MegaScenes: Scene-Level View Synthesis at Scale
Supplemental Material

Joseph Tung*[1], Gene Chou*[1], Ruojin Cai[1], Guandao Yang[2], Kai Zhang[3], Gordon Wetzstein[2], Bharath Hariharan[1], and Noah Snavely[1]

[1] Cornell University
[2] Stanford University
[3] Adobe Research

1 Visualizations of Dataset Characteristics

We provide additional figures to convey the wealth of information in the MegaScenes dataset. In Fig. 1, we highlight the diversity of images registered to a reconstruction, Wikidata class information, and image subcategories for an outdoor scene. In Fig. 2, we show examples of image pairs with calculated two-view geometries for an indoor scene.

2 Details for Dataset Curation

2.1 Processing Wikidata Entries for Scene Identification

In the first dataset curation step, "Identifying Scenes," we collect raw Wikidata entries from broad categories that link to Wikimedia Commons categories. Each Wikidata entry points to some Wikimedia Commons category; we take the set of these Wikimedia Commons categories to use as scenes. Before we download images from these categories, we do additional cleaning steps to determine which Wikimedia Commons categories to use.

Filtering Wikidata Entries based on Cyclic Links. Some collected Wikidata entries point to broad Wikimedia Commons categories, like Fountains or Cultural heritage monuments in Toropetsky District, unsuitable to use as single scenes. We note that a Wikidata entry points to some Wikimedia Commons category, and a Wikimedia Commons category points towards some, but not necessarily the same, Wikidata entry. To clean the set of Wikimedia Commons categories as described above, we ensure that there is a cyclic link between the Wikimedia Commons Category and its corresponding Wikidata entry. A Wikimedia Commons category like Fountains will point to a Wikidata entry about fountains, and not the original Wikidata entry that pointed to the Fountains Wikimedia Commons category.

Filtering Wikidata Entries based on GLAM Instances. We find that some categories related to galleries, libraries, archives, and museums (GLAM) contain many images that are unhelpful in 3D reconstructions, such as 2D scans

* Equal contribution.

of paintings or text. To minimize the number of such 2D scans, we ignore all Wikidata entries that are *exclusively* GLAM instances. We keep the Wikidata entries that are also instances of at least one other unrelated class, as these are more likely to have images that are not exclusively of 2D scans.

2.2 Subcategory Recursion when Downloading Images

In the second dataset curation step, "Downloading Images from Scenes," we download images from every Wikimedia Commons category deemed a scene. From the original Commons category, we recurse a maximum depth of four subcategories and download all associated images.

However, some subcategories are unrelated to the original scene, and we want to avoid downloading images unhelpful in 3D reconstruction. For instance, if a subcategory begins with "People associated with...", then the subcategory will link to Wikimedia Commons pages that contain images of individuals, rather than the original scene.

To fix the above issue, we define two conditions that must be met in order to recurse into a related subcategory. First, we create a list of excluded keywords that the subcategory must not contain. We curate this list by experimentally finding common diverging subcategories, which includes keywords like "People associated with...". Second, the subcategory must contain a substring that includes one of the following names:

- The name of the original Wikimedia Commons category
- The name of the Wikidata entity associated with the original Wikimedia Commons category
- Any alias recorded on the Wikidata entity associated with the original Wikimedia Commons category

2.3 Details on Reconstruction and Cleaning

We elaborate on the third curation step, "Reconstructing Scenes with SfM and Cleaning Reconstructions."

Reconstruction Orientation Alignment. The reconstructions created by COLMAP [8] are not necessarily aligned to the real world. For instance, the gravity axis as seen in input images may not align with the down-axis of the reconstruction's coordinate system. Thus, we orient all reconstructions using COLMAP's implementation of Manhattan world alignment. This process aligns the sparse reconstruction using the Manhattan World assunption [3], which assumes that most surfaces are aligned along the three major axes.

Cleaning Watermarked Reconstructions. Some scenes have many images with watermarks. COLMAP finds spurious matches between the watermarks of two images, which result in incorrect sparse reconstructions. We assume that most images uploaded to Wikimedia Commons have watermarks that are non-destructive and are near the borders of the image. To fix this issue, we mask

all keypoints within a certain distance near the image border before we rerun COLMAP's feature matching and reconstruction phases. Specifically, we target all scenes where at least 10 percent of the inlier pairs are "watermark pairs" as labeled in COLMAP's output database. We find that a border defined by 5 percent of the image diagonal is able to mask watermarks in most images.

Using Doppelgangers to Clean Reconstructions. As discussed in the main paper, we use the Doppelgangers [2] pipeline to fix incorrect SfM reconstructions caused by visual ambiguities. Incorrect reconstructions arise from false correspondences between image pairs, such as in photos that depict different surfaces that are similar in appearance. Doppelgangers uses a binary classifier that predicts the likelihood of whether an input image pair should be matched; the input pairs to SfM are filtered by passing them through the Doppelgangers classifier. After we filter the image pairs, we rerun COLMAP's reconstruction phase. We find that the default threshold of keeping pairs with a confidence score of ≥ 0.8 is able to correctly disambiguate most scenes. If thresholding at 0.8 is unsuccessful, we try increasing thresholds until the reconstruction is correct.

3 Additional Details on Dataset Statistics

3.1 Wikidata Classes to Identify Scenes

We show the Wikidata classes we use to identify scenes in our dataset curation process in Table 1. Refer to Sec. 2.1 for how the 660K Wikidata entries are filtered into 430K scenes.

3.2 Scene Overlap with Google Landmarks Dataset V2

While MegaScenes and Google Landmarks Dataset V2 (GLDv2) [11] both source images from Wikimedia Commons, we find that neither dataset has a majority overlap with the other in terms of categories used as scenes. MegaScenes contains 430K categories as scenes, while GLDv2 contains 213K categories as scenes; there are 74K scenes that are found in both datasets. This means that MegaScenes has 356K scenes not found in GLDv2, and GLDv2 has 139K scenes not found in MegaScenes. We attribute this to differing data curation methods for both datasets: GLDv2 queries the Google Knowledge Graph, while MegaScenes utilizes Wikidata.

4 Details for Novel View Synthesis Experiments

4.1 Data Setup for Evaluation

DTU. We use the test split of 15 scenes from previous work [7] on the DTU dataset [10]. Each scene contains 49 images with the same exact array of camera positions, and we pick two reference locations that are across from each other.

WikiData Class	Entry Count
religious building	233,253
monument	75,196
tourist attraction	55,051
museum	40,653
landmark	34,950
bridge	33,207
chapel	29,866
commercial building	24,859
public building	24,227
shrine	22,055
tower	17,915
square	13,817
statue	10,872
palace	10,237
Catholic church building	8,792
fountain	5,496
high-rise building	4,083
Eastern Orthodox church building	3,782
cathedral	3,326
mosque	3,093
house of prayer	3,092
library building	742
arch	349
gurdwara	57
Total	**659,024**

Table 1: WikiData classes that have been selected to identify a set of scenes. Some WikiData entries may be present in multiple classes. Multiple WikiData entries may link to the same Wikimedia Commons category.

See the supplement for more details. For each reference image, we exhaustively form pairs with all other images in the scene. This results in 95×2 pairs (we count (a, b) and (b, a) as separate pairs) per scene, for a total of 2,850 pairs.

Mip-NeRF 360. We use all 9 scenes from the original Mip-NeRF 360 dataset [1]. We leverage how the images in these scenes form a 360 degree orbit about a central location to identify reference images. We align the provided COLMAP sparse point cloud using the Manhattan world assumption [3], then sort the images by increasing viewing direction angle on the XZ plane. We sample ten evenly distributed images from this sorted list. For each reference image, we pair it with all images that share at least 50 3D points in the sparse point cloud. This results in a total of 15,862 pairs across all scenes.

RealEstate10K (Re10K). We adopt the pair sampling strategy from Synsin [12] for Re10K [13], which selects a reference and target video frame no more than 30 frames apart. To identify more challenging frames, the authors choose pairs with an angular change greater than 5° and a positional change greater than 0.15, whenever possible. We take the intersection of these pairs and ZeroNVS' [7] held out set (since ZeroNVS was trained on RE10K). In total, we obtain 644 pairs across 163 clips.

4.2 Finetuning and Inference

For all finetuning, we use 6 NVIDIA A6000 GPUs with a total batch size of 1656 until the metrics of the validation set stop improving. We finetune ZeroNVS [7] for 30,000 iterations which takes 1-2 days. We finetune Zero-1-to-3 [6] for 75,000 iterations. During inference, we use 50 DDIM [9] steps for all qualitative and quantitative results. We use classifier-free guidance scales [4] of 3.0 for all finetuned models and *SD-inpainting*. We observed less realistic generations with little to no improvements in metrics when setting a higher scale. For *Zero-1-to-3 (released)* and *ZeroNVS (released)*, we use a scale of 7.5 following the default in ZeroNVS, as the models match the target poses better at a higher scale.

Our method (denoted *Ours* in the main paper) is finetuned from ZeroNVS and combines the warped images and extrinsic matrices as conditions. Similar to Zero-1-to-3 and ZeroNVS, we pass the target and reference images through a pretrained VAE to obtain their latents, each with shape (4, 32, 32). We downsample the warped image from (3, 256, 256) to (3, 32, 32) and concatenate it with the target and reference latents, so that our input to the first layer of the diffusion model has shape (11, 32, 32). The conditioning of the extrinsic matrices is exactly follows ZeroNVS. We scale the translation vector of the extrinsic matrix from COLMAP by the 20th quantile of the aligned depth (Fig 4 of the main paper).

For *SD-inpainting*, we use the checkpoint *sd-v1-5-inpaint.ckpt* (`https://huggingface.co/runwayml/stable-diffusion-inpainting`) from Runway ML. Since the model is trained on 512x512 images, we use 512x512 images and 64x64 latents to match. Then, we downsample the outputs to 256x256 for qualitative results and calculating metrics.

5 Additional Qualitative Results and Comparisons

We show additional qualitative results uniformly sampled from our test set in the PDF file named *qualitative results.pdf*. We will release data, seeds, and models for reproduction. In virtually all cases, finetuned models *(MS)* outperform base models in terms of pose consistency and realism. *Ours* generally follows the target pose more closely than *ZeroNVS (MS)* and *Zero-1-to-3 (MS)*.

We also finetune ZeroNVS on MegaDepth [5], denoted *ZeroNVS (MD)*. MegaDepth is the most similar dataset to MegaScenes, consisting of diverse internet photos with COLMAP reconstructions. However, MegaDepth is of a significantly smaller scale. For the task of novel view synthesis, we process MegaDepth through the same process as described in Section 4.1 of the main paper, and obtain a total of 368,028 training pairs and 181 scenes, roughly 6 times fewer pairs, and 180 times fewer scenes. We find that *ZeroNVS (MS)* generally outperforms *ZeroNVS (MD)* in terms of both following the target pose and realism, especially of less common scenes.

6 Videos

We show videos on our project page. Videos contain sequential frames and make it easy to visualize the pose consistencies from frame to frame. We also include an example of autoregressive generation, where we take the last view of a generated sequence as the first frame of a new sequence. However, since each image is sampled independently, long sequences eventually drift. In the future, adding temporal constraints or directly generating multiple frames could solve this issue.

7 Broader Impact

This work primarily focuses on the task of novel view synthesis, and similar to other generative models, present risks such as the potential for generating misleading or harmful content. It is essential to develop robust frameworks for ethical use.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022) 5
2. Cai, R., Tung, J., Wang, Q., Averbuch-Elor, H., Hariharan, B., Snavely, N.: Doppelgangers: Learning to disambiguate images of similar structures. In: ICCV (2023) 3
3. Coughlan, J.M., Yuille, A.L.: Manhattan world: Compass direction from a single image by bayesian inference. In: ICCV (1999) 2, 5
4. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022) 5

5. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Computer Vision and Pattern Recognition (CVPR) (2018) 6
6. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023) 5
7. Sargent, K., Li, Z., Shah, T., Herrmann, C., Yu, H.X., Zhang, Y., Chan, E.R., Lagun, D., Fei-Fei, L., Sun, D., et al.: Zeronvs: Zero-shot 360-degree view synthesis from a single real image. arXiv preprint arXiv:2310.17994 (2023) 3, 5
8. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 2, 9
9. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 5
10. Sølund, T., Buch, A.G., Krüger, N., Aanæs, H.: A large scale 3d object recognition dataset. In: 3DV (2016) 3
11. Weyand, T., Araujo, A., Cao, B., Sim, J.: Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In: Proc. CVPR (2020) 3
12. Wiles, O., Gkioxari, G., Szeliski, R., Johnson, J.: Synsin: End-to-end view synthesis from a single image. In: CVPR. pp. 7467–7477 (2020) 5
13. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. ACM Trans. Graph. (Proc. SIGGRAPH) **37** (2018), https://arxiv.org/abs/1805.09817 5

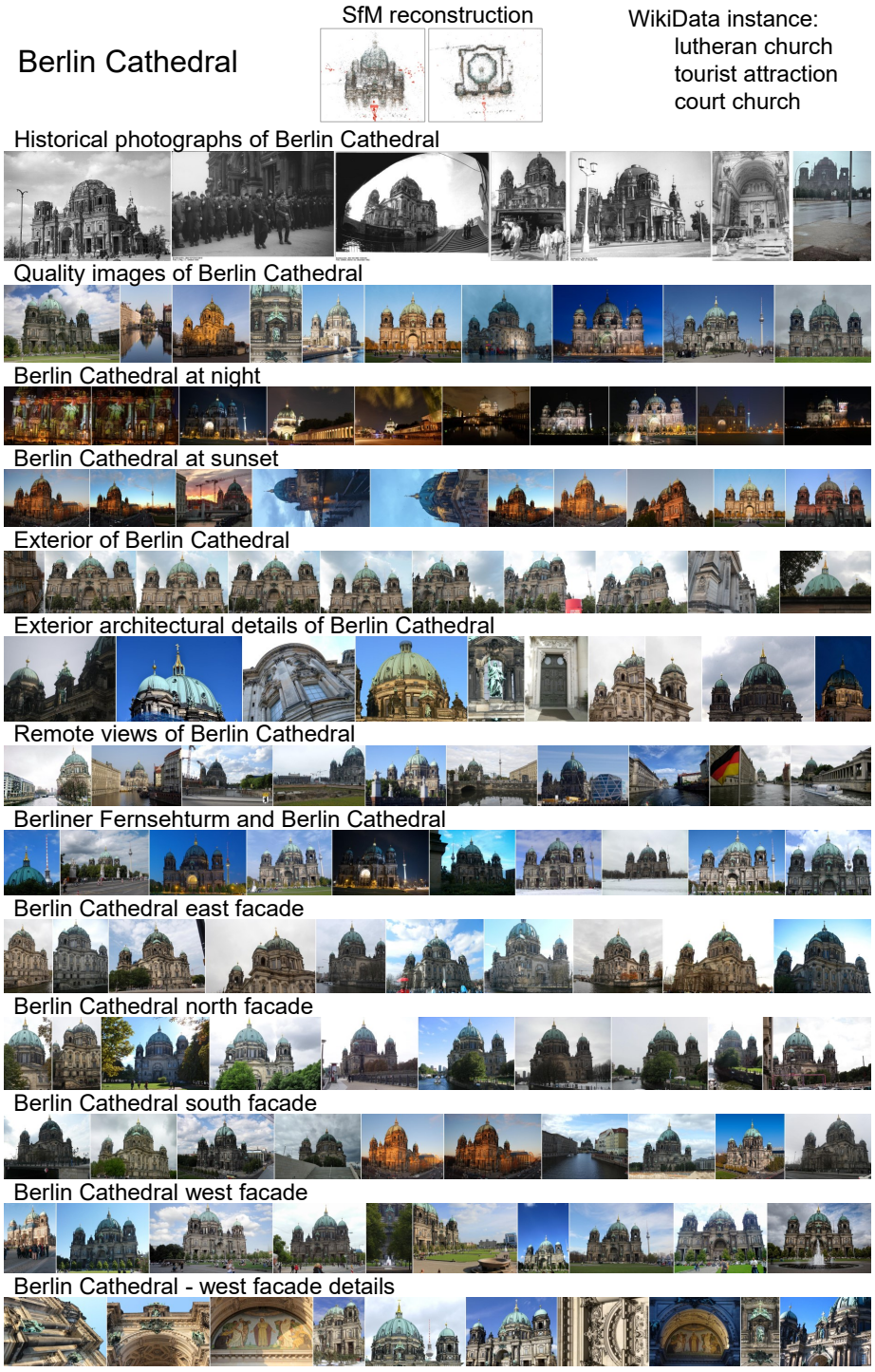

Fig. 1: Registered images of Berlin Cathedral, organized by Wikimedia Commons subcategories. Each text label corresponds to a subcategory (possibly nested) of the main category.

Fig. 2: Feature matching visualization of the Natural History Museum's interior in London, with the SfM reconstruction shown in the middle of the top row. On the left, selected image pairs are shown, and on the right, the extracted keypoints (in blue) and their matches (in color-coded) from the two-view geometry table in COLMAP [8] database are displayed.