# GeoFlow: Real-Time Fine-Grained Cross-View Geolocalization via Iterative Flow Prediction

## Supplementary Material

In this supplementary material, we provide additional technical and experimental details, organized as follows:

- **Implementation Details** (Sec. A). We offer more details about our final model architecture, training routines, hyperparameters and IRS parameters
- **Feasibility of 3-DoF Extension** (Sec. B). We present the methodology and quantitative results demonstrating GeoFlow's capability to generalize to the full 3-DoF pose estimation task with minimal overhead.
- **Backbone Capacity and Efficiency Trade-Off** (Sec. C). We quantify the trade-off between model size and accuracy by comparing two EfficientNet backbones, justifying our final design choice for real-time deployment.
- **Efficiency Analysis and Deployment Considerations** (Sec. D). We analyze GeoFlow's computational efficiency, including its decoupled inference design, unified efficiency benchmark, and implications for real-world edge deployment.
- **Uncertainty and Convergence Dynamics** (Sec. E). We offer visualizations showing how the IRS algorithm refines predictive uncertainty and builds confidence.
- **Additional Qualitative Results** (Sec. F). We provide extensive visual evidence of the learned regression field, showcasing refinement trajectories in both same-area and cross-area settings.
- **Discussion of Failure Cases** (Sec. G). We provide an analysis of specific failure modes (e.g., highly vegetated areas) that contribute to the shift in the error distribution.
- **Societal Impact and Limitations** (Sec. H). We provide the required discussion on the broader context, efficiency advantages, and practical limitations of our system.

## A. Implementation and Training Details

**Implementation Details:** GeoFlow is implemented in PyTorch [2]. For both $\psi_g$ and $\psi_s$, we use an EfficientNet-B0 [3] backbone, initialized with pre-trained ImageNet weights. The cross-attention module (Sec. 3.1 of the manuscript) is a standard multi-head attention layer with 4 heads. The coordinate projection layer embeds the 2D hypothesis into a 16-dimensional vector. This is concatenated with the 128-dimensional visual representation $\mathbf{f}_{vis}$, creating a 144-dimensional input for the regression MLPs.

**Training Details:** Our model is trained end-to-end for 200 epochs by optimizing the combined NLL loss (Eq. 11) on a single NVIDIA H100 GPU. We employ the AdamW optimizer [1] with a batch size of 80 and a weight decay of $1\times10^{-2}$. We use a differential learning rate: the pre-trained EfficientNet backbones are fine-tuned with a low learning rate of $1 \times 10^{-4}$, while the rest of the network is trained with a higher learning rate of $1 \times 10^{-3}$. For both VIGOR and KITTI, we resize the ground image $\mathbf{I}_g$ to $256 \times 1024$ and the aerial image $\mathbf{I}_s$ to $512 \times 512$. During training, the initial hypothesis $\mathbf{q}_0$ is formed by uniformly sampling a 2D translation $(x_0, y_0)$ from the valid spatial range. To ensure consistency across the model, all spatial inputs and outputs, including the initial hypothesis $\mathbf{q}_0$, the target location $\mathbf{q}_1$, are normalized to the continuous range $[-1, 1]$ relative to the reference satellite map. During inference, all reported inference results and comparisons (accuracy and FPS) are measured on a single NVIDIA V100 GPU to ensure a fair comparison with existing literature.

**Inference and IRS Parameters:** During inference, we use our Iterative Refinement Sampling (IRS) algorithm (Sec. 3.4 in the main paper) to ensure a robust prediction. Unless otherwise stated, we adopt the following default parameters for evaluation: we initialize $N = 10$ candidate poses (seeds) by sampling uniformly from the pose space. We then perform $R = 5$ iterative refinement rounds, where each of the $N$ poses using Eq. 12. The final location estimate is computed as the mean of all $N = 10$ refined poses after the final round.

## B. Extending GeoFlow to Full 3-DoF

**The primary objective of this feasibility study is to demonstrate the inherent modularity and generalizability of the GeoFlow architecture**. We show its capability to seamlessly extend from 2-DoF planar localization (translation-only) to the full 3-DoF pose estimation task (translation $\mathbf{g}_{(x,y)}$ and orientation $\gamma$) with minimal architectural changes and negligible computational overhead. This extension validates that the learned cross-view spatial representations are rich enough to capture both metric displacement and angular alignment between ground and satellite views.

### B.1. Context and Clarification of Main Results

For fair comparison with existing state-of-the-art methods, the primary 2-DoF evaluation in the main paper (Table 1 in main paper) uses test data where images contain random orientation shifts ($\pm10°$), following other methods [4, 5]. Moreover, the main 2-DoF model in the paper only estimates the translation vector $(x, y)$, without the orientation prediction itself. In this supplementary study, we extend the model to explicitly predict the vehicle's orientation angle

Table 1. 3-DoF Pose Estimation Comparison on KITTI. GeoFlow demonstrates competitive accuracy against state-of-the-art methods while maintaining a superior computational profile. Best results are in **bold**

| | | Loc. (m) | | Orientation ($\gamma$) Errors ($^\circ$) | | | | Inference |
|---|---|---|---|---|---|---|---|---|
| Dataset | Method | ↓Mean | ↓Median | ↓Mean | ↓Median | ↑R@1 | ↑R@5 | Speed (FPS) |
| **Same Area** | FG$^2$ [4] | **0.75** | **0.52** | **1.28** | **0.74** | **61.17%** | **95.65%** | 4.20 |
| | GeoFlow (3-DoF) | 1.03 | 0.69 | 2.51 | 1.93 | 27.30% | 87.80% | **29.49** |
| **Cross Area** | FG$^2$ [4] | **7.45** | **4.03** | **3.33** | **1.88** | **30.34%** | **81.17%** | 4.20 |
| | GeoFlow (3-DoF) | 8.53 | 5.68 | 3.87 | 2.75 | 20.00% | 72.30% | **29.49** |

$\gamma$, demonstrating GeoFlow's adaptability to the full 3-DoF pose estimation task.

## B.2. Methodology: Architectural Extension

The extension from 2-DoF to 3-DoF is achieved by appending a lightweight orientation prediction head, $\mathbf{v}_\gamma$, to the decoder, thus preserving the core GeoFlow framework.

### B.2.1. Orientation Representation: Unit Circle

Rather than directly regressing the angle $\gamma$, we adopt a unit circle representation by predicting the pair $(\cos\gamma, \sin\gamma)$. This design choice resolves the inherent periodicity and discontinuity problems associated with angular quantities (e.g., the wrap-around at $\pm 10^\circ$). Representing orientation as a point on the unit circle in $\mathbb{R}^2$ ensures continuity, providing smooth and stable gradients for optimization.

The ground-truth orientation $\gamma_{\text{gt}}$ is converted to the unit vector representation

$$\mathbf{g}_\gamma = \begin{bmatrix} \cos(\gamma_{\text{gt}}) \\ \sin(\gamma_{\text{gt}}) \end{bmatrix}, \qquad \text{where} \quad \gamma_{\text{gt}} = \gamma_{\text{gt}} \cdot \frac{\pi}{180}. \tag{1}$$

### B.2.2. Orientation Head and Loss Function

The orientation prediction head $\mathbf{v}_\gamma$ is a simple MLP appended to the decoder: $\mathbf{v}_\gamma : \mathbb{R}^{144} \to \mathbb{R}^{64} \to \mathbb{R}^2$. The output prediction $\mathbf{p}_\gamma = [p_{\cos}, p_{\sin}]$ is $\ell_2$-normalized to $\hat{\mathbf{p}}_\gamma = \mathbf{p}_\gamma / \|\mathbf{p}_\gamma\|_2$ to enforce consistency with the unit circle manifold.

We employ a cosine similarity loss $\mathcal{L}_\gamma$ to measure the angular distance $\Delta\gamma$ between the predicted $\hat{\mathbf{p}}_\gamma$ and ground-truth $\mathbf{g}_\gamma$ unit vectors:

$$\mathcal{L}_\gamma = \mathbb{E}\left[1 - \frac{\hat{\mathbf{p}}_\gamma \cdot \mathbf{g}_\gamma}{\|\hat{\mathbf{p}}_\gamma\| \, \|\mathbf{g}_\gamma\|}\right] = \mathbb{E}\left[1 - \cos(\Delta\gamma)\right]. \tag{2}$$

This loss is stable ($\mathcal{L}_\gamma \in [0, 2]$), differentiable, and correctly handles circular continuity. The final training objective for the 3-DoF model is simply the sum of the original translation losses and the new orientation loss:

$$\mathcal{L}_{\text{3-DoF}} = \mathcal{L}_r + \mathcal{L}_\theta + \mathcal{L}_\gamma. \tag{3}$$

## B.3. Computational Overhead Analysis

The addition of the orientation head results in a minimal parameter increase, demonstrating high architectural modularity. The base 2-DoF model contains 7.38 M parameters, while the extended 3-DoF model contains 7.39 M parameters, representing an overhead of only $0.13\%$ increase. This minimal overhead is entirely contained within the small new orientation MLP. Furthermore, we measured inference time on a single V100 NVIDIA GPU and found no measurable impact on speed. The base and extended models both maintain an inference rate around $29.49$ FPS, confirming that the dominant computation remains in the backbone feature extraction and cross-attention. **Crucially, this demonstrates that the full 3-DoF capability is achieved with negligible computational overhead, reinforcing the core efficiency argument of the GeoFlow framework.**

## B.4. Quantitative Results and Interpretation

Quantitative results in Table 1 demonstrate that GeoFlow can reliably estimate full 3-DoF pose, achieving comparable accuracy against state-of-the-art method FG2 [4] while maintaining a significant advantage in computational efficiency. The 3-DoF model achieves comparable translation accuracy relative to FG2 [4]. For the Same Area, the median localization error is $0.69$m, and for the Cross Area, it is $5.68$m. This confirms that adding the orientation prediction task does not compromise the learned translation features, highlighting the robustness of the GeoFlow architecture. Moreover, GeoFlow achieves a significant breakthrough in efficiency, processing data at 29.49 FPS. This establishes GeoFlow as the fastest full 3-DoF pose estimation model, operating $7.02\times$ faster than competitive methods like FG2 ($4.20$ FPS), a critical advantage for real-time deployment. Furthermore, the orientation prediction results validate the feasibility of extending GeoFlow to estimate heading. On the Same Area, the median orientation error is $1.93^\circ$, with $87.8\%$ of predictions within $5^\circ$ (R@5 success rate). On the Cross Area, the median error is $2.75^\circ$, with a $72.3\%$ R@5 success rate. **This suggests that GeoFlow can provide fair and practical orientation prediction for real-time applications, especially given the significant speed advantage**

**and minimal architectural overhead. These results confirm GeoFlow's suitability for autonomous navigation tasks requiring accurate and rapid position and heading estimation.**

### B.5. Key Insights and Conclusions

This feasibility study provides strong evidence for GeoFlow's architectural flexibility. The model was seamlessly extended to a new task (3-DoF pose) with negligible computation overhead. This confirms that GeoFlow provides an efficient and generalizable framework, with learned cross-view representations rich enough to capture both metric displacement and angular alignment, extending to a complete pose 3-DoF estimation pipeline.

## C. Backbone Capacity and Efficiency Trade-Off

To address potential concerns regarding model capacity, we conducted an ablation comparing our default lightweight backbone (EfficientNet-B0 [3]) against a larger architecture (EfficientNet-B5 [3]). This study quantifies the fundamental trade-off between absolute accuracy and computational overhead.

The results in Table 2 clearly demonstrate the substantial cost required for marginal performance gains. While the larger EfficientNet-B5 backbone is more accurate, providing a slight improvement of $0.08$m in Median Error (Same Area) and $0.04$m gain in Median Error (Cross Area), this comes at a severe computational penalty: the model is $\mathbf{7.4x}$ larger and incurs a $\mathbf{46.7}\%$ reduction in real-time speed. Our GeoFlow is designed to be backbone agnostic and demonstrates clear generalizability, working seamlessly with larger architectures like EfficientNet-B5. However, as mentioned before, this flexibility comes with a significant computational overhead ($\mathbf{7.4x}$ more parameters and $\mathbf{46.7}\%$ slower inference speed), which is not justified by the resulting marginal accuracy gains. This is especially critical for real-world deployment scenarios, as devices like small drones, mobile robots, and embedded systems possess severely limited memory and computational power, meaning they simply cannot run large-scale models like EfficientNet-B5. **This decisively validates our initial design choice: the lightweight EfficientNet-B0 provides the optimal balance between necessary accuracy and SOTA efficiency, ensuring the model remains viable for real-time deployment.**

## D. Efficiency Analysis and Deployment Considerations

This section provides additional analysis of the computational efficiency of GeoFlow and its implications for deployment in real-world localization systems. In practical navigation scenarios, inference speed and memory footprint are critical constraints, particularly for embedded platforms such as drones, mobile robots, and edge vision devices. While improving fine-grained cross-view geolocalization accuracy is important, many real-world applications require models that can operate reliably under strict computational and memory budgets.

### D.1. Architectural Efficiency

GeoFlow is designed with a strong emphasis on computational efficiency. Compared with existing FG-CVG methods, our model maintains a significantly smaller computational footprint in terms of both FLOPs and memory consumption. This lightweight design enables deployment on resource-constrained hardware while still maintaining competitive localization accuracy.

### D.2. Efficient Iterative Refinement

The efficiency of the proposed Iterative Refinement Sampling (IRS) algorithm arises from a decoupled computation strategy. The most computationally expensive operations (feature extraction using the EfficientNet backbone and cross-attention-based visual fusion) are performed only once per image pair to obtain the shared visual context representation $\mathbf{f}_{vis}$.

Subsequent refinement iterations operate exclusively on lightweight modules:

- Coordinate encoding layers
- Small regression MLPs used for flow prediction

Since these components represent only a small fraction of the total model computation, multiple refinement iterations can be performed with minimal additional cost. This design allows GeoFlow to benefit from iterative hypothesis refinement without significantly increasing inference latency.

### D.3. Unified Efficiency Benchmark

Table 3 compares GeoFlow with representative FG-CVG approaches in terms of computational cost, memory footprint, and inference latency.

GeoFlow achieves the lowest computational cost among the compared approaches, reducing FLOPs by 34% and memory consumption by 64% relative to HC-Net while also achieving lower inference latency. We achieved this stability while still outperforming CCVPE in mean localization accuracy (8.42m vs 9.16m) and maintaining parity with HC-Net. We simply trade peak static precision in rare cases for the operational reliability required to keep the platform airborne.

### D.4. Deployment Implications

The reduced computational and memory requirements make GeoFlow particularly suitable for edge deployment

Table 2. Ablation Study: Accuracy vs. Computational Overhead. We report Mean and Median localization error (m) for both Same and Cross-Area splits on KITTI.

| | Efficiency | | | Same Area (m) | | Cross Area (m) | |
|---|---|---|---|---|---|---|---|
| Backbone | ↓Params (M) | ↓Mem (MiB) | ↑Speed (FPS) | ↓Mean | ↓Median | ↓Mean | ↓Median |
| EffNet-B0 (Default) | **7.38** | **686** | **29.49** | 0.98 | 0.68 | 8.42 | 5.60 |
| EffNet-B5 (Large) | 54.81 | 1169 | 15.70 | **0.86** | **0.60** | **7.93** | **5.56** |

Table 3. Unified efficiency comparison across representative FG-CVG methods.

| Model | GFLOPs | VRAM | Inference Time | FPS |
|---|---|---|---|---|
| CCVPE | 31.18 | 4730 MiB | 41.7 ms | 24.0 |
| HC-Net | 11.56 | 1900 MiB | 40.0 ms | 25.0 |
| **GeoFlow** | **7.65** | **686 MiB** | **26.0 ms** | **29.5** |
| *Gain (vs HC-Net)* | **-34%** | **-64%** | **-35%** | **+18%** |

scenarios. On embedded platforms such as micro-UAVs or mobile robots, available memory must be shared among multiple perception modules (e.g., obstacle detection, planning, and control). Models with large memory footprints can therefore become impractical even if their theoretical accuracy is high.

By requiring only 686 MiB of VRAM, GeoFlow leaves sufficient memory headroom for other concurrent perception tasks while maintaining video-rate inference (26 ms per frame). Lower computational cost also reduces power consumption and thermal load, which are critical considerations for battery-powered platforms.

Furthermore, lower inference latency directly improves closed-loop navigation stability. For example, for a platform moving at 15 m/s, reducing latency from 60 ms to 26 ms decreases the distance traveled between localization updates by nearly 60%, improving the responsiveness of downstream navigation systems.

## E. Additional Uncertainty Results

Figure 1 shows additional results of the uncertainty produced by our GeoFlow and how our IRS refines the uncertainty during inference, enhancing accuracy and confidence. These visualizations further support our contribution to the IRS algorithm. **The convergence of hypotheses from a broad initial distribution to a sharp, final cluster visually proves that IRS is an effective mechanism for building predictive confidence and enhancing overall localization robustness.**

## F. Additional Qualitative Results

This section provides a deeper qualitative analysis of GeoFlow's performance on the VIGOR dataset through two distinct visual scenarios. Figure 2 visualizes the refinement trajectories for the Same-Area setting, demonstrating precise localization within known operational environments. Figure 3, in contrast, showcases the robustness of the learned regression field in the highly challenging Cross-Area setting, highlighting the model's generalization capabilities to unseen areas. **The combined visual evidence across both known and unseen environments supports the effectiveness of the learned regression field, demonstrating the framework's reliable precision and generalization capabilities.**

## G. Failure Cases Analysis

While GeoFlow generally demonstrates robust performance, understanding the characteristics of its error distribution is essential. **This section highlights the specific failure modes responsible for the outliers in our error distribution.** As illustrated in Figure 4, the majority of these errors are isolated to environments dominated by dense and visually ambiguous vegetation, particularly in the VIGOR Cross-Area setting. The homogeneous textures in the satellite image prevent GeoFlow from establishing the necessary feature variance to guide the hypotheses correctly, leading to convergence at an incorrect, distant location. These catastrophic, isolated failures contribute to the error distribution (i.e., the shift between the Mean and Median error).

## H. Societal Impact and Limitations

**Societal Impact.** The proposed GeoFlow advances the field of autonomous navigation by providing a robust, low-latency alternative to GPS. **By enabling accurate localization at real-time speeds with minimal computational overhead, our framework is particularly valuable for resource-constrained platforms, such as small delivery drones or mobile search-and-rescue robots operating in GPS-noisy areas, such as metropolitan areas, GPS-denied environments, and mountainous areas.** Furthermore, the robustness gained from iteratively refining diverse poses allows downstream decision-making systems to receive a reliable final location estimate, mitigating risks in critical scenarios like disaster response or urban navigation.

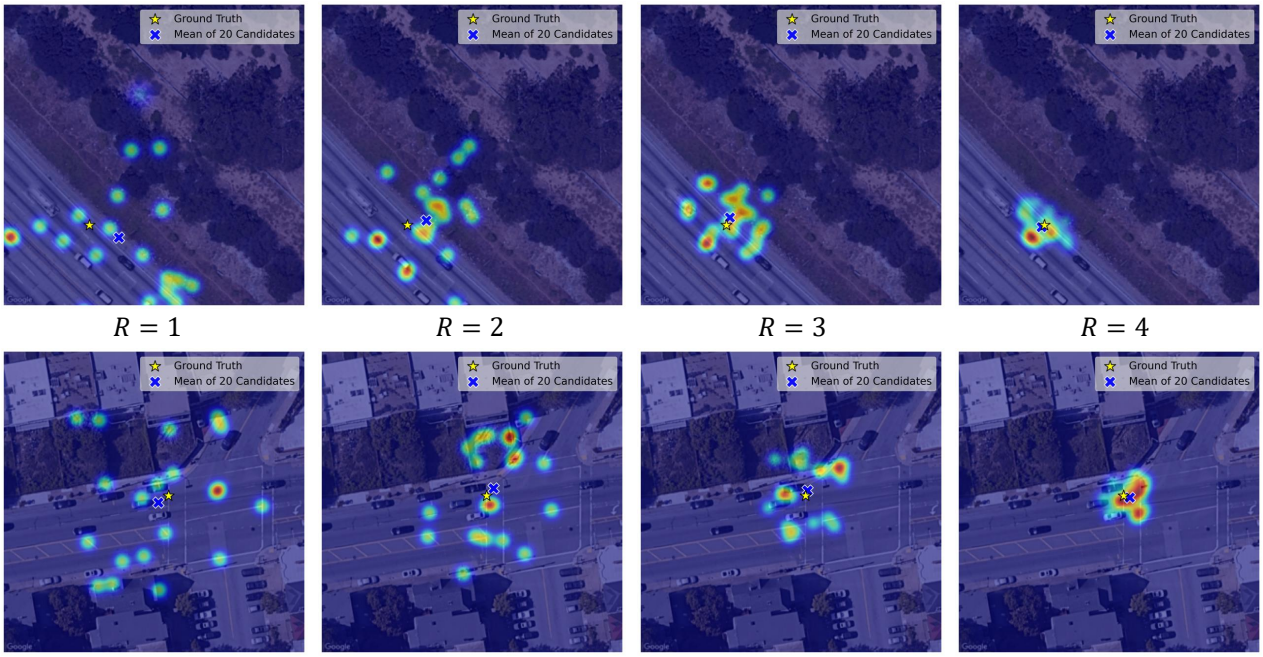


Figure 1. Additional results for IRS refinement dynamics: Visualization of pose hypotheses distribution over refinement rounds $R \in \{1, 2, 3, 4\}$, demonstrating convergence towards the ground truth and how hypothesis consensus reflects confidence.

**Limitations.** Our method achieves state-of-the-art efficiency, but practical deployment for the FG-CVG task still faces challenges inherent to the domain. Performance is sensitive to factors such as the difficulty in accurately predicting location with limited field-of-view (LFOV) ground images, referring to those with a severely restricted viewing angle (unlike the relatively wide LFOV in the KITTI benchmark), where local visual context is severely restricted.

## References

[1] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 1

[2] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 1

[3] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019. 1, 3

[4] Zimin Xia and Alexandre Alahi. Fgˆ 2: Fine-grained cross-view localization by fine-grained feature matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6362–6372, 2025. 1, 2

[5] Zimin Xia, Olaf Booij, and Julian F. P. Kooij. Convolutional cross-view pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3813–3831, 2024. 1

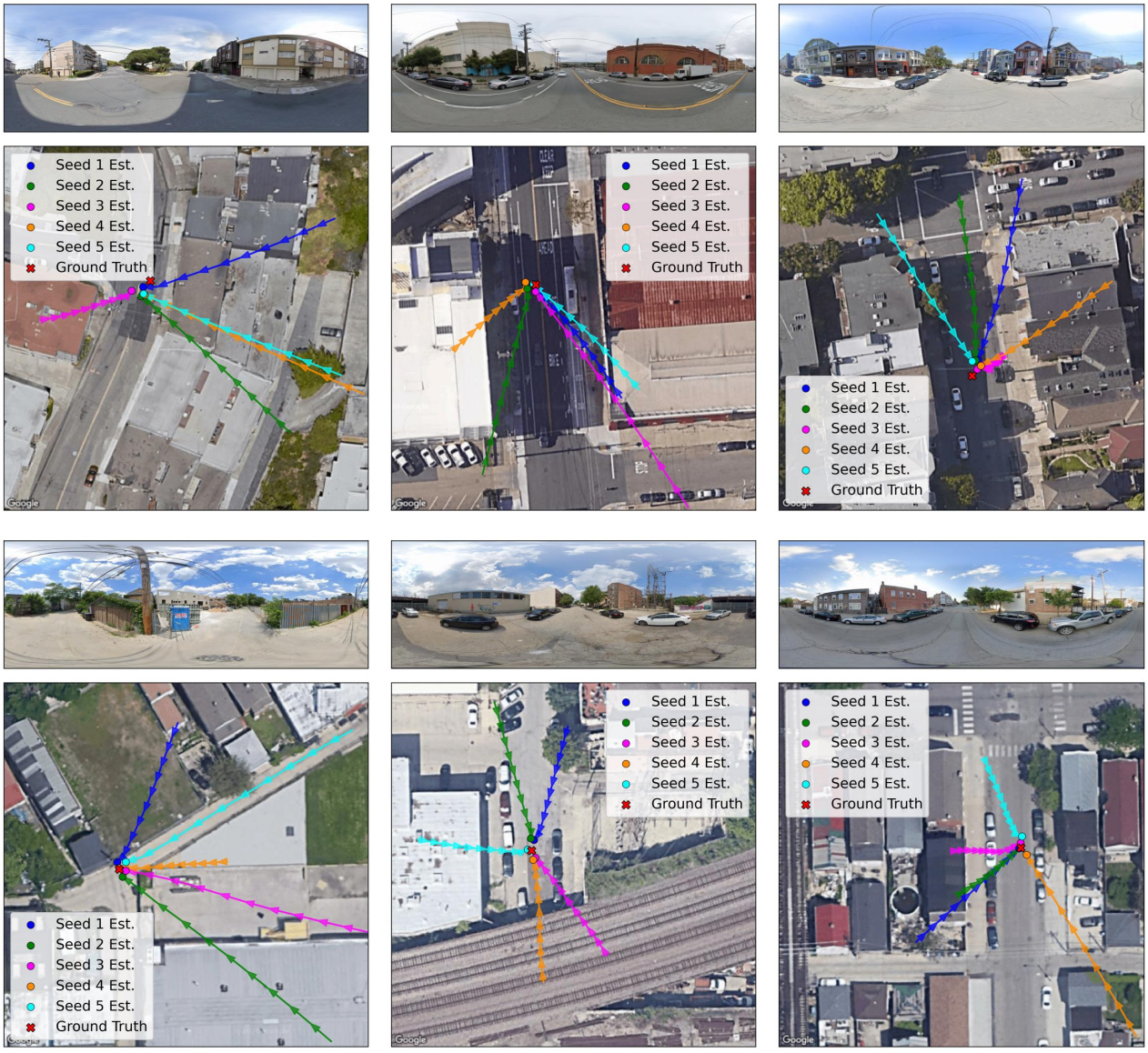


Figure 2. Additional Qualitative localization examples by GeoFlow on the Same-Area VIGOR dataset. The figure visualizes the trajectories generated by the learned regression field by GeoFlow: trajectories show how five randomly sampled initial pose hypotheses (seeds) are transformed to their respective predicted locations (colored circular markers), relative to the ground truth (red X).

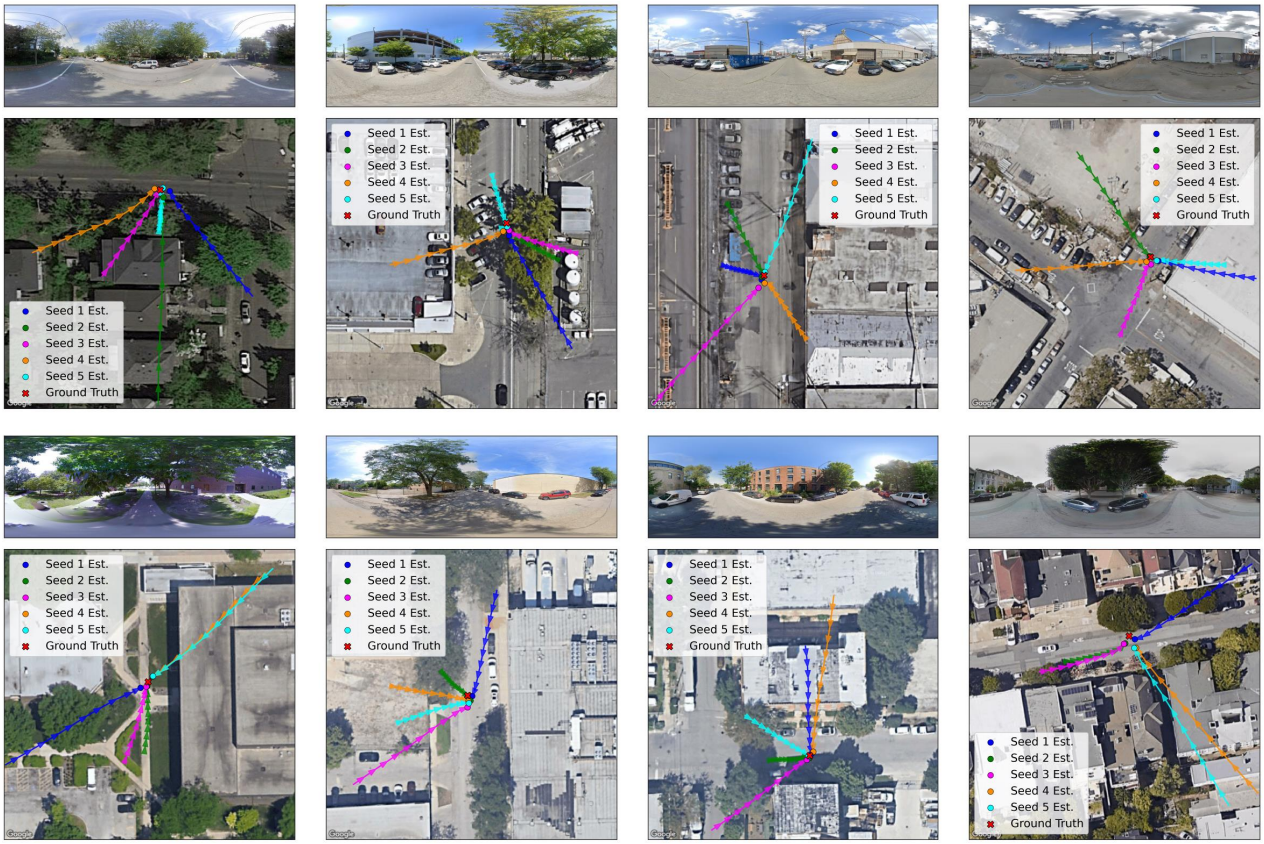


Figure 3. Additional Qualitative localization examples by GeoFlow on the Cross-Area VIGOR dataset. The figure visualizes the trajectories generated by the learned regression field by GeoFlow: trajectories show how five randomly sampled initial pose hypotheses (seeds) are transformed to their respective predicted locations (colored circular markers), relative to the ground truth (red X).

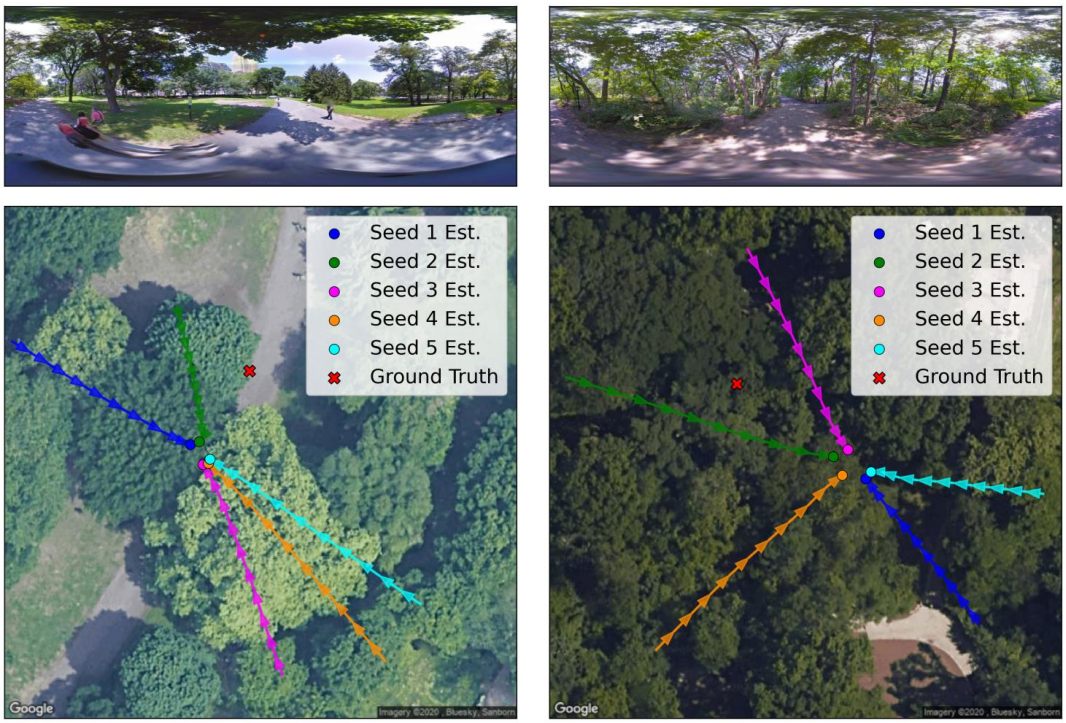


Figure 4. Failure Cases: Localization difficulties in densely vegetated forest environments. The top row shows the ground images, and the bottom row visualizes the refinement trajectories and final converged hypotheses on the corresponding satellite images.