Geo^2: Geometry-Guided Cross-view Geo-Localization and Image Synthesis

Supplementary Material

We provide additional details about the design and parameter settings of Geo^2, along with extended quantitative and qualitative results. The supplementary material is organized as follows:

- Section 6 **Implementation Details.** We describe the model architecture, experimental environment, and training parameters in more detail.
- Section 7 **Evaluation Protocol and Baselines.** We elaborate on the evaluation protocols used in the same-area, cross-area, and cross-dataset settings. More detailed descriptions of the baselines are also provided.
- Section 8 **Ablation Analysis.** We present additional ablation results for our joint training framework as described in Algorithm 1, as well as an analysis of different pre-trained backbones.
- Section 9 **E2P Transformation.** Since GFMs such as VGGT typically require perspective inputs, we explain how equirectangular panoramas are converted into perspective views.
- Section 10 **Discussion of PSNR and SSIM.** We provide a detailed discussion on the limitations of using PSNR and SSIM to evaluate the quality of synthesized images.
- Section 11 **Discussion on Bi-directional Synthesis.** We discuss the bi-directional synthesis process in detail.
- Section 12 **CVGL Augmentation with Synthesized Images.** We explore the downstream task of applying our Geo^2 to augment the training of existing cross-view geo-localization models.
- Section 13 **Robustness of VGGT on Different Angles.** We discuss how the VGGT model is robust to ground image angle changes.
- Section 14 **Efficiency Analysis** We analyze the efficiency of our Geo^2.
- Section 15 **Qualitative Comparison.** We provide additional qualitative results for the CVIS task.

6. Implementation Details

In this section, we provide more details about our hyperparameter settings, experimental environment, and model architecture. We will open-source our code upon acceptance.

Experiment Setup. For CVGL, we set the temperature in the InfoNCE loss $\mathcal{L}_{GL}$ to $\tau = 0.07$, a value commonly used in previous works [8, 53, 54]. The embedding dimension for both ground and satellite features is set to 1024, following prior work [8]. During joint cross-view training, we train GeoMap for $T_1 = 50$ epochs, and GeoFlow for $T_2 = 500$ epochs. We then jointly fine-tune both models for an additional 50 epochs using the consistency loss $\mathcal{L}_{KL}$. The learning rates are set to $\eta_1, \eta_3 = 1\text{e}{-4}$ for GeoMap, and $\eta_2 = 2\text{e}{-4}$ for GeoFlow. GeoFlow is trained on images with a resolution of 256×256. We use a batch size of 128 for joint training. All experiments are conducted on NVIDIA A100 GPUs. The architectures of GeoMap and GeoFlow are described below.

GeoMap Implementation Details. We use two pre-trained backbones to extract geometry and appearance features from the input ground and satellite images. For geometry features, we use VGGT [36], and for appearance features, we use the ConvNeXt [21] backbone from Sample4Geo [8]. Both the VGGT and ConvNeXt backbones are kept frozen during training, and only the convolutional and attention layers in the GeoMap head are learnable. GeoMap is also compatible with other backbone architectures, and we provide preliminary results with alternative backbones in Section 8. We emphasize that the key of Geo^2 is leveraging geometric priors to embed ground and satellite images into a geometry-aware latent space, which facilitates cross-view geo-spatial tasks. Although we use VGGT and Sample4Geo as backbones in our implementation, Geo^2 is not restricted to these two architectures.

For satellite images, we use the dense feature map from VGGT's DPT head, where the feature $t^s \in \mathbb{R}^{C\times H_1\times W_1}$ has channel dimension $C = 256$ and spatial dimensions $H_1, W_1 = 518$. A convolutional layer is used to downsample t^s to $t^{s\prime} \in \mathbb{R}^{D\times H_1'\times W_1'}$, where $D = 1024$ is the embedding dimension. We then extract the appearance feature using ConvNeXt, $q^s \in \mathbb{R}^{D\times H_2\times W_2}$, where $D = 1024$ and $H_2, W_2 = 12$. The downsampled VGGT feature $t^{s\prime}$ and the ConvNeXt feature q^s are flattened along the spatial dimensions and treated as tokens of dimension 1024. We use q^s as the query token and perform cross-attention, with the number of heads set to 16. The attention output is mapped to the final embedding $f^s \in \mathbb{R}^{1024}$ via mean pooling followed by Layer Normalization.

We note that GeoMap uses two branches to process ground and satellite images separately. The ground branch is mostly symmetric to the satellite branch. The only difference is that, for ground images, the ConvNeXt features $q^g \in \mathbb{R}^{D\times H_2\times W_2}$ have spatial dimensions $H_2 = 4$ and $W_2 = 24$. Additionally, since we split the ground images into four perspective views to extract the VGGT features, the ground features $t^g \in \mathbb{R}^{V\times C\times H_1\times W_1}$ include an additional view dimension $V = 4$. We defer the details about the perspective transformation to Section 9. The convolutional layer is applied to each view independently, and the four-view downsampled ground features are concatenated along the width dimension. Cross-attention is computed in

Table 7. Panorama-to-satellite configuration across VIGOR, CVUSA, and CVACT datasets.

Dataset	#Ground Image	Ground Image Res.	#Satellite Image	Satellite Image Res.	Vertical FOV	Ground Crop Res.
CVUSA	44,416	1232×224	44,416	750×750	∼90°	224×224
CVACT	128,334	1664×832	128,334	1200×1200	180°	416×416
VIGOR	105,214	2048×1024	90,618	640×640	180°	512×512

the same way as in the satellite branch.

GeoFlow Implementation Details. The backbone of our GeoFlow follows that of RAE [57], which adopts DiT [56] with a DDT [39] head. We use the pretrained RAE encoder (DINOv2-B) and decoder to encode images into the latent space and decode the latent representation back into image space. Specifically, the image is encoded into a representation of size $16 \times 16 \times 768$, with the patch size set to 1. The depth of DiT is 28, and the hidden size is 1152. The DDT head consists of two layers with a hidden size of 2048. Both DiT and DDT use 16 attention heads. The model is trained end-to-end using the loss $\mathcal{L}_{IG}$.

7. Evaluation Protocol and Baselines

7.1. Evaluation Protocol

In Table 7, we provide more detailed information about the datasets used for training and evaluation. Both CVUSA and CVACT provide one-to-one ground-to-satellite matches, i.e., only one positive satellite image serves as the reference for each ground query. VIGOR, on the other hand, offers one-to-many ground-to-satellite matches, where multiple satellite images are considered positive references for each ground query. The VIGOR dataset is split into four cities: Chicago, New York, San Francisco, and Seattle. Following prior works [8, 53, 54], our evaluation for CVGL is conducted under three different settings: same-area, cross-dataset, and cross-area. In the same-area setting, the training and testing data are from the same geographical area. In the cross-dataset setting, we train the model on CVUSA and evaluate on CVACT, or vice versa. In the cross-area setting, we train on New York and Seattle from VIGOR and evaluate on Chicago and San Francisco. The cross-dataset and cross-area settings better reflect the generalizability of geo-localization models.

7.2. Choice of Baselines

Cross-View Geo-Localization. We choose LPN [40], SAFA [29], TransGeo [59], GeoDTR [53], SAIG-D [60], and Sample4Geo [8] to provide a comprehensive comparison against the state-of-the-art in cross-view geo-localization. These methods represent a clear progression, from early deep learning approaches (LPN and SAFA) to more recent high-performing techniques. Specifically, we include TransGeo to benchmark against early transformer-based methods in this domain, and GeoDTR and SAIG-D to compare against models that incorporate more advanced attention and alignment mechanisms. Finally, Sample4Geo serves as the most contemporary and challenging benchmark, allowing us to validate the performance of our proposed method against the latest standard.

Table 8. Albaltion of CVGL task on CVACT dataset. "+Geometry" denotes model incorporating geometry features from VGGT are used. "$+\mathcal{L}_{KL}$" denotes model finetuned with consistency loss.

Dataset	Approach	R@1	R@5	R@10	R@1%
CVACT Val	Baseline	90.53	96.61	97.53	98.78
	+Geometry	93.81	97.02	97.77	98.92
	$+\mathcal{L}_{KL}$	94.36	97.41	97.97	99.05
CVACT Test	Baseline	71.51	92.42	94.45	98.70
	+Geometry	74.37	93.24	95.02	98.86
	$+\mathcal{L}_{KL}$	75.08	94.89	95.77	99.01

Satellite-to-Ground Synthesis. Sat2Density [26], ControlNet [50], and CrossViewDiff [5] are selected to provide a comprehensive comparison against state-of-the-art methods in satellite-to-ground cross-view image synthesis. We include Sat2Density and CrossViewDiff as they are recent methods that directly address satellite-to-ground image generation, establishing relevant domain-specific baselines. The inclusion of ControlNet enables us to evaluate whether our Geo^2 method offers advantages in fidelity and accuracy over a general-purpose conditional diffusion framework adapted to this task.

Ground-to-Satellite Synthesis. We select X-Seq [28], AerialDiff [14], GPG2A [2], ControlNet [50], and SkyDiffusion [46] to provide a comprehensive evaluation against state-of-the-art methods in ground-to-satellite image synthesis. We include specialized cross-view methods such as X-Seq, GPG2A, AerialDiff, and SkyDiffusion to establish strong baselines. Additionally, the inclusion of ControlNet helps assess the benefit of our specialized approach over general-purpose conditional generation frameworks, ensuring a robust validation of Geo^2.

8. Ablation Analysis

In this section, we validate the effectiveness of the geometric prior and the joint training strategy used in Geo^2.

Joint Training. As shown in Table 8, we first evaluate the effectiveness of the proposed methods on the CVGL task. Our baseline is the pre-trained Sample4Geo [8] model, which does not incorporate any geometric prior. GeoMap introduces geometric priors from VGGT into the baseline model, enhancing the geometric information in both ground

Table 9. Albaltion of CVIS task on CVACT dataset. "+Geometry" denotes model incorporating geometry features from VGGT. "+$\mathcal{L}_{KL}$" denotes model finetuned with consistency loss. G2S is ground-to-satellite synthesis, and S2G is satellite-to-ground synthesis.

Direction	Approach	FID (↓)	LPIPS (↓)	PSNR (↑)	SSIM (↑)
G2S	Baseline	37.51	0.613	13.17	0.143
	+Geometry	33.02	0.568	14.08	0.155
	+$\mathcal{L}_{KL}$	31.72	0.552	14.62	0.162
S2G	Baseline	33.41	0.573	11.08	0.377
	+Geometry	30.86	0.511	11.93	0.408
	+$\mathcal{L}_{KL}$	27.77	0.483	13.57	0.457

and satellite embeddings. This leads to improved retrieval accuracy on both the validation and test splits of the CVACT dataset. By fine-tuning GeoMap with the consistency loss $\mathcal{L}_{KL}$ during joint CVGL-CVIS training (Algorithm 1), the retrieval accuracy is further improved.

Similarly, we analyze the effectiveness of our proposed method on the CVIS task in Table 9. The baseline in this case is a Flow Matching model without any additional conditioning. By conditioning the Flow Matching model on the geometry-aware embeddings from GeoMap, we observe that both ground-to-satellite (G2S) and satellite-to-ground (S2G) synthesis improve by a noticeable margin. Adding the consistency loss further improves generation quality, especially in the S2G direction. Most importantly, the consistency loss $\mathcal{L}_{KL}$ effectively aligns the ground and satellite embeddings. These consistent, geometry-aware embeddings are shown to mutually benefit both the CVGL and CVIS tasks.

Different Backbones. In Geo^2, we adopt multiple backbones to extract features. Specifically, in GeoMap, we use VGGT [36] to extract geometric features, and a pretrained ConvNeXt [21] model from Sample4Geo [8] to extract semantic features. In GeoFlow, we use a pretrained DINOv2-B-based RAE encoder/decoder [57]. In this section, we analyze the potential impact of using different backbones.

As shown in Table 10, we compare the performance of the CVGL task when using different semantic backbones. Specifically, we replace Sample4Geo with TransGeo [59], while keeping all other components unchanged. We evaluate geo-localization performance on the VIGOR dataset under both the same-area and cross-area settings. Although using a stronger backbone generally improves performance, the geometric prior introduced by GeoMap consistently enhances retrieval accuracy even when used with the TransGeo backbone. This demonstrates the general applicability of incorporating geometric information into CVGL models.

For the geometry backbone, potential alternatives include DUSt3R, MASt3R, and AerialMegaDepth [16, 35, 38]. However, these methods are based on pairwise registration and rely on a process called global alignment

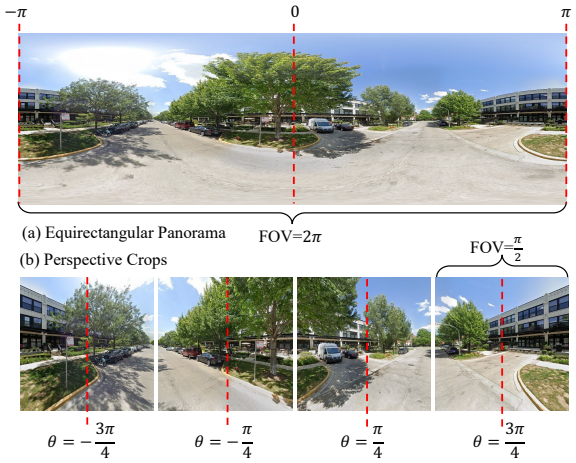


Figure 9. Illustration of the Equirectangular-to-Perspective transformation. Perspective-view crops are obtained by placing cameras with different yaw angles θ.

Table 10. Ablation on different backbones on VIGOR dataset. Ours† denotes GeoMap with TransGeo as the semantic backbone, and ours denotes GeoMap with Sample4Geo as backbone.

Dataset	Approach	R@1	R@5	R@10	R@1%	Hit Rate
Same-area	TransGeo	61.48	87.54	91.88	99.56	73.09
	Sample4Geo	77.86	95.66	97.21	99.61	89.82
	Ours†	68.33	91.06	93.14	99.59	76.34
	Ours	81.59	96.53	98.62	99.68	90.35
Cross-area	TransGeo	18.99	38.24	46.91	88.94	21.21
	Sample4Geo	61.70	83.50	88.00	98.17	69.87
	Ours†	29.56	46.71	51.35	91.22	30.07
	Ours	66.71	87.34	91.02	98.25	72.13

to align multiple input views [38]. Global alignment is known to be time-consuming [36, 38], which makes it difficult to incorporate these methods into cross-view training tasks. Designing efficient GFMs is an active research direction [34, 37], and incorporating improved GFM backbones could potentially boost the performance of Geo^2. Similarly, it is possible to replace the RAE encoder/decoder with VAE-based [12] counterparts. However, we remark that the core focus of Geo^2 is to explore the potential of geometric priors in cross-view tasks, rather than designing new GFMs or autoencoders. Therefore, we leave the exploration of alternative GFMs and autoencoders as future work.

9. E2P Transformation

The Equirectangular-to-Perspective (EP2) transformation is important because most GFMs are trained exclusively on perspective images [16, 36–38]. In cross-view datasets, however, the ground images are equirectangular panoramas. As illustrated in Figure 10 (b), directly cropping panoramas and feeding them into GFMs often leads to inaccurate geometry. In contrast, with the Equirectangular-to-

Perspective transformation, GFMs can reconstruct reliable geometry even when the perspective crops have no overlap, as shown in Figure 10 (a).

In this section, we provide more details about how ground panoramas are transformed into perspective-view crops, as shown in Figure 9. Given a ground panorama with width W and height H, pixels (u, v) are represented in equirectangular coordinates, where $u \in [0, W-1]$ and $v \in [0, H-1]$. For ground panoramas, each rectangular pixel (u, v) can also be represented by a spherical coordinate (λ, ϕ), where $\lambda \in [-\pi, \pi]$ is the longitude and $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the latitude. The conversion between (u, v) and (λ, ϕ) is given by:

$$u = \frac{W}{2\pi}(\lambda + \pi), v = \frac{H}{v_{\text{range}}}\left(\frac{v_{\text{range}}}{2} - \varphi\right)$$

where v_{range} is the vertical FOV of the panoramas. The values of the parameters differ between datasets and are detailed in Table 7.

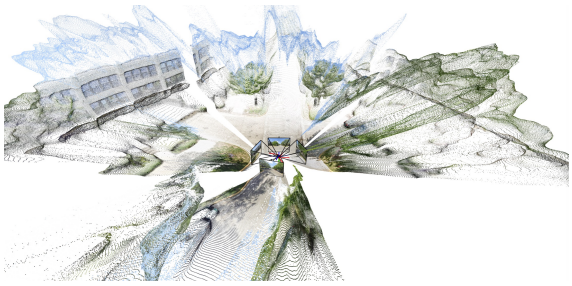
To transform the ground panorama into perspective-view crops, we split the spherical coordinates along the horizontal direction. To align with the VGGT input, we define each perspective-view crop as an image captured by a simple pinhole camera, where the horizontal and vertical FOVs are set to $\frac{\pi}{2}$. Accordingly, we split the longitude $\lambda \in [-\pi, \pi]$ into four non-overlapping crops, i.e., $[-\pi, -\frac{\pi}{2}]$, $[-\frac{\pi}{2}, 0]$, $[0, \frac{\pi}{2}]$, and $[\frac{\pi}{2}, \pi]$. Each crop corresponds to a camera defined by the yaw θ (horizontal rotation) and pitch ϕ (vertical rotation). Since we only need to split the longitude, we set the pitch for all crops to 0 and set the yaw to the center of each crop, i.e., $-\frac{3\pi}{4}$, $-\frac{\pi}{4}$, $\frac{\pi}{4}$, and $\frac{3\pi}{4}$, as illustrated in Figure 9 (b). For each camera, given the yaw θ and pitch ϕ, the rotation matrix can be computed as follows:

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix},$$

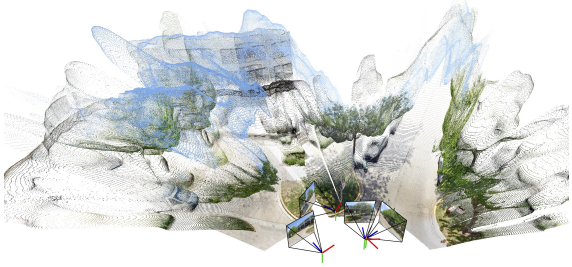
$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}.$$

Given the rotation matrices above, the final rotation matrix of each camera is defined as $R = R_x(\phi)R_y(\theta)$. With the rotation matrix, we show how pixels in the perspective view (x, y) are mapped to pixels in the equirectangular coordinates (u, v). For each pixel in the perspective view (x, y), a ray r from the pinhole camera can be defined as:

$$\mathbf{r}(x, y) = \begin{bmatrix} x\tan\left(\frac{f_h}{2}\right) \\ y\tan\left(\frac{f_v}{2}\right) \\ 1 \end{bmatrix}.$$



(a) Reconstruction with E2P Transformation



(b) Reconstruction without E2P Transformation

Figure 10. Illustration of (a) accurate reconstruction from perspective crops, and (b) inaccurate reconstruction from directly cropped panoramas without E2P transformation.

where the FOVs f_h and f_v are set to $\frac{\pi}{2}$, as mentioned above. To map the ray $\mathbf{r}$ to spherical coordinates, we first normalize it and then apply the rotation as:

$$\mathbf{d} = R\frac{\mathbf{r}}{\|\mathbf{r}\|}.$$

where $\mathbf{d} = (d_x, d_y, d_z)$ is a unit vector in 3D space, which can be converted into spherical coordinates by:

$$\lambda = \arctan 2(d_x, d_z), \varphi = \arcsin(d_y).$$

The spherical coordinates (λ, ϕ) can be easily converted into the equirectangular coordinates (u, v), as mentioned above. Finally, we note that since the converted coordinates (u, v) are not necessarily integers, bilinear sampling is used to compute the final pixel value at (x, y) in the perspective view.

10. Discussion of PSNR and SSIM

As reported in Tables 5 and 6, our Geo^2 attains competitive performance across most evaluation metrics, though its PSNR and SSIM scores are relatively lower. This behavior is expected and aligns with extensive prior findings that PSNR and SSIM scores correlate poorly with perceptual fidelity in generative tasks [52]. Recent studies in video interpolation [23, 24] further demonstrate that these pixel-based metrics fail to reflect structural or semantic correctness.

In cross-view image synthesis, the limitation is even more pronounced because the target view is not strictly pixel-aligned with the input. As a result, small variations

Figure 11. Visualization of two satellite images generated from the same ground image from our Geo² and GPG2A [2], respectively, on the VIGOR dataset. Although the results demonstrate similar PSNR and SSIM values, our generated satellite image is more visually and geometrically similar to the ground truth satellite image than GPG2A's result. From left to right are the ground truth satellite image, GPG2A's generated satellite image, our Geo²'s generated satellite image, and the input ground image.

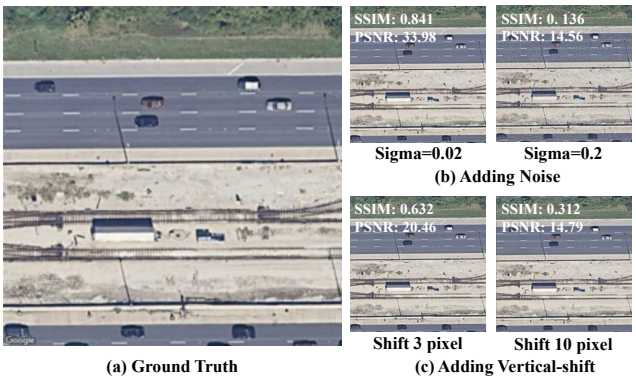


Figure 12. Visualization of the change of the PSNR and SSIM values with respect to the different levels of noise and vertical pixel shift.

in object color (e.g., buildings and roads), transient objects (e.g., vehicles and pedestrians), and sky appearance disproportionately penalize PSNR and SSIM scores despite preserving geometric structure. To illustrate this effect, Figure 11 shows two ground-to-satellite synthesized results from GPG2A [2] and our Geo², respectively, from the VIGOR dataset. As observed from this figure, our Geo² produces a more geometrically consistent and semantically faithful satellite image compared to the ground truth than GPG2A. However, their PSNR and SSIM values are close to each other. This confirms that these metrics are insufficient indicators of cross-view synthesis quality and motivates our use of perceptual and structure-oriented measures.

To further demonstrate this point, we conduct an experiment to visualize the change in SSIM and PSNR values with respect to increasing amounts of Gaussian noise and vertical pixel shift, as shown in Figure 12. As we can observe, with the increase in noise and the number of shifted pixels, both SSIM and PSNR decrease drastically. However, the degraded images still maintain strong geometric and visual similarity to the ground truth image, again validating that SSIM and PSNR are not the best choices for evaluating image synthesis quality.

11. Discussion on Bi-directional Synthesis

As discussed in Section 3.3, our GeoFlow only needs to train once, while it can perform both ground-to-satellite and satellite-to-ground synthesis, differentiating itself from existing works, such as X-fork [28], GPG2A [2], CrossViewDiff [5], SkyDiffusion [46], and Sat2Density [26]. In Section 3.3, we explained the mechanism to reverse the process in sampling without re-training. In this section, we further explain that training in two directions (ground-to-satellite and satellite-to-ground) is equivalent to each other. Recall that the vector field $v = x^s - x^g$, which is pointing to the satellite direction. Consider another vector field $v' = x^g - x^s$, which is pointing to the ground direction. In fact, v and v' have the same magnitude but point in exact opposite directions. Recall that our loss $\mathcal{L}_{IG}$ is defined as,

$$\mathcal{L}_{IG} = \|G_\theta(x_t, t, c) - v\|_2.$$

Thus, this indicates that training with either v or v' results in the same G_θ but to predict opposite directions. Consequently, it is simple to invert the predicted vector field from G_θ to have an estimation of both directions with a single model. For instance, if training with v' instead of v, one can rewrite the equation as,

$$x^g = x^s + \int_0^1 G'_\theta(x_t, t, c)\, dt,$$

and,

$$x^s = x^g - \int_0^1 G'_\theta(x_t, t, c)\, dt,$$

where $G'_\theta(x_t, t, c)$ is the new model which is trained to predict v'. One can easily reverse the equations in Section 3.3 to the equations above to obtain the predicted ground and satellite images, respectively.

12. CVGL Augmentation with Synthesized Images

To further explore the downstream tasks that our proposed Geo² can benefit, following prior studies [2], we adopt Sample4Geo [8] as a baseline and augment the training set with our generated images. The results are summarized in Table 11, which illustrates that our Geo² can further improve

Table 11. Evaluation of augmenting Cross-View Geo-Localization methods by using generated images from the proposed Geo² on CVACT [19] Val and CVACT Test sets.

	CVACT Val				CVACT Test			
Approach	R@1	R@5	R@10	R@1%	R@1	R@5	R@10	R@1%
Baseline	90.35	96.61	97.53	98.78	71.51	92.42	94.45	98.70
Baseline + Augmentation	91.56	96.94	97.71	98.83	72.83	93.15	94.74	98.81

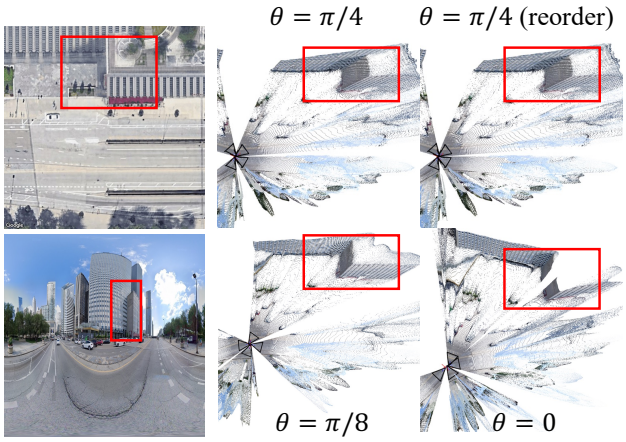


Figure 13. VGGT [36] reconstructions on VIGOR images under varying orientation angles and input orders.

Table 12. Ablation study of different base navigation angle θ on CVACT Val set.

θ	R@1	R@5	R@1%
π / 4	94.36	97.41	99.05
π / 8	94.25	97.42	99.05
0	94.31	97.43	99.04

CVGL performance of Sample4Geo [8](Baseline) on both CVACT [19] Val and challenging CVACT Test sets.

13. Robustness of VGGT on Different Angles

As shown in Fig. 13, different input orders produce almost identical output, and different base navigation angles θ produce similar reconstructions. To further demonstrate this, we ablate on base navigation angles as shown in Table 12, which shows that different angles yield similar performance on the CVACT Val set.

14. Efficiency Analysis

We first examine the efficiency of our GeoMap module. The computation cost of it depends on the number of input views for VGGT [36]. As shown in Tab. 13, since ground views are four times satellite views, the cost is quadrupled. We then analyze the efficiency of our GeoFlow module by varying the steps of its ODE solver. The results are summarized in Table 14. In the main paper, we use a 10-step ODE solver. However, as shown in Table 14, sampling steps can be reduced, where 5 steps preserve synthesis quality and 2 steps are fast with only minor degradation. Moreover, distilled [7] and one-step flow-matching models [13] can be used in our framework for higher efficiency. Similarly, using a smaller VGGT (e.g., the upcoming 200M variant) can also reduce computational cost. However, these optimizations are orthogonal to our work.

Table 13. Efficiency (in GFLOPS) and number of Parameters of our Geo².

Module	Sat.	Grd.	#Params
Conv Module	45	31	88M
VGGT	1387	5548	942M
GeoMap	40	158	43M

Table 14. Ablation on ODE steps on CVACT Val set.

Steps	FPS	GFLOPs	S2G FID/LPIP		G2S FID/LPIP	
2	3.21	1143	29.09	0.4896	39.69	0.5594
5	1.38	2857	27.82	0.4831	33.65	0.5530
10	0.71	5715	27.77	0.4833	31.72	0.5520

15. More Qualitative Visualization

In this section, we provide more visualizations of CVGL on the CVACT dataset in Figure 14, where our method generally achieves more accurate retrieval. We also provide additional qualitative CVIS results of Geo² on the CVUSA, CVACT, and VIGOR datasets in Figure 15, Figure 16, and Figure 17, respectively. These visualizations further demonstrate that the GeoFlow module enables our model to synthesize both satellite-view and ground-view images that preserve global scene geometry while producing realistic textures consistent with the target domain. Across diverse environments, Geo² successfully captures appearance and structural correspondences between satellite and ground views, corroborating the observations discussed in the main manuscript.

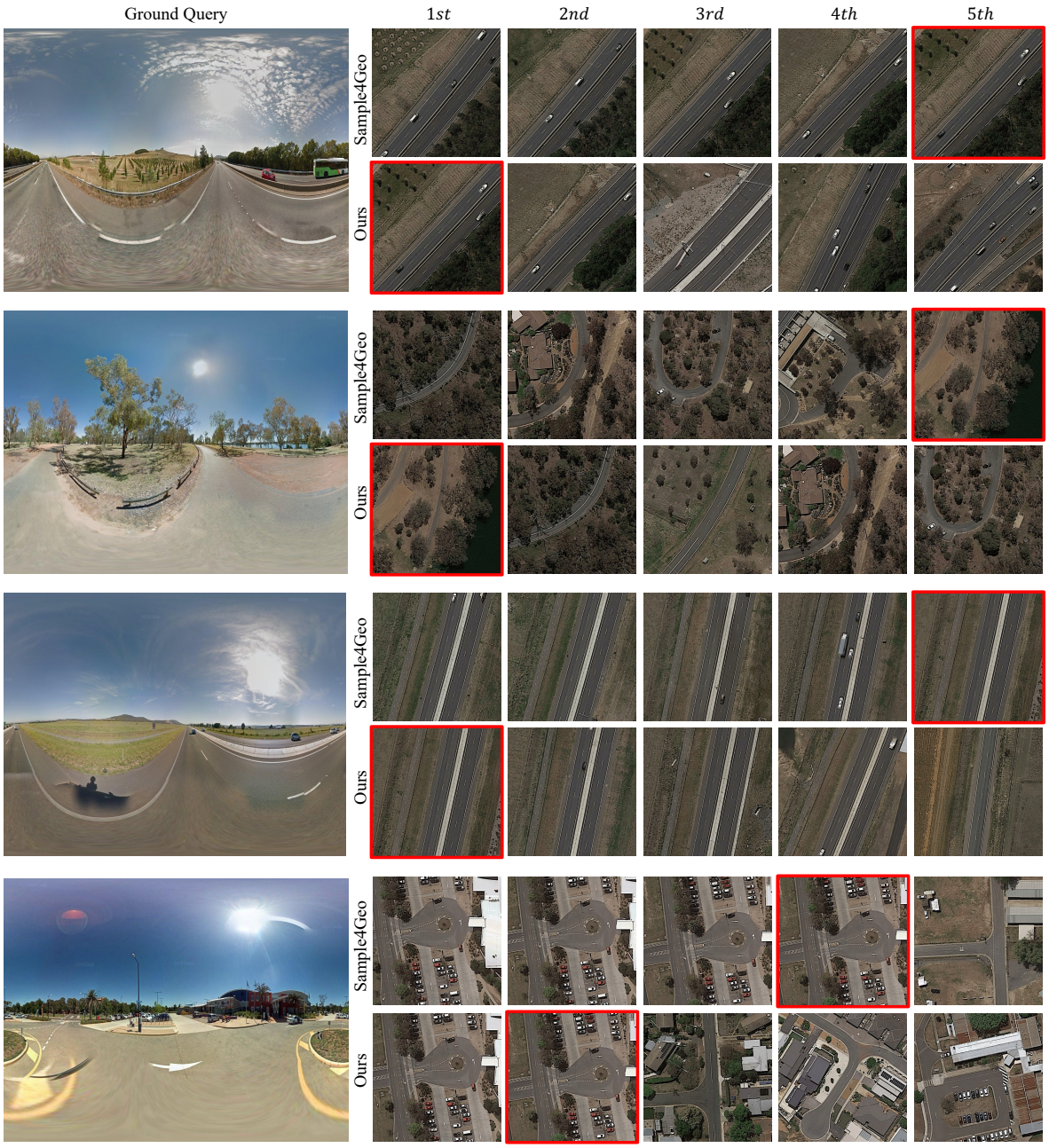


Figure 14. More Visualizations of Top-5 Retrieval Results from the CVACT Dataset. We compare the top-5 retrieval results of our method with the Sample4Geo [8] baseline. The query ground images are shown on the left, and the retrieval results are shown on the right.

Figure 15. More Visualization of Generated images from Geo^2 on CVUSA dataset. From left to right are the ground truth satellite image, the ground truth ground image, the Ground-to-Satellite generated image, and the Satellite-to-Ground generated image.

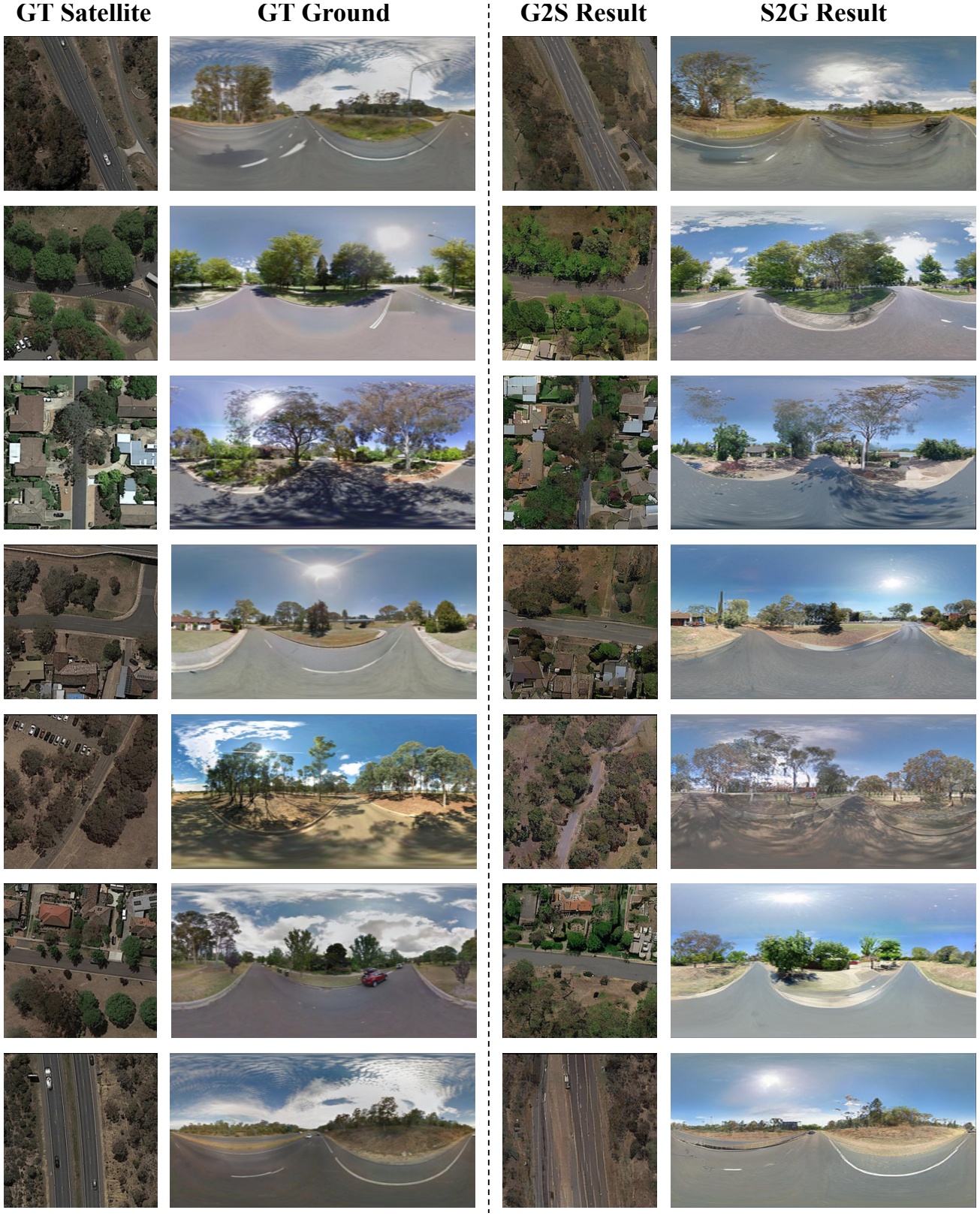


Figure 16. More Visualization of Generated images from Geo^2 on CVACT dataset. From left to right are the ground truth satellite image, the ground truth ground image, the Ground-to-Satellite generated image, and the Satellite-to-Ground generated image.

Figure 17. More Visualization of Generated images from Geo^2 on VIGOR dataset. From left to right are the ground truth satellite image, the ground truth ground image, the Ground-to-Satellite generated image, and the Satellite-to-Ground generated image.