldsymbol{e}_l^{(k)}$ during training. In this way, we can relight the rendering by varying the lighting embedding passed to the implicit function $\mathcal{M}$, and unseen illuminations can be synthesized by interpolating between an arbitrary pair of the

latent embeddings:

$$\boldsymbol{e}_l^{(i,j,\tau)} = \tau \boldsymbol{e}_l^{(i)} + (1-\tau)\boldsymbol{e}_l^{(j)}, \tag{3}$$

where $\tau \in [0, 1)$ is a scalar value denoting the interpolation weight.

In Fig. 9, we show our relighting results at a novel view point, where we empirically set the dimensionality of the lighting embeddings as $E = 8$. The proposed lighting embedding effectively reasons about the global illumination effects and further improves the rendering fidelity of real-world large-scale scenes. Please refer to the supplementary video to see the smoothly interpolated illuminations.

8. Visualizations of Surface Ambiguity

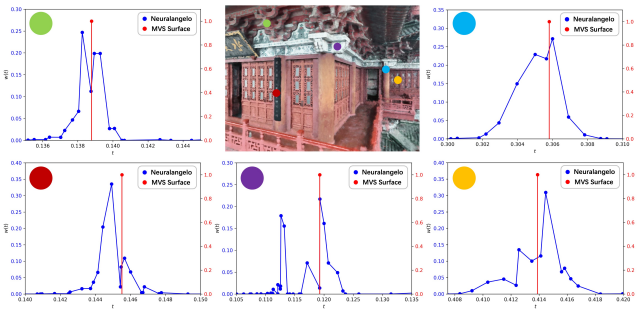


Figure 11. Visualization of the volume rendering weight distributions of Neuralangelo [11]. The middle image is the final RGB rendering. In each surrounding sub-figure, we plot the weight $w(t)$ for each individual sample at certain normalized distance t traced from one of the labeled pixels. The red vertical lines denote the surface locations calculated by mesh rasterization. For brevity, we only show important samples near the surface.

In Fig. 11, we take Neuralangelo [11] as an example and visualize the weight distributions (the blue dot curves in the surrounding sub-figures) along the labeled pixel rays from the middle rendered image. Note that we use the mesh surface (the red vertical lines) reconstructed from MVS to densely supervise the SDF field in 3D [7] throughout the experiments. However, the resulting weight distributions still scatter across a thick region, where most samples are in lack of multi-view consistency yet assigned with high volume rendering weight. As a result, these non-surface samples induce colour ambiguity and propagate inconsistent colour gradients to the surface sample, thus leading to excessive blurries that can be found in the rendering.

9. Limitations and Future Work

Given that our representation is essentially a surface-based featurization, the incompleteness of the mesh reconstruction (e.g., far-away background regions or the holes caused by erroneous correspondence) is inevitably reflected in the rendering. However, since our primary focus is real-world large-scale scene reconstruction, where highly-specular surfaces or large textureless regions are not commonly witnessed, the mesh reconstructed by existing MVS methods is generally reliable enough. Regarding the background issue, we can easily address it by incorporating another volumetric neural field, similar to [3, 4, 11]. Except for the incompleteness, remarkably, our method is robust to other topological or geometric degradations without the need to explicitly fix the mesh – see the '$\leq$1M' results in Fig. 5 and 6, where our method effectively represents the missing details of the suboptimal geometry.

Our future work is to explore scalable differentiable rendering pipelines for more flexible optimization of the topology, and to investigate physically-based inverse rendering techniques for the explicit acquisition of the underlying intrinsic properties, which is indispensable for downstream industrial applications.

References

[1] Agisoft LLC. Agisoft metashape, 2021. 1

[2] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 696–712. Springer, 2020. 6

[3] 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. 7

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023. 1, 7

[5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 1

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 5

[7] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35:3403–3416, 2022. 7

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 1, 2, 3, 4, 5, 6

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[10] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 3, 4

[11] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 1, 2, 3, 4, 5, 7

[12] 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. 5

[13] 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, 2, 3, 4, 5

[14] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 1

[15] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics (ToG)*, 41(4):1–14, 2022. 6

[16] 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. 5

[17] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 6

[18] G. Wang, J. Zhang, K. Zhang, R. Huang, and L. Fang. Giganticnvs: Gigapixel large-scale neural rendering with implicit meta-deformed manifold. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (01):1–15, 2023. 1, 4, 6

[19] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 1, 5