Supplementary Material:
Patch-based 3D Natural Scene Generation from a Single Example

Weiyu Li[1] Xuelin Chen[2] Jue Wang[2] Baoquan Chen[3]
[1] Shandong University [2] Tencent AI Lab [3] Peking University

Figure 1. Artistic imagery created with various 3D scenes generated by our method (background sky post-added). The original exemplar scenes for generating these results are: (from left to right and top to bottom): The Vast Land [35], Heal Mountain [17], Devil's Tower ©2022 Google, Callanish [24], Meteora ©2022 Google. Green Island [18],

A. More Visual Results

More artistic pieces, that are created with high-quality and diverse general natural scenes generated by our method, are shown in Figure 1. Moreover, in Figure 2, we also visualize the underlying high-quality and diverse geometry of more generated samples. Figure 15, 16 presents more samples generated with our method.

In addition to more dynamic 3D viewing of a large collection of generated samples presented in the supplementary video, please also see the anonymous project website http://wyysf-98.github.io/Sin3DGen for a more immersive view into our 3D results.

B. Implementation Details

All synthesis results presented in this paper share the following default setting, unless specified. *We will release the code for reproducing the results presented in this paper, upon the publication of this work.*

Random Generation. By default, the synthesized scene S shares the same bounding box $\mathbb{B}$ with E in the random synthesis task. Each scene is located inside a cuboid, of which the aspect ratio varies according to different exemplars. In Table 1, we list the final resolution of S_N in the pyramidal generation framework for each exemplar scene. At the N-th scale in the multi-scale framework, the resolution along the maximum dimension of S_N is set to 121, considering the trade-off between the quality and computational efficiency of the generation. The resolution along the maximum dimension of the higher-resolution E^{high} is 512. The scaling factor between consecutive scales in the pyramid is $r = 4/3$, and the coarsest resolution is 16. We use $N = 7$ in the pyramid, which results in 8 scales in total. The patch size at all scales is set to $p = 5$. We set the number of PCA components to 3, the truncate scale of SDF $t = 3 \times w$, where w is the voxel size. The weight of the appearance feature is $w_a = 0.5$, the completeness trade-off weight $\alpha = 0.01$, and the initial noise $\sigma = 0.5$. At coarser scales ($n < 5$), exact NNF is applied $T_e = 10$ times, which means the value-based NNF is performed with $T_e - 1 = 9$ times and followed by one mapping-based NNF search; At the finest scale ($n >= 5$), approximate NNF via PatchMatch is performed $T_a = 2$ times. The "jump flood" radius is 8, and the random search radius is fit to the max resolution of the current exemplar.

Applications. In contrast to the random synthesis task, the σ for noise used in all applications is set to 0. More specifically:

- 1) **Retargeting:** The goal is to resize a 3D scene to a target size (typically of a different aspect ratio), while maintaining the local patches in the exemplar. We simply set the resolution of S_N to the target size, and the resolution of $S_0, ..., S_{N-1}$ is adapted accordingly with the default scaling factor r.
- 2) **Editing:** Users can manipulate on a 3D proxy, which can be the underlining mapping field or mesh, for editing an exemplar or generated scene, such as removal, duplication, and modification. The manually manipulated proxy is then converted and fed as the initial guess at the coarsest scale for synthesizing the final scene. As editing the 3D scene requires more meticulous 3D interaction, we set the resolution at the coarsest scale to a higher value (resolution along the max dimension is 28), and use 6 scales in total. We perform the exact NNF at the first 3 scales, followed by 3 finer scales with the approximate NNF.
- 3) **Structural analogies:** Given two scenes A and B, we create a scene with the patch distribution of A, but which is structurally aligned with B. This is realized by using the exemplar pyramid of A, and an identity mapping as the initial guess, but replacing $\hat{E}_0(\tilde{S}_0)$ with the transformed features in B, and vice versa. As the content of the generated scene at the coarsest scale is already specified by $\hat{E}_0(\tilde{S}_0)$, the pyramidal generation starts with a higher-resolution scale (51 voxels along the max dimension), finishes the generation with 4 scales in total, and performs exact NNF in the first scale.
- 4) **Re-decoration:** Trivially, we do not need to re-synthesize the scene in the re-decoration application. Given an already generated scene, a novel scene can be obtained by simply remapping the coordinate-based synthesis result to an exemplar of different appearance.

C. Datasets

We collected a rich variety of 3D scene models to examine the performance of our method on random scene generation, ranging from rocks to plants, sculptures, landscapes, terrains, artistic scenes, etc. For each 3D scene model, we render 200 images at the resolution 1024 × 1024, with cameras distributed on a sphere in Blender [10]. Then the Plenoxels-based exemplar pyramid is obtained via coarse-to-fine training on these images. Notably, in Figure 11, we also demonstrate our method on real-images collected from a real-world scenic site. To this end, we collect 300 images with the resolution 1280 x 720 from Google Earth Studio [1], where we can manually specify cameras for simulating a drone programmed to fly over a scenic spot, for training the Plenoxels-based exemplar. Specifically, we move the camera in a spiral motion and gradually elevate

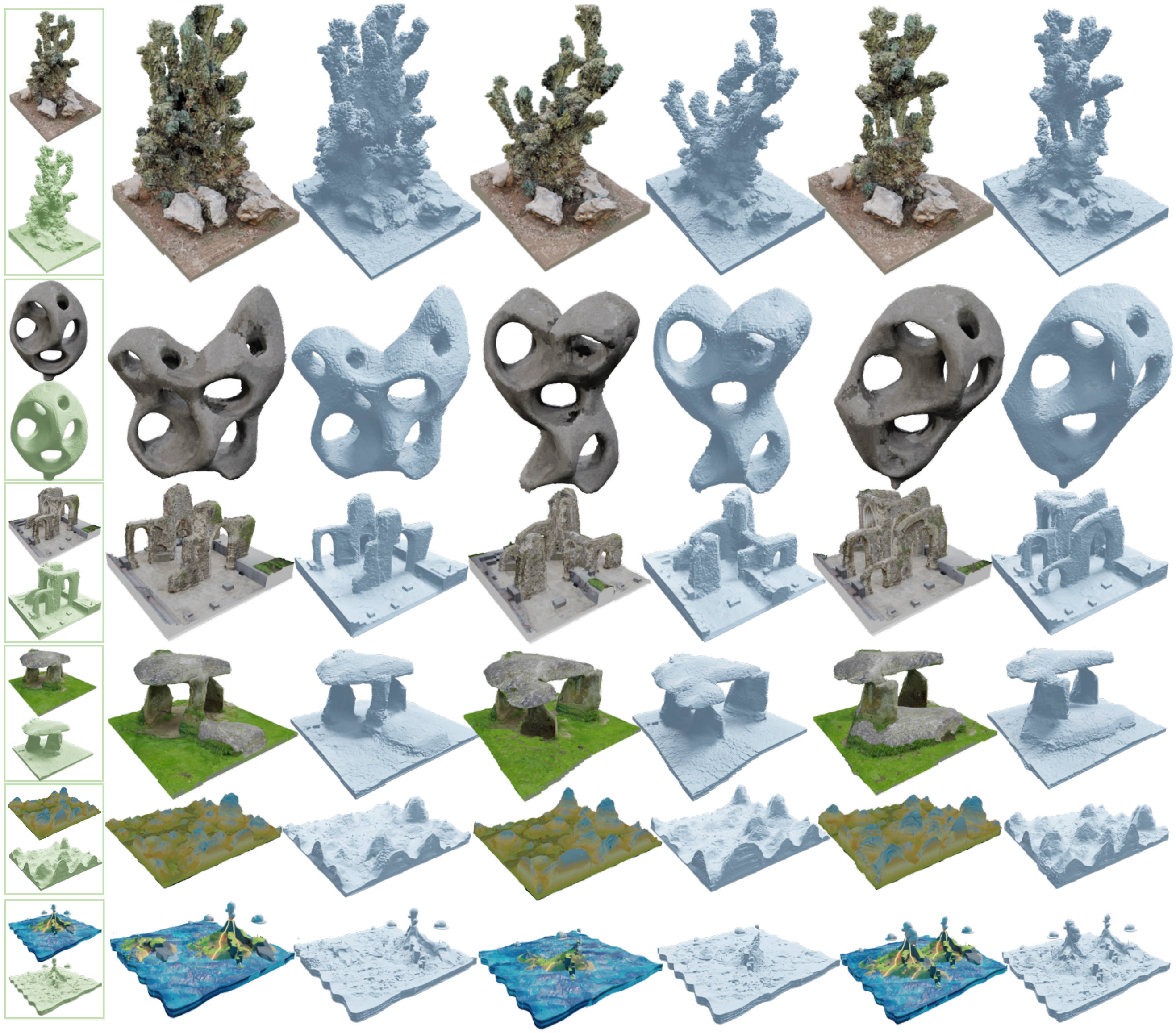

Figure 2. Visualization of the high-quality geometries generated by our methods. The original exemplar scenes for generating these results are: (from top to bottom) Cactus Cereus [25], Stone Sculpture [3], St Alphage [6], Spinsters Rock [4], The Vast Land [35], Volcano Island Lowpoly [5].

the camera from a high altitude to a low altitude. Then, we use COLMAP [31, 32] to estimate the camera parameters. More details can be found in the video. Figure 13 presents the visuals of all exemplars used in this paper.

D. More Analysis

In general, our method is robust to varying hyper-parameters to some extent. We shall show the effects of using different parameters in the following.

Effects of Different Noise z_0. In Figure 3, we show the results obtained by different initial guesses, i.e., the identity mapping shuffled with different noises, at the coarsest level.

Effects of Different w_a. Empirically, the trade-off parameter w_a for balancing the appearance and geometry feature is set to 0.5 by default. While we have shown this setting yields robust and high-quality generation, we shall also demonstrate the effects of varying weights. Generally, in Figure 4, we can see that our method is robust to varying w_a

Table 1. Resolution configuration for figures in the **main paper**.

Figure	Data	Resolution of S_N
Fig. 1	Cactus Cereus [25]	$92 \times 108 \times 121$
Fig. 2 & Fig. 5	Green Island [18]	$121 \times 121 \times 47$
Fig. 3 & Fig. 5	St Alphage [6]	$121 \times 121 \times 92$
Fig. 5	Calda House [7]	$121 \times 121 \times 71$
	Callanish [24]	$121 \times 121 \times 47$
	Stone Arch [36]	$121 \times 51 \times 71$
	Desert Lowpoly [12]	$121 \times 121 \times 92$
	Meteora ©2022 Google	$121 \times 121 \times 47$
	Spinsters Rock [4]	$121 \times 121 \times 84$
	Stone Sculpture [3]	$108 \times 84 \times 121$
	Volcano Island Lowpoly [5]	$121 \times 121 \times 71$
	The Vast Land [35]	$121 \times 121 \times 47$
Fig. 7	Mountain with Lakes [40]	$121 \times 121 \times 72$
Fig. 10	Autumn Camping [21]	$121 \times 99 \times 72$
	Winter Camping [23]	$121 \times 99 \times 72$
Fig. 5 & Fig. 10	Devil's Tower ©2022 Google	$121 \times 121 \times 63$
	Cactus Saguaro [13]	$71 \times 72 \times 121$
	Camping Lowpoly [22]	$121 \times 99 \times 72$
	Mountain [29]	$121 \times 121 \times 84$
	Stylized Cactus [30]	$121 \times 121 \times 121$

Table 2. Resolution configuration for *high-resolution synthesis* and *the retargeting application* in the **main paper**.

Figure	Exemplar	Resolution of S_N
Fig. 6	The Vast Land [35]	$288 \times 288 \times 112$
Fig. 10	Cactus Cereus [25]	$92 \times 108 \times 152$
	Stone Arch [36]	$242 \times 51 \times 71$
	Tiny Castle [9]	$121 \times 63 \times 237$
	Train Wagon [2]	$47 \times 182 \times 47$

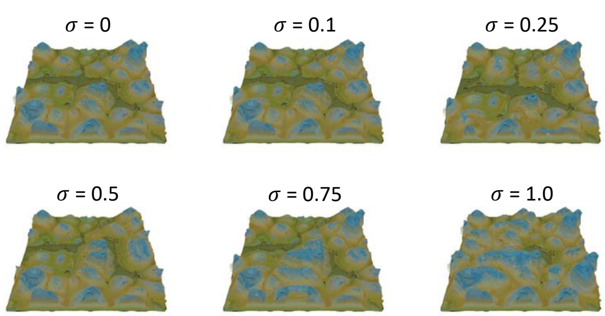

Figure 3. Effects of different initial noise. Intuitively, noise with smaller σ values leads to more similar scenes to the exemplar, while larger values result in more diverse ones.

to some extent. However, setting w_a to an extremely small value, which pays much attention to the geometry feature, will lead to inconsistent appearance and artifacts. On the other end, if we put all emphasis on the geometry feature, the synthesis fails and produces empty scenes.

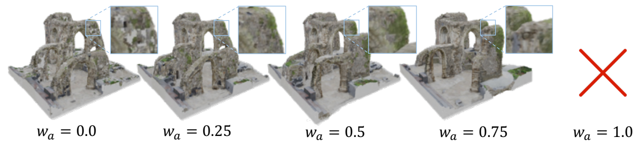

Figure 4. Effects of varying w_a. While w_a around 0.5 produces relatively stable results, extreme values introduce visual inconsistency and artifacts (see left w_a), or even fail (see right $w_a = 1.0$).

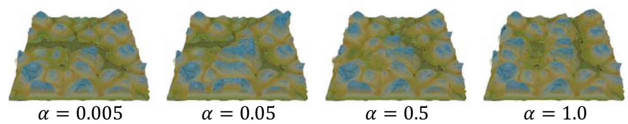

Figure 5. Effects of varying α. α serves as a coarse control knob for visual completeness. The river becomes shorter, suggesting lower visual completeness, as α increases.

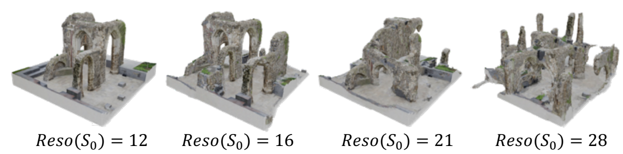

Figure 6. Effects of different resolutions of S_0. Lower-resolution S_0 (larger receptive field at coarse scale) results in less structural diversity, producing almost identical to the exemplar. On the contrary, with smaller receptive fields at the coarsest scale, the global arrangement can not be well preserved (see messy structures on the right).

Effects of Different α. Figure 5 presents the effects of varying α for different levels of visual completeness. Nevertheless, we also found the degrees of such control may not be always perfectly explicit, which we also observed with [14].

Effects of Different Resolution for S_0 at the Coarsest Scale. Given a fixed patch size, which is $p = 5$ in our work, a larger resolution at the coarsest scale suggests a smaller effective receptive field (the same concept as in convolutional neural networks) and less-considered global layouts at the coarser scales, and vice versa. In our work, we by default use the setting where the patch at the coarsest scale captures 1/3 of the content in the exemplar, balancing the local diversity and global layout. In Figure 6, we also show the impact of varying resolutions at the coarsest scale, that capture contents of different sizes in the generation.

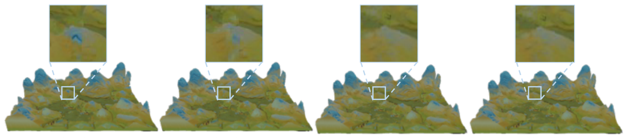

Figure 7. Effects of different resolutions of S_N. The overall visual synthesis is stable at a resolution of 121. Some minor aliasing can be found in lower resolution (left), with a larger scale for synthesis, more details can be obtained.

Figure 8. Effects of different downscale ratios r. As r increasing, fine structures, e.g., the arch doors, gradually disappears, producing bulky structures and less diversity in each instance.

Figure 9. Effects of different truncate scales t, where w is the current voxel size at each scale. A small t will lead to separated fragments due to the loss of geometry information. On the contrary, a very larger truncate scale will blur the geometric feature and only synthesize instances with a coarse shape.

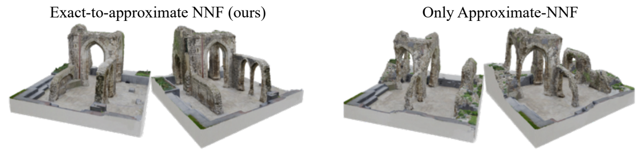

Figure 10. Approximate NNF-only generation. See the distorted arches occurred in the results on the right.

Effects of Different Resolution of S_N at the Finest Scale. The synthesis at finer scales only considers visual coherence and adds local details. We have shown that synthesizing with a maximum resolution 121 by default in the pyramid is sufficient in most cases for the trade-off between quality and efficiency. Moreover, in Figure 7, we show that using higher resolutions for S_N only leads to negligible visual gains. Besides, we also observed that as we use approximate NNF at finer scales, the inaccurate NNF search may introduce some wrong patches and lead to performance degradation in the generation.

Effects of Different Downscale Ratio r. The downscale ratio r used for building the pyramid affects the transition between scales. As the ratio increases, the transition of the generation between scales becomes more inconsistent and unstable due to large gaps between consecutive scales, leading to the loss of fine structures and less diversity as shown in Figure 8.

Effects of the Truncated Scale t. The truncated scale t controls the range of geometric features we keep for patching matching. Smaller truncate scales only consider information near to surface and degrade to the occupancy field, which may produce many tiny pieces and incomplete instances, while larger values lead to blurry results. Figure 9 presents the visual results.

Only Exact or Approximate NNF-3D. The mix use of exact NNF and approximate NNF in our framework has shown the efficacy and efficiency in 3D generation. Using only exact NNF would quickly lead to prohibitive computational cost and prevent us from synthesizing high-resolution results. See Table 3 for the detailed computation overhead. On the other end, only using approximate NNF all the time will harm the generation, producing distorted results, as the approximate NNF is inaccurate. In Figure 9, we show the visual results when only using approximate NNF-3D.

E. More Experiments

Working with Unbounded Scenes. Benefiting from using Plenoxels, which trains on 2D images, for representing the input scene, our method can also work on images collected from a *real-world unbounded* scene. To this end, we use COLMAP [31, 32] to estimate the camera parameters, and model the background using an implicit neural network, similar to NeRF++ [39]. Figure 11 presents the results, more visual details can be found in the video. Note that, existing NeRF-based models often struggle in handling "unbounded" real-world scenes, and disentangling the foreground and background. Nevertheless, some works [8, 20] attempt to tackle these problems, showing promising results. We believe these methods can help boost the performance of our method on more real-world scenes, which, however, is not in the scope of this work and stimulates future research.

Computational Overhead. In Table 3, we reported the detailed time and memory usage for the exact-only NNF and approximate-only NNF. As aforementioned, using either exact-only or approximate-only NNF would not be satisfying, and our exact-to-approximate scheme is the key to enable synthesizing high-quality results with limited computational resources.

Figure 11. Samples generated with images collected from a real-world scenic site – Bryce canyon ©2022 Google. Notably, we only synthesize the region of interest (i.e., the odd rocks) and the background is disentangled out by modeling with an independent implicit neural network.

Table 3. Computational overhead. We report the time and memory consumption in each NNF iteration for exact-only and approximate-only NNF.

Resolution	Only Exact NNF		Only Approximate NNF	
	Time (s)	Memory (GB)	Time (s)	Memory (GB)
$16 \times 16 \times 5$	0.01	0.07	0.55	0.07
$21 \times 21 \times 7$	0.01	0.12	0.59	0.14
$28 \times 28 \times 10$	0.02	0.68	0.69	0.21
$38 \times 38 \times 14$	0.08	1.48	0.98	0.43
$51 \times 51 \times 19$	0.57	3.66	1.69	0.92
$68 \times 68 \times 26$	4.06	9.74	3.62	2.02
$91 \times 91 \times 35$	27.47	25.17	8.47	4.77
$121 \times 121 \times 47$	N/A	N/A	20.07	11.28
$162 \times 162 \times 63$	N/A	N/A	48.91	17.05
$216 \times 216 \times 84$	N/A	N/A	116.09	21.95
$288 \times 288 \times 112$	N/A	N/A	278.53	23.27
$384 \times 384 \times 150$	N/A	N/A	662.23	26.60

Failure Cases. As mentioned in the paper, our method favors scenes with complex structures and textures for matching the internal distribution, lacking sufficient diverse patch candidates will lead to mode collapse issues. Besides, with voxelized volumetric representations, our method can not perfectly synthesize scenes with tiny thin structures. Moreover, our method operates on the patch level, so we can not guarantee that highly semantic or structural features in the exemplar can be preserved intactly in the synthesized results. Figure 12 shows some failure cases when working with our method.

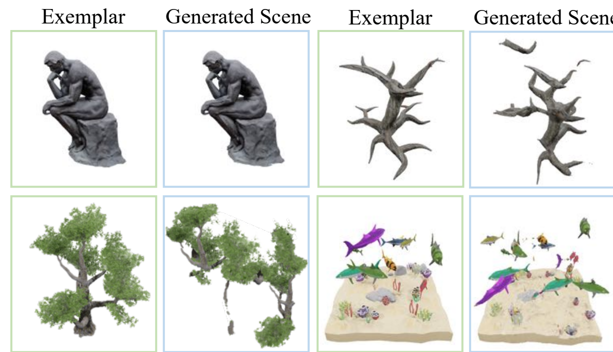

Figure 12. Failure cases. An exemplar scene, that does not have sufficient diverse patch candidates, would result in identical generation results to the input (See top left sculpture [27]). Our method failed on scenes with tiny thin structures, such as branches [16] and trees [26]. Scenes with highly semantic or structural information can not be correctly handled (See broken fishes on the bottom right [11, 28]).

F. Evaluation

F.1. Baselines

GRAF [33] We use the official implementation, and replace the camera poses with ones in our work. We follow the default setting for training, one model for each exemplar scene is trained with renderings of resolution 512^2 for $7200k$ samples, which takes about 3 days in a single V100 GPU. The final visuals are rendered at the resolution 512^2.

StyleNeRF [15] We use the official release of StyleNeRF. Same as GRAF, we replace the camera extrinsic and intrinsic parameters with the real distribution and set the background to white. All models are trained in the resolution of 512^2 following the default setting by going through $6000k$ images for about 3 days using 4 V100 GPUs.

GPNN-3D We naively extend the GPNN [14] for working on Plenoxels-based exemplar scenes. The density value and SH values are normalized to fit $[-1, 1]$, and we follow all parameters as described in the original paper. The maximum resolution reached by GPNN-3D is only 38 due to computational efficiency issues.

F.2. Camera Pose Sampling

To quantitatively evaluate the synthesized scenes from 2D projections, we uniformly sample $K = 50$ camera poses on the upper hemisphere with radius $R = 2.5$, and use elevation angles range from $0°$ to $90°$. The focal length of the camera is set to 512 times the pixel size, equivalently FoV $\approx 39.6°$, for all exemplar scenes.

F.3. Metrics

For each method, we produce 50 generated scenes on each of the evaluated exemplars, render 50 multi-view images and extract the 3D geometric surface points of the exemplar and each in the generated, and then rate the performance using a combination of several common metrics in both 2D and 3D generation:

Visual Quality measures how well the model captures the internal statistics of the input exemplar from the 2D perspective, by simply computing the averaged SIFID [34] over multiple views of a generated scene. Concretely, for each image rendered from a generated scene, we compute the single image SIFID of this image against the image rendered at the associate viewpoint in the exemplar scene. Then the SIFID-MV for a generated scene is the average over the multiple views. We finally report the mean SIFID-MV averaged over multiple generated scenes.

Visual Diversity of the set of generated scenes is measured via extending the image diversity score as in [34] to multi-view images of a scene. First, under each view, we calculated the standard deviation (std) of the intensity values of each pixel over 50 images rendered from 50 generated scenes, averaged it over all pixels, and normalized by the std of the intensity values of the image rendered from the exemplar. Then, we report the average of std values obtained at 50 views as the Visual Diversity of a set of generated scenes.

Geometry Quality of a generated scene is measured as the Minimal Matching Distance [37] (multiplied by 10^2) between the set of generated patches and exemplar patches (represented as point clouds sampled on the surface). As mentioned in the paper, Plenoxels often produce invisible noise, so we only pick point cloud patches on the surface. Specifically, for a scene represented by a discrete volume of resolution 256^3, we extract mesh using Marching Cubes [19], and evenly sample 102400 points from the mesh surface. To extract patches, we randomly pick 1000 points center, then combined them with the nearest 1024 points via k-NN search. Then the geometry quality of a generated scene is calculated as the MMD between the set of patches in a generated scene and the set of patches in the exemplar scene. We then report the averaged geometry quality score over 50 generated scenes.

Geometry Diversity of generated scenes is measured by summing up all the differences among the 50 generated scenes, i.e., Total Mutual Difference as in [37]. Specifically, we evenly sample 10240 points on the surface of each generated mesh to obtain a point cloud, forming a set of 50 scene point clouds. Empty scenes are deprecated to calculate the geometry diversity. Then the Geometry Diversity of 50 generated scenes is reported as the TMD calculated on this set of point clouds.

References

[1] Google earth studio. https://earth.google.com/studio/. 2

[2] 3ddominator. Train wagon. https://sketchfab.com/3d-models/train-wagon-42875c098c33456b84bcfcdc4c7f1c58, 2019. License: CC Attribution. 4

[3] 3dhdscan. Stone sculpture i. https://sketchfab.com/3d-models/stone-sculpture-i-fb6e253b75db40c3b38e3977ee60bc5c, 2018. License: CC Attribution. 3, 4

[4] Alec. Spinsters rock. https://sketchfab.com/3d-models/spinsters-rock-14f654eb8c4d42d991c61f907379644a, 2018. License: CC Attribution. 3, 4

[5] Animateria. Volcano island lowpoly. https://sketchfab.com/3d-models/volcano-island-lowpoly-4a6591dc9fee40d8bfda8350683af9af, 2020. License: CC Attribution. 3, 4

[6] artfletch. https://sketchfab.com/3d-models/st-alphage-church-ruin-6979d7daf8ad4bd887694ef65c1eeeae, 2018. License: CC Attribution. 3, 4

[7] artfletch. Calda house. https://sketchfab.com/3d-models/calda-house-439588b47dfb4c0984b20555163db514, 2020. License: CC Attribution. 4

[8] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 5

[9] bilgehan.korkmaz. Tiny castle. https://sketchfab.com/3d-models/tiny-castle-2836b219403d4fd487d0434f88b65363, 2018. License: CC Attribution. 4

[10] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2022. 2

[11] EdwinRC. L-p shark cage diving scene. https://sketchfab.com/3d-models/l-p-shark-cage-diving-scene-5bf4fcbbc6ea40b69c734f54180fa38c, 2018. License: CC Attribution. 6

[12] EdwinRC. Low poly desert. https://sketchfab.com/3d-models/low-poly-desert-c25524a942704f5c833ac53b645cda82, 2019. License: CC Attribution. 4

[13] evolveduk. Cactus 2 (saguaro). https://sketchfab.com/3d-models/cactus-2-saguaro-e7ba6d448e2c44ce89c716fa71cbb716, 2022. License: CC Attribution. 4

[14] Niv Granot, Ben Feinstein, Assaf Shocher, Shai Bagon, and Michal Irani. Drop the gan: In defense of patches nearest neighbors as single image generative models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4, 6

[15] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2022. 6

[16] Harry_L. Old dead tree. https://sketchfab.com/3d-models/old-dead-tree-8c680561463a4e909fd24b4cc8bc7e48, 2018. License: CC Attribution. 6

[17] Hirnlaich. Heal mountain. https://sketchfab.com/3d-models/stacked-heal-mountain-4ea3d857952347578dbb57f4e642ce9f, 2022. License: CC Attribution. 1

[18] Bilgehan Korkmaz. Green island. https://sketchfab.com/3d-models/green-island-79445f7196e24f4e833247f82aeaea0b, 2019. License: CC Attribution. 1, 4

[19] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Transactions on Graphics (SIGGRAPH)*, 21(4):163–169, 1987. 7

[20] Inc. Luma AI. Luma ai. https://lumalabs.ai/, 2022. 5

[21] marzipanne. Autumn camping. https://sketchfab.com/3d-models/autumn-camping-e41d700dcc0342cb92b8e0af59c1dabc, 2020. License: CC Attribution. 4

[22] marzipanne. Camping low poly. https://sketchfab.com/3d-models/camping-low-poly-d05f47384a6046ecaf39f857bd06c5a9, 2020. License: CC Attribution. 4

[23] marzipanne. Winter camping. https://sketchfab.com/3d-models/winter-camping-01c176fd6b204446b495f60056a7a064, 2020. License: CC Attribution. 4

[24] Megalitharchive. Callanish. https://sketchfab.com/3d-models/callanish-bf4e9b9a0e764b39b196537377c082ea, 2021. License: CC Attribution. 1, 4

[25] netgis. Cactus cereus. https://sketchfab.com/3d-models/cactus-cereus-1a51f9da8f62466797b271648e652a3d, 2021. License: CC Attribution. 3, 4

[26] Nicholas-3D. More realistic trees free! https://sketchfab.com/3d-models/more-realistic-trees-free-b5b506fc4f5d4af9b546283bdf0c6a15, 2022. License: CC Attribution. 6

[27] noe 3d.at. Der denker. https://sketchfab.com/3d-models/der-denker-434542838ef140d4a706dc42e775eb12, 2020. License: CC Attribution. 6

[28] Oneeeee. Mini low poly fish pack. https://sketchfab.com/3d-models/mini-low-poly-fish-pack-8a9b0be8fa6c4e3e9b6cd9f1f4e0967a, 2022. License: CC Attribution. 6

[29] Resistance Studios. Black mountain. https://sketchfab.com/3d-models/mountain-vr-test-aad376d6ad634d0bb5964a863f1e9a51, 2018. License: CC Attribution. 4

[30] rubenve. Stylized cactus. https://sketchfab.com/3d-models/stylized-cactus-d515ed6d19184b80830a6feba04863a8, 2019. License: CC Attribution. 4

[31] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 5

[32] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 3, 5

[33] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:20154–20166, 2020. 6

[34] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *International Conference on Computer Vision (ICCV)*, 2019. 6, 7

[35] Shahriar Shahrabi. The vast land. https://sketchfab.com/3d-models/the-vast-land-733b802f5a4743ef99ad574279d49920, 2021. License: CC Attribution. 1, 3, 4, 10

[36] Werniech. Large stone arch 3d. https://www.turbosquid.com/3d-models/large-stone-arch-3d-1908101#, 2022. License: 3D Model License: Standard. 4

[37] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *European Conference on Computer Vision (ECCV)*, pages 281–296. Springer, 2020. 7

[38] Wang Ximeng. A thousand li of rivers and mountains. https://en.wikipedia.org/wiki/Wang_Ximeng, 1113. 10

[39] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 5

[40] Šimon Ustal. Grassy mountains with lakes. https://sketchfab.com/3d-models/grassy-mountains-with-lakes-408364942d794699bbf99aaced81881d, 2021. License: CC Attribution. 4

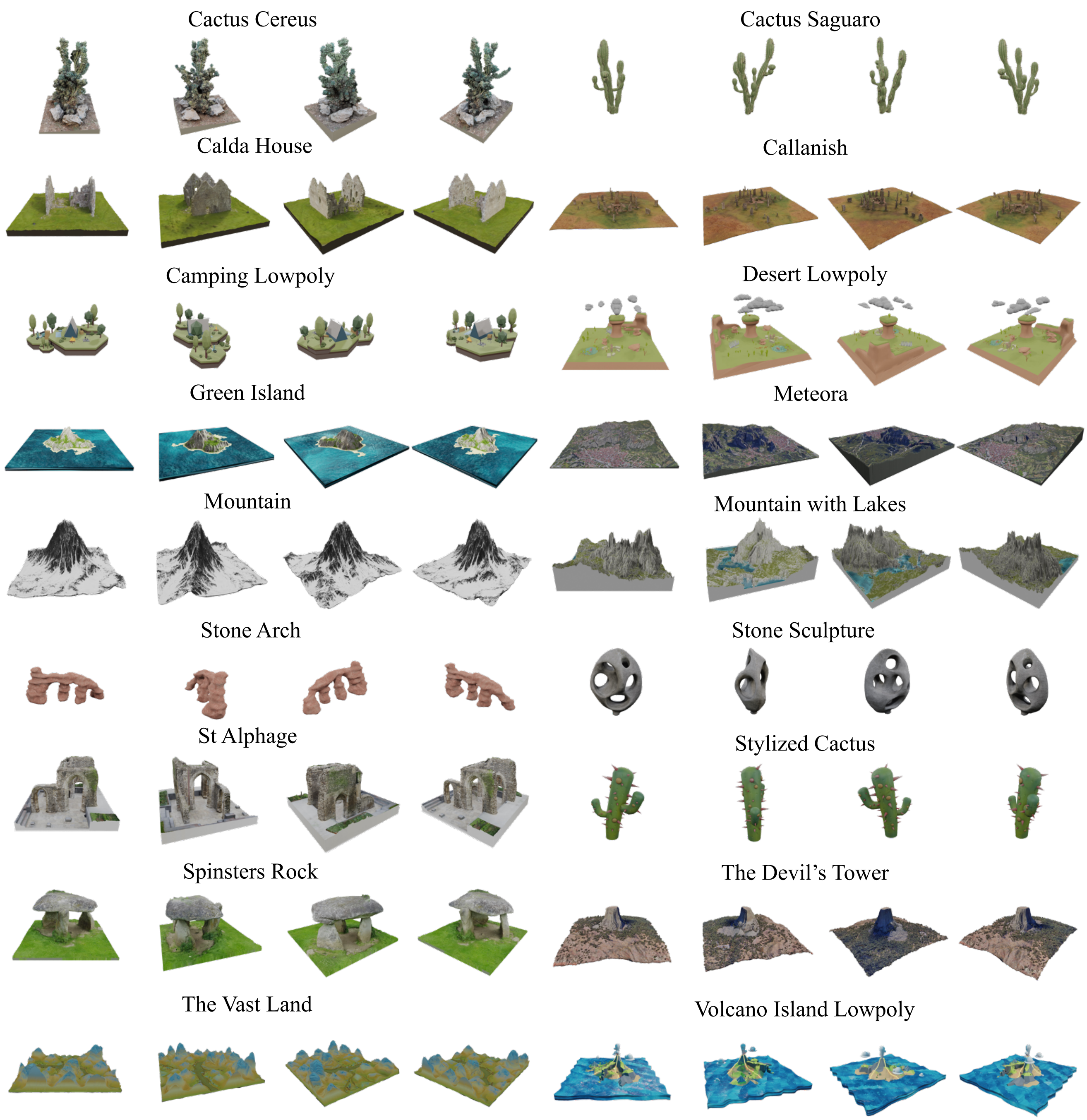

Figure 13. Visualization of the exemplars used in the **main paper**. More scenes can be found in the project page and video.

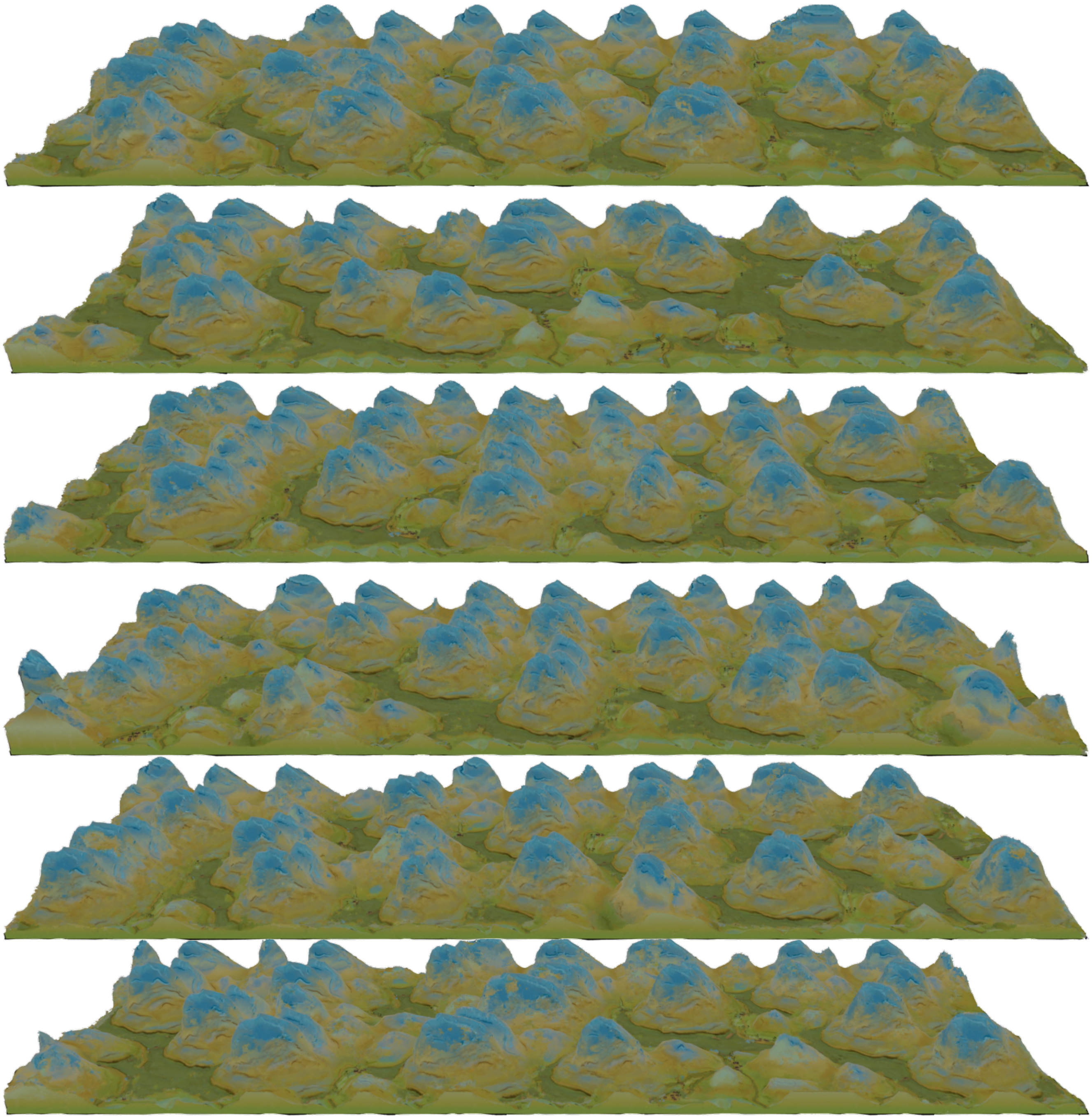

Figure 14. Diverse *"A Thousand Li of Rivers and Mountains"* [38] generated from The Vast Land [35] by our method. Specification: E_N - $288 \times 288 \times 112$, E^{high} - $512 \times 512 \times 200$, S_N - $747 \times 288 \times 112$, $E^{high}(S_N)$ - $1328 \times 512 \times 200$, final rendering resolution - 4096×1024.

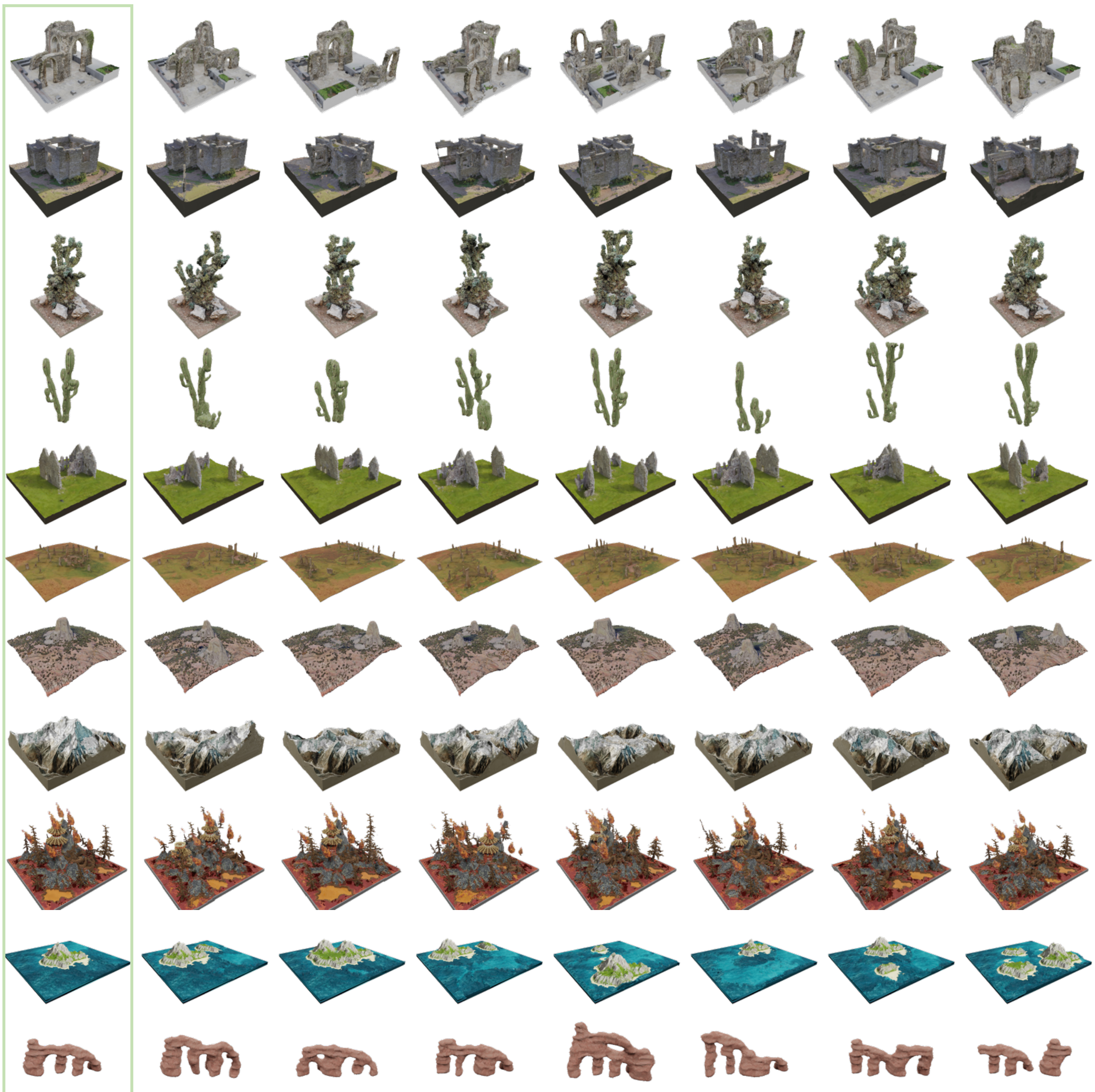

Figure 15. Diverse samples generated by our method. The input is shown in the green box on the left, followed by 7 generated novel scenes.

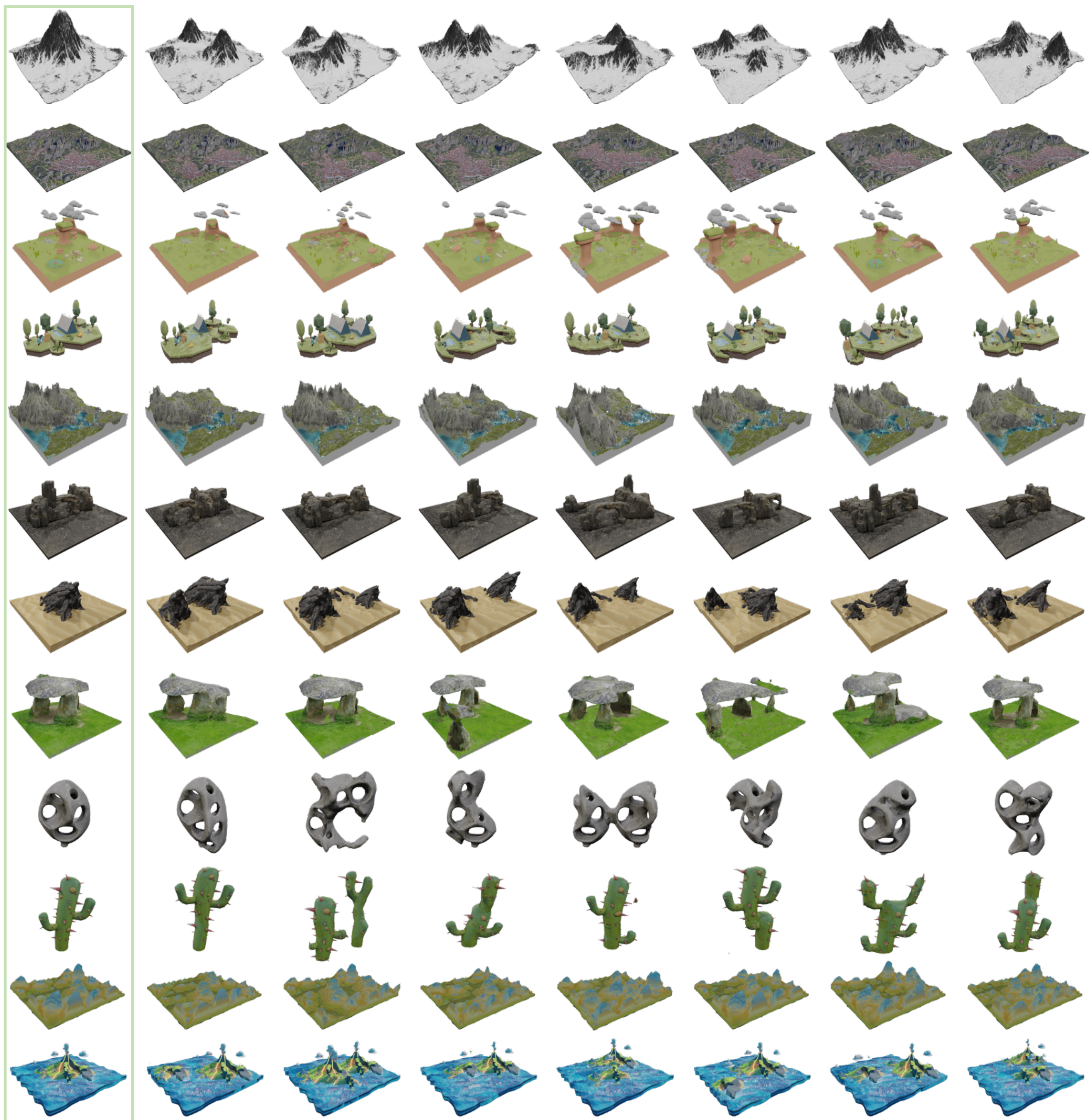

Figure 16. Diverse samples generated by our method. The input is shown in the green box on the lef, followed by 7 generated novel scenes.