Cross-View Splatter: Feed-Forward View Synthesis with Georeferenced Images

Supplementary Material

In this supplementary material, we provide additional implementation details omitted from the main paper. We describe our custom benchmark datasets, outline the evaluation protocol, present further experimental analysis of our method, and illustrate more qualitative results of our model's performance alongside comparisons to baseline methods. We further discuss the limitations of the proposed Cross-View Splatter approach.

A. More Implementation Details

Training. We train our method with an initial learning rate of 1×10^{-4} using the AdamW [4] optimizer (weight decay 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.95$), and apply a cosine annealing schedule for $70K$ iterations. For training, we initialize with [3] weights and freeze ground-level patch embedding, $\text{Attn}_{\text{frame}}$ and $\text{Attn}_{\text{global}}$ layers and fine-tune all output heads which include the camera, depth, and Gaussian prediction heads as well as our $\text{Attn}_{\text{meta}}$ layers. Our total supervision loss is given by:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \lambda_{\text{cam}}\mathcal{L}_{\text{cam}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{const}}\mathcal{L}_{\text{const}} \\ + \lambda_{\text{height}}\mathcal{L}_{\text{height}} \\ + \lambda_{\text{ground}}\mathcal{L}^{\text{ground}}_{\text{RGB}} + \lambda_{\text{combined}}\mathcal{L}^{\text{combined}}_{\text{RGB}} + \lambda_{\text{sat}}\mathcal{L}^{\text{sat}}_{\text{RGB}} \\ + \lambda_{\text{sky}}\mathcal{L}_{\text{sky}} + \lambda_{\text{bev}}\mathcal{L}_{\text{BEV}},\end{aligned} \tag{1}$$

where $\mathcal{L}_{\text{sky}} = \mathcal{L}_{\text{sky\_depth}} + \mathcal{L}_{\text{sky\_alpha}}$. We set $\lambda_{\text{cam}} = 1.0$, $\lambda_{\text{depth}} = 1.0$, $\lambda_{\text{const}} = 1.0$, $\lambda_{\text{height}} = 1.0$, $\lambda_{\text{ground}} = 1.0$, $\lambda_{\text{combined}} = 1.0$, $\lambda_{\text{sat}} = 1.0$, $\lambda_{\text{sky}} = 0.1$, and $\lambda_{\text{BEV}} = 0.5$. Note, we utilize ground truth terrain heights *only* during training for our height regression loss $\mathcal{L}_{\text{height}}$. At inference time, we utilize satellite RGB images.

Gaussian rendering. For Cross-View Splatter, we are able to render images from ground Gaussians $\mathcal{G}^{\text{ground}}$, satellite Gaussians $\mathcal{G}^{\text{sat}}$, and combined Gaussians $\mathcal{G}^{\text{combined}}$. In the worst case, to apply our ground level RGB losses $\mathcal{L}^{\text{ground}}_{\text{RGB}}$, $\mathcal{L}^{\text{sat}}_{\text{RGB}}$, and $\mathcal{L}^{\text{combined}}_{\text{RGB}}$ we require three forward calls to the `gsplat` [17] rasterizer. Instead, we do two forward calls, one for $\mathcal{G}^{\text{ground}}$ and $\mathcal{G}^{\text{sat}}$ and alpha-blend to obtain:

$$C^{\text{combined}}_{\text{3DGS}} \approx C^{\text{ground}}_{\text{3DGS}} + (1-\alpha_{\text{ground}})C^{\text{sat}}_{\text{3DGS}}, \tag{2}$$

where $\alpha_{\text{ground}} = \sum_{i=1}^{M} \alpha_{\text{ground},i} \prod_{j=1}^{i-1}(1-\alpha_{\text{ground},j})$ is the accumulated transparency for ground Gaussians. This formulation is not strictly equivalent to rendering the unified set $\mathcal{G}^{\text{combined}} = \mathcal{G}^{\text{ground}} \cup \mathcal{G}^{\text{sat}}$, because satellite Gaussians that would occlude ground Gaussians along the camera ray are implicitly omitted in Eq. (2).

Scene normalization. The choice of scene normalization during training is a critical design choice. Approaches vary, including scaling by ground truth depth maps [12, 13, 15] or camera baseline distances [16], or adopting a fully metric coordinate system [1, 14]. As mentioned in Sec. 3.4, we adopt the per-batch ℓ_2-norm scaling derived from back-projected depths [12, 15] to normalize the depth and pose of the ground-level imagery. A crucial difference is that we also regress height maps $\mathbf{h}^{\text{sat}}$ relative to I_0^{ground} from orthoimages with a known spatial resolution r^{sat} (expressed in pixels per meter). The spatial resolution r^{sat} is used to map between satellite pixel space and world coordinates. Although we regress a single per-pixel scalar value for height maps, its spatial consistency with the ground level depth and camera poses is paramount. Therefore, we integrate $\mathbf{h}^{\text{sat}}$ and r^{sat} into the same normalization scheme. Specifically, we compute a scalar value s with:

$$s = \frac{1}{M}\sum_{j=1}^{M} \|\boldsymbol{\mu}_j\|_2, \quad \text{where } \boldsymbol{\mu}_j = \text{backproject}(d_j, \boldsymbol{K}_j, \boldsymbol{T}_j). \tag{3}$$

s is then used to normalize all metric quantities during training: camera poses, depth maps, height maps, and the satellite spatial resolution factor: $\hat{\boldsymbol{T}} = \frac{\boldsymbol{T}}{s}$, $\hat{d} = \frac{d}{s}$, $\hat{\boldsymbol{h}}^{\text{sat}} = \frac{\boldsymbol{h}^{\text{sat}}}{s}$, $\hat{r}^{\text{sat}} = s \cdot r^{\text{sat}}$. We then train our network to regress values in this normalized space.

Height ambiguity. We have one unknown degree of freedom in our training data, mainly the height of the ground level camera with respect to the BEV height maps. We follow prior works [9, 10] and set the ground level height to 2 meters off the ground for all datasets. Due to this ambiguity, it is possible that the satellite-to-ground level renders are not perfectly aligned with ground perspectives.

B. Geoaligned Benchmark Dataset

Benchmark alignment. We introduce a new task, *novel-view synthesis with geolocalized images*, and construct an evaluation dataset that remains in-domain with prior work to ensure fair comparison. Neither the Tanks and Temples (Table 2) nor the DL3DV-Benchmark (Table 3) datasets provide GPS metadata for geolocating ground-level images. We thus perform manual alignment to localize scenes. We first identify cues in the input images, such as street signs, building names, or distinctive landmarks (e.g., statues or monuments) and use these to obtain approximate GPS coordinates via Google Maps.

After establishing a coarse estimate, we refine the localization by projecting sparse COLMAP reconstruction points to satellite imagery. We used COLMAP reconstructions provided by DL3DV. For Tanks and Temples, we used

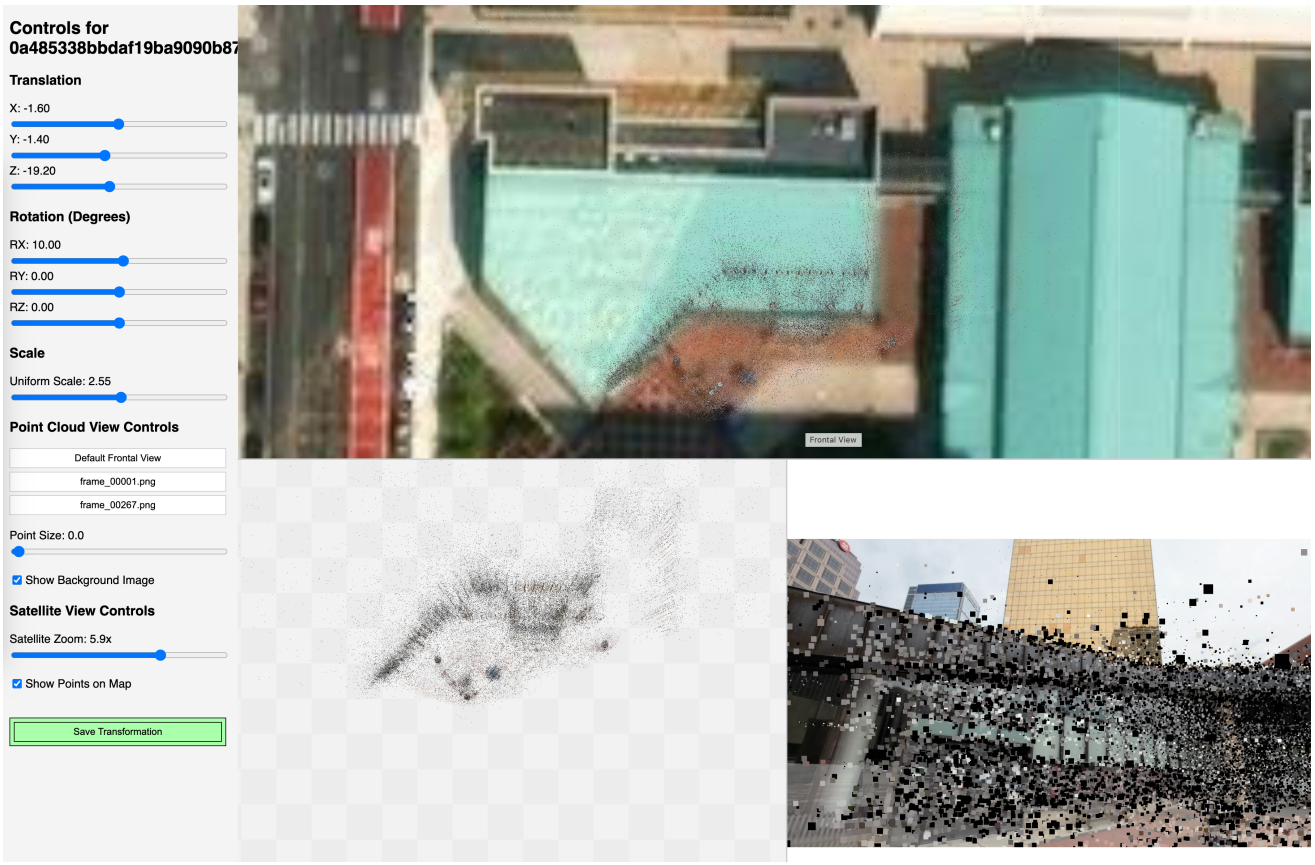

Figure 1. **Benchmark geoalignment tool.** We manually align COLMAP reconstructions to satellite imagery for 10 scenes from Tanks and Temples and 40 scenes from DL3DV-Benchmark datasets. Top: satellite image. Bottom-left: aligned COLMAP pointcloud. Bottom-right: visualization of points projected to a scene image.

camera intrinsics and camera poses provided by the dataset and ran 'colmap point_triangulator' command to generate the sparse reconstruction.

We then manually find translation, rotation, and scaling factor that aligns the point cloud to satellite imagery with known spatial resolution, transforming scenes to metric space. In Fig. 1 we visualization the alignment process with a scene from the DL3DV-Benchmark dataset. Since the COLMAP reconstructions for these scenes are rather dense, we hypothesize that this manual alignment is accurate within a few meters, but not pixel-perfect. We will release the aligned COLMAP poses and location information for our benchmark scenes to facilitate further research in this area. We visualize satellite imagery and ground images for Tanks and Temples samples in Fig. 2 and DL3DV-Benchmark in Fig. 3.

Test split creation. We construct our context and target view splits such that we have a range of increasingly challenging and representative scenarios where overlap is reduced.

To compute frame overlap for a frame pair, we compute the IOU of visible COLMAP tracked points. We count the number of COLMAP tracked points visible in both frames and divide by the union of points across frames.

For all splits, we pick the first image in the sequence as a context view. For the 2 and 3 context-view splits, we greedily select context frames that most closely satisfy a target IOU overlap to the first context image (0.15 for DL3DV and 0.25 for Tanks and Temples). We then select four target frames that each satisfy an average IOU to selected context frames. Those targets are 0.02, 0.05, 0.07, and 0.1 for DL3DV and 0.03, 0.07, 0.1, and 0.15 for Tanks and Temples.

We've found that the same targets IOUs don't yield the desired behavior across datasets given differences in how COLMAP reconstructions were created and the image resolution affecting the number and coverage of tracked points. Therefore, we select a different set of target IOU values.

Evaluating baselines. We evaluate all baselines using their publicly available code and pretrained weights, strictly following each method's recommended evaluation protocol. MVSplat and DepthSplat require ground-truth camera

Figure 2. **Our Tanks and Temples benchmark.** Visualization of satellite and ground level images.

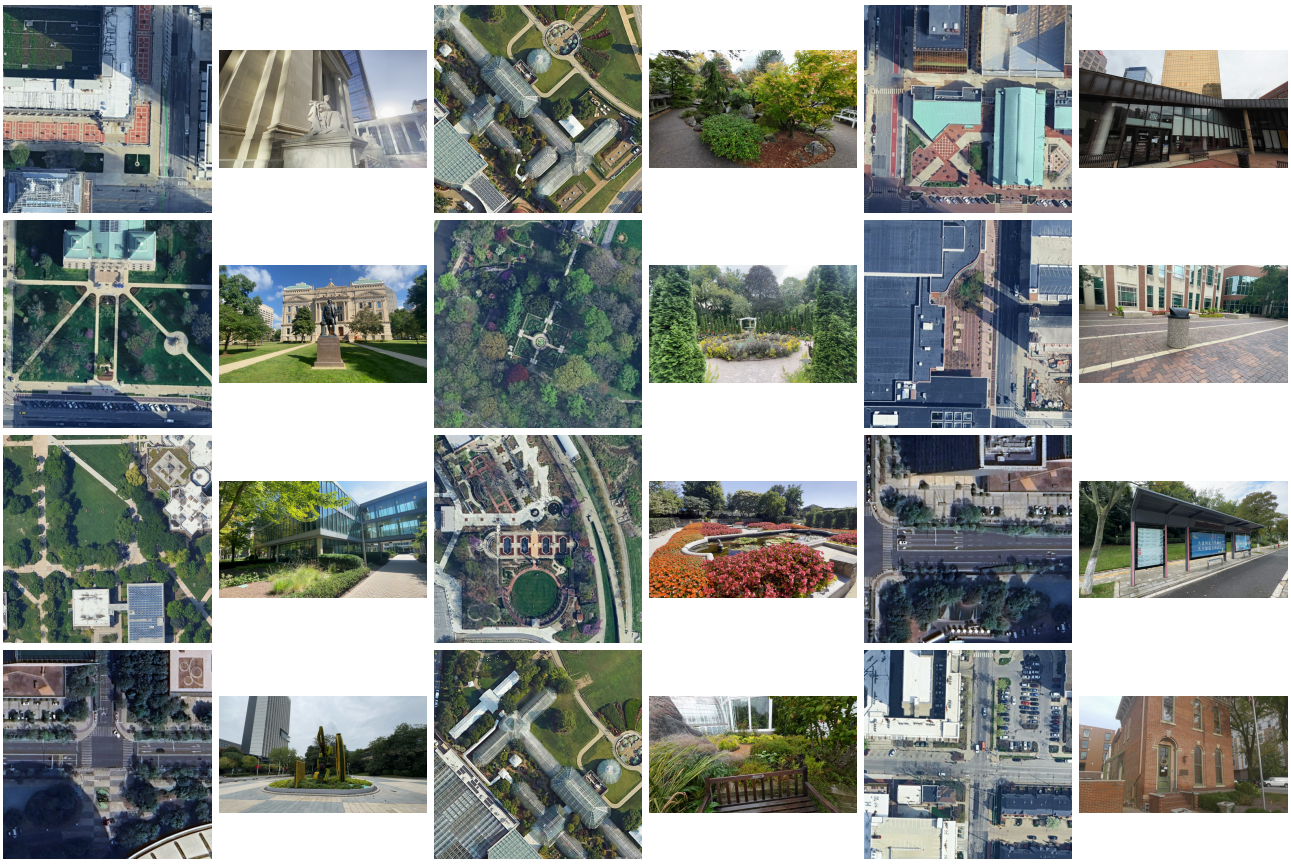

Figure 3. **Our DL3DV benchmark.** Visualization of satellite and ground level images.

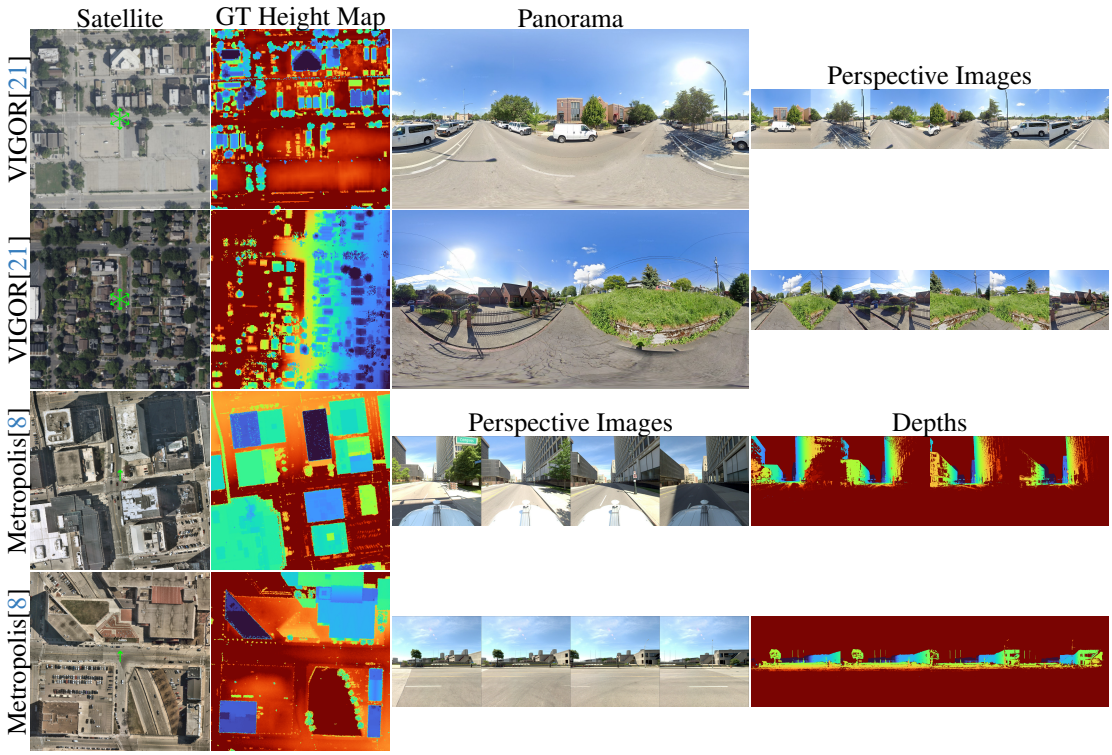

Figure 4. **Training data visualization**. We showcase our training data that consists of satellite images and terrain height maps aligned with ground level images.

poses for novel-view synthesis. NoPoSplat first reconstructs a splat and then refines each novel camera pose for 200 iterations to align it with the reconstruction. Long-LRM uses Plucker rays for the target-views. AnySplat performs two forward passes: one using only the context views, and another using both context and target views; the latter provides estimated target poses that are then used to evaluate the context-only model. We adopt the same evaluation settings as AnySplat for our Cross-View Splatter evaluation protocol.

C. Training Data

Dataset reproducibility. Our satellite API sources (Google, Azure, Esri) have licensing constraints on sharing. Therefore, we are unable to directly share raw satellite images; however, code to query data for georeferenced locations can be provided; although, exact replication of training data is uncertain due to the black-box nature of the APIs that can change with time. For our terrain height data, we are able to release the full raw training data.

VIGOR[21]. The original VIGOR dataset contains panorama images with non-centered satellite images with large zoom levels. We regenerate the dataset satellite images such that they are centered at the panorama latitude, longitude location. We also generate height maps for these locations. We create perspective images with 90° FOV from the panoramas and sample these as our context and target images during training.

Metropolis [8]. The Mapillary Metropolis dataset provides high-resolution satellite imagery, perspective driving images (captured from forward, backward, left, and right cameras), panoramas, and MVS depth maps. We use the original satellite images and extract centered crops at ground-level positions. We also project Lidar depths to satellite images using the GDAL [2] library and these serve as our height maps. The forward and backward perspective images suffer from severe occlusions caused by the vehicle itself, and we observed that training directly on this data leads to degraded reconstruction quality due to multi-view inconsistencies. To mitigate this, we mask out thge vehicle using binary masks generated by SAM2.

We visualize various training data samples in Fig. 4.

D. Discussion of baselines

AnySplat [3]. We use AnySplat model at `https://github.com/InternRobotics/AnySplat` for our model initialisation and comparisons. The model was trained on DL3DV-10K [7] dataset, however the 140

scenes used in DL3DV-Benchmark split were removed from the training split, see https://github.com/InternRobotics/AnySplat/issues/9.

Long-LRM [22]. Long-LRM also removes the 140 scenes used in DL3DV-Benchmark split from the training set, see https://github.com/arthurhero/Long-LRM?tab=readme-ov-file#long-lrm-evaluation-results.

FLARE [19]. Unfortunately, FLARE uses a custom train/test split of original DL3DV-10K, see https://github.com/ant-research/FLARE/blob/main/assets/DL3DV.json. Thus, their training set includes scenes from DL3DV-Benchmark making comparison unfair.

Furthermore, FLARE trains on Megadepth dataset [6], which contains scenes of Tanks and Temples dataset [5], *e.g.* scenes in folders 5000, 5001, 5002, 5003, 5004, 5005, 5006, 5007, 5008, 5009, 5010, 5011, 5012, 5013.

As a result, we omit direct evaluation of FLARE on our benchmarks.

Non-public baselines. The paper does not show comparisons to some closely related methods. This is due to the fact that our evaluation method requires GPS locations for input images, so we cannot compare scores on existing benchmarks. We thus re-run baselines on our geolocalized evaluation scenes. For some methods, there was no code available to run at the time of the submission and re-implementation is non-trivial, see answer to "Can reviews request comparison to closed source?" on CVPR ReviewerGuidelines page[1].

Below we list some of the competing methods that we aim to add to the evaluation table when official implementations are available.

GS-LRM [18]. At the time of submission no code or model is available on the project web page https://sai-bi.github.io/project/gs-lrm/.

Bolt-3D [11]. At the time of submission no code or model is available on the project web page https://szymanowiczs.github.io/bolt3d.

Sat2Density++ [9]. At the time of submission a pretrained model is not available on the project web page https://qianmingduowan.github.io/sat2density-pp/.

E. More Experimental Analysis

Here we provide more experimental analysis of the design choices of our proposed Cross-View Splatter.

[1] https://cvpr.thecvf.com/Conferences/2025/ReviewerGuidelines

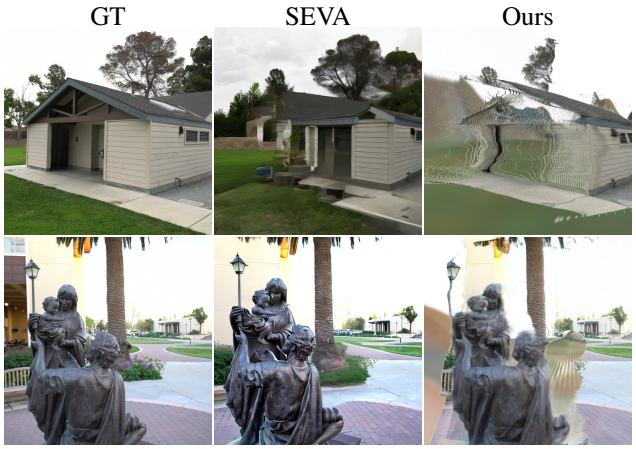

Figure 5. **Qualitative comparison to SEVA.** SEVA is a generative based model capable of hallucinating unseen areas whereas our Cross-View Splatter is a feed-forward approach that predicts geometry *only for visible regions* in ground images and satellite image.

E.1. Comparison to Diffusion Based Method

We compare our method to Stable Virtual Camera (SEVA) [20], which is a state-of-the-art diffusion model for view-synthesis that takes in ground-level imagery and target render poses. We show qualitative renders in Fig. 5 and quantitative results in Tab. 1.

E.2. GPS Sensitivity Analysis

We conduct satellite alignment sensitivity analysis in Tab. 2, simulating GPS noise and showing that *Ours* is robust to noise. Specifically, we add Gaussian noise to the 3DoF translation and rotation at increasing intervals (σ) and report mean and variance after five runs (random seeds). As satellite images have a lower sampling density compared to ground views, minor coordinate shifts do not significantly impact ground NVS renders.

F. More Qualitative Analysis

We show more qualitative renders from our model. In Fig. 7 we show more comparisons to baseline methods. In Fig. 8 we visualize side-by-side comparisons of our ground only model (referred to as Ground in the main paper) and our satellite enabled full model (referred to as Combined in the main paper) to visually demonstrate the improved coverage and completeness obtained from our full model. In Fig. 9 we visualize the outputs from the satellite branch independently. Finally, in Fig. 10 we compare our satellite branch predictions to those obtained from Sat2Density [10]. We observe that we obtain more detailed BEV height map estimates as well as more realistic geometry when we render ground views from the height estimates.

Table 1. **Comparison to diffusion based SEVA model** on our geoaligned Tanks and Temples benchmark.

Method		GT Pose?	1 context view			2 context views			3 context views		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
SEVA		✓	10.39	0.3024	**0.6066**	**12.09**	0.3614	**0.5284**	**12.65**	0.3723	**0.5034**
Ours	*Combined*	-	**11.13**	**0.3764**	0.6286	11.67	**0.3725**	0.5984	12.00	**0.3855**	0.5699

Table 2. **GPS sensitivity analysis** results for the 1-context view setting for *Combined* (Cross-View Splatter) method on Tanks & Temples.

Trans. Noise (σ)	***Combined* PSNR↑**	**Rot. Noise (σ)**	***Combined* PSNR↑**
0m (Manual aligned)	11.13	0° (Manual aligned)	11.13
1m	11.09 ± 0.04	5°	11.14 ± 0.03
3m	11.13 ± 0.06	10°	11.16 ± 0.18
5m	11.12 ± 0.04	15°	11.11 ± 0.12

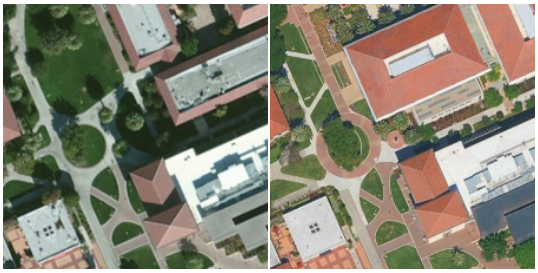

Figure 6. **Limitations of satellite imagery**. Notice how a building has been rebuilt and expanded in the right frame compared to the left taken a few years ago. This is Family scene in Tanks and Temples.

G. Limitations

Our method struggles in scenarios where the ground-level camera observes directions that fall outside the satellite orthographic view. For example, when looking upward toward the sky or downward at the ground. Because these look-at directions are not seen in the BEV views, their geometry cannot be reliably inferred. The approach is also unsuitable for environments that are not visible from above, such as indoor scenes, tunnels, underpasses, or structures with significant overhangs. Additionally, in small-baseline, high-overlap input views, the advantages of incorporating satellite information diminishes, since ground-level geometry alone already provides sufficient coverage. Finally, our BEV training data is sourced primarily from USA city regions, which limits the model's generalization to geographic areas with different styles, satellite characteristics, or geography. There can also be inconsistencies in the satellite imagery itself; for example, if there has been a long temporal gap between satellite acquisition and ground-level capture. We illustrate such a case in Fig. 6.

Figure 7. **Qualitative baseline comparisons.** Additional qualitative comparisons of various baselines on our georeferenced Tanks and Temples dataset.

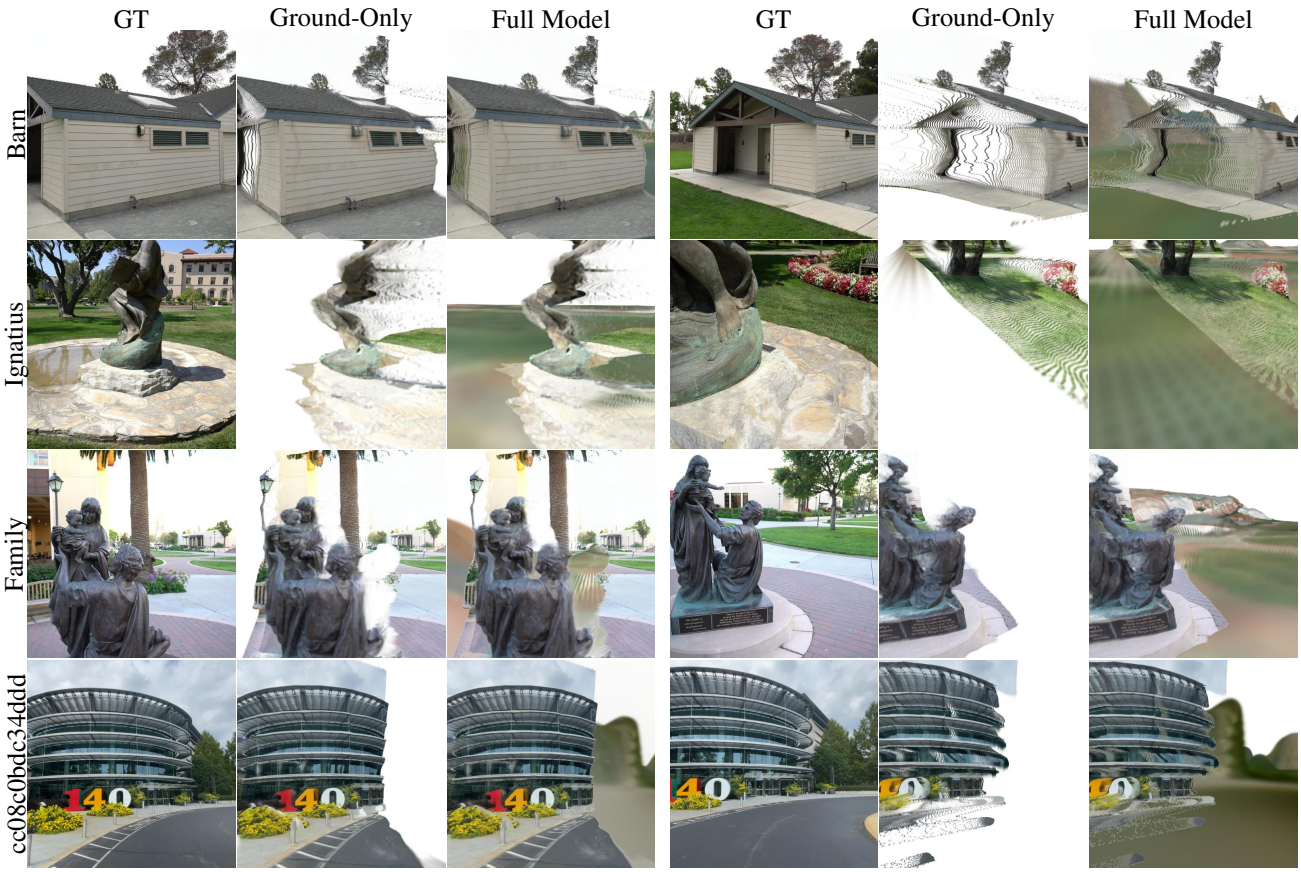

Figure 8. **Cross-View Splatter Ground-Only vs Full-Model**. We visualize the benefit of Cross-View Splatter's satellite branch on qualitative rendering on the Tanks and Temples and DL3DV benchmarks. Our Full-Model achieves better coverage and completeness compared to ground only imagery in sparse-view settings.

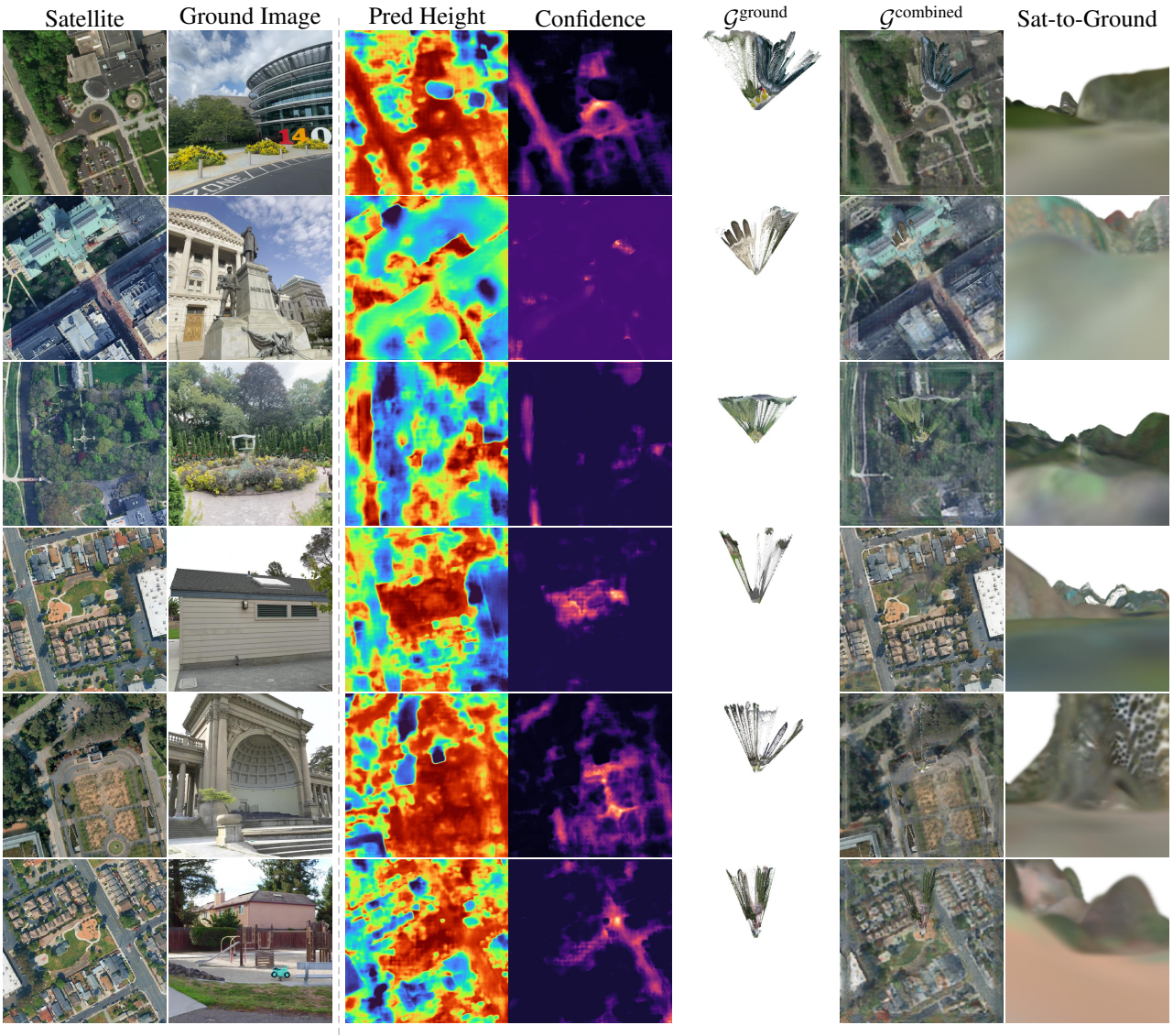

Figure 9. **Cross-View Splatter satellite qualitative**. We show visuals of our full-model satellite head predictions on our benchmark scenes. The first two rows are the inputs to the model, i.e. a BEV perspective and a ground level image. We predict height maps, confidence values, ground level splats, and satellite splats that can then be rendered to ground level views.

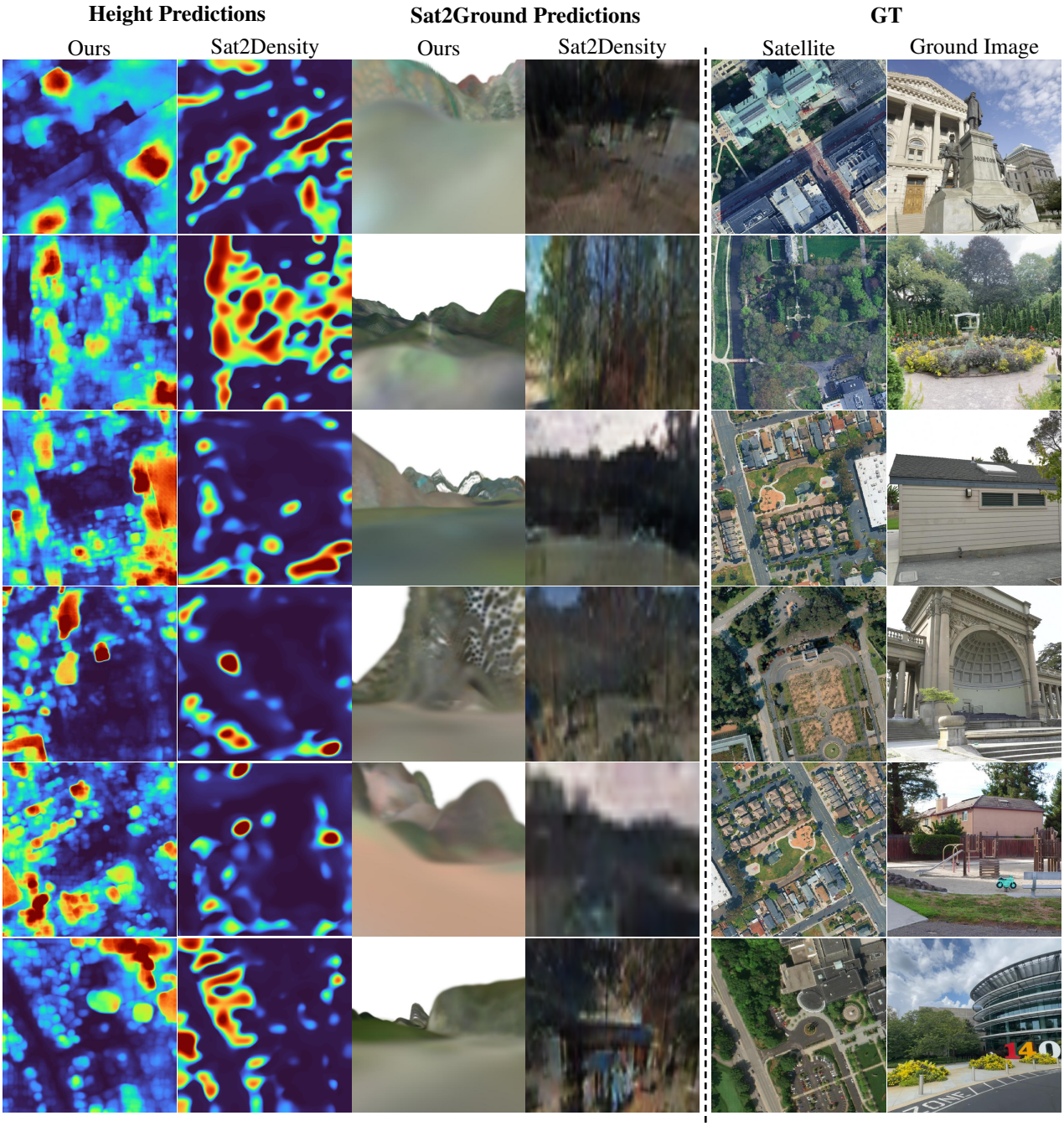

Figure 10. **Sat2Density [10] comparison.** We compare our predictions (Columns 1-4) with Sat2Density height estimates and Sat2Density ground renders against the Ground Truth inputs (Columns 5-6). Both models get the same satellite image and ground image as inputs.

References

[1] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MASt3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *3DV*, 2025. 1

[2] GDAL Developers. Gdal: Geospatial data abstraction library. https://gdal.org, 2024. 4

[3] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *arXiv preprint arXiv:2505.23716*, 2025. 1, 4

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015. 1

[5] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *TOG*, 2017. 5

[6] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 5

[7] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 4

[8] Mapillary. Mapillary Metropolis Dataset. https://www.mapillary.com/dataset/metropolis. Accessed: 2025-10-18. 4

[9] Ming Qian, Bin Tan, Qiuyu Wang, Xianwei Zheng, Hanjiang Xiong, Gui-Song Xia, Yujun Shen, and Nan Xue. Seeing through satellite images at street views. *arXiv preprint arXiv:2505.17001*, 2025. 1, 5

[10] Ming Qian, Jincheng Xiong, Gui-Song Xia, and Nan Xue. Sat2density: Faithful density learning from satellite-ground image pairs. In *ICCV*, 2023. 1, 5, 10

[11] Stanislaw Szymanowicz, Jason Y. Zhang, Pratul Srinivasan, Ruiqi Gao, Arthur Brussee, Aleksander Holynski, Ricardo Martin-Brualla, Jonathan T. Barron, and Philipp Henzler. Bolt3D: Generating 3D Scenes in Seconds. In *ICCV*, 2025. 5

[12] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025. 1

[13] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, 2025. 1

[14] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. 1

[15] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1

[16] Botao Ye, Sifei Liu, Haofei Xu, Li Xueting, Marc Pollefeys, Ming-Hsuan Yang, and Peng Songyou. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *ICLR*, 2025. 1

[17] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *JMLR*, 2025. 1

[18] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *ECCV*, 2024. 5

[19] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *CVPR*, 2025. 5

[20] Jensen (Jinghao) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. 5

[21] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *CVPR*, 2021. 4

[22] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In *ICCV*, 2025. 5