# NARUTO: Neural Active Reconstruction from Uncertain Target Observations

Supplementary Material

## 6. Overview

In this supplementary material, we provide a detailed outline structured as follows: Sec. 7 delves into additional implementation specifics of NARUTO. Sec. 8 examines the computation costs associated with each module. Complementing the results in Sec. 4, Sec. 9 extends our analysis with per-scene evaluations for MP3D and Replica.

## 7. Implementation Details

**Hardware Details** We run the experiments on a desktop PC with a 2.2GHz Intel Xeon E5-2698 CPU and NVIDIA V100 GPU.

**Memory requirement** Memory consumption varies depending on the scene size. As a reference, in a $120m^3$ scene, the corresponding GPU memory and RAM are 8.1GB and 8.6GB respectively. The consumption can be further reduced with a more efficient implementation as our current implementation involves intensive exchanges between RAM and GPU memories.

### 7.1. Neural Mapping Details

We adopt Co-SLAM [73] as the foundational mapping framework for our system, adhering to the hyperparameter configurations established therein. For details pertaining to the hyperparameters specific to the mapping component, we direct readers to [73] for comprehensive information.

### 7.2. Efficient RRT Details

Path planning in three-dimensional spaces presents significant computational challenges, particularly when employing standard 3D RRT algorithms [35]. In our approach, we introduce an accelerated version of RRT, dubbed E-RRT (Efficient RRT), which incorporates several optimizations for improved performance.

The primary innovation in E-RRT, drawing inspiration from RRT-Connect [34], is its strategy to first attempt direct connections from the growing tree to the goal at each iteration. While this does not ensure the shortest path, it significantly enhances the efficiency of finding a viable path.

Furthermore, E-RRT enhances the process of node expansion. Instead of adding a single node, our method integrates a series of feasible points uniformly distributed between a randomly generated node and its nearest neighbor in the tree, based on a predefined step size, for instance, 10 cm, up the distance of $M \times$ step size. Here $M$ equals to 10. This modification substantially accelerates the expansion of the tree, especially in the initial growth stages.

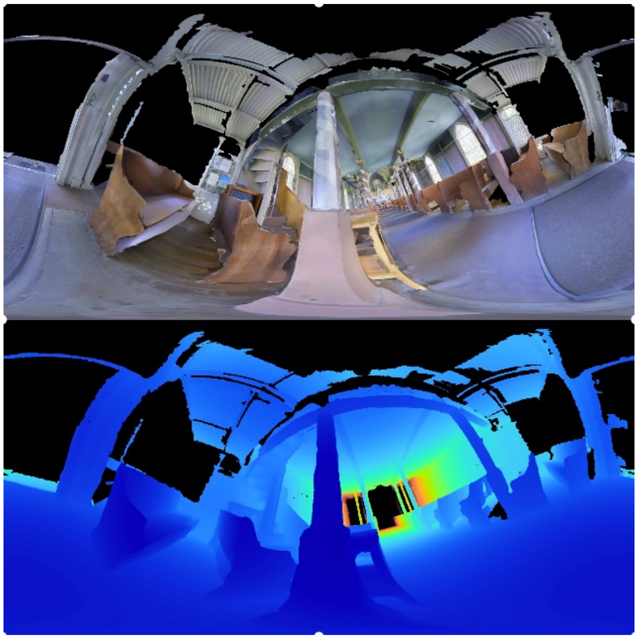

Figure 7. **Equirectangular RGB-D Example** Black regions refer to the invalid regions with zero depth measurement. The ratio of black regions increases significantly when the agent leaves the building. This is used as a signal for collision detection.

Lastly, we address the increasing computational load associated with nearest-neighbor searches as the tree expands. By leveraging parallel processing on a GPU, E-RRT achieves a consistently high search speed, thus mitigating the computational costs that typically escalate with tree complexity.

### 7.3. Collision Detection

We have tailored two distinct collision detection methodologies to align with the nuances of the Replica and Matterport3D datasets.

For experiments conducted within the Replica dataset, collision detection is facilitated through an SDF map derived from our hybrid scene representation. We assess potential collisions by sampling points at 2 cm intervals between consecutive states and querying the SDF map at these points. A collision is inferred when the SDF value at any point falls below the 5 cm threshold, consistent with our model of the agent as a sphere with a 5 cm radius.

This protocol effectively prevents the agent from intersecting with wall surfaces during simulations. Nonetheless, it cannot preclude the agent from exiting the scene through non-watertight boundaries. In contrast, the Matterport3D dataset, reflecting real-world environments, presents unique challenges with regions devoid of geometry—artifacts of incomplete depth data during dataset construction. These gaps in the environment can erroneously permit the agent to traverse through “walls” or exit buildings. To counteract this, in addition to the SDF-based collision detection, we have developed a specialized collision detection system that assesses equirectangular depth measurements (*e.g.* Fig. 7) at prospective states, calculating the proportion of invalid regions. An increase in this proportion signals potential egress from the building, and by establishing a threshold ratio, we can determine the validity of the next state, thereby preventing unintended departure from the environment.

### 7.4. Rotation Planning

As delineated in Sec. 3.3, when the agent arrives at a designated goal state $\mathbf{s}_g$, it proceeds to sequentially observe the top-10 points of uncertainty within its sensing radius through a series of rotational movements. In an effort to reduce the number of steps necessary to cover all ten of these uncertain perspectives, we have devised a straightforward rotational planning algorithm. This method involves identifying the subsequent viewpoint that can be reached with the least rotational effort and then executing the transition using a Spherical Linear Interpolation (SLERP) strategy.

### 7.5. Active Ray Sampling Details

In the context of mapping optimization within Co-SLAM[73], the conventional approach entails the random selection of 2048 pixels from the database, supplemented by a minimum of 100 pixels from the current viewpoint. Our Active Ray Sampling strategy introduces a refinement to this process. Specifically, we quadruple the count of randomly sampled pixels, thus drawing 8192 pixels from the database and ensuring at least 400 pixels from the current viewpoint. Within this augmented sample set, we then identify and prioritize the 500 most uncertain pixels. The remaining 1548 pixels are selected from the database, in addition to a minimum of 100 random points from the current viewpoint. This hybrid sampling method effectively combines the breadth of random sampling with the targeted insight of Active Ray Sampling, thereby capturing a broad yet informative snapshot of the environment.

## 8. Runtime Analysis

### 8.1. System Runtime

In this section, we present a detailed runtime analysis of the three major modules in NARUTO, as illustrated in Fig. 8. The first module is a simulator for data generation. The second is a mapping module optimized for a hybrid scene representation. Lastly, we have an uncertainty-aware planning module. For data generation, HabitatSim requires, on average, 24.4ms to generate 680 × 1200 RGB-D data per iteration. The Mapping module, although taking about 300ms per iteration, averages 60.5ms since it is activated only every five keyframes. The Active Planning module averages 2.1ms, which includes 0.3ms for collision detection per iteration. Additionally, Active Planning encompasses two modules that are triggered occasionally when the ’PLAN_REQUIRED’ condition is met. These are the uncertainty-aware goal searching, averaging 6.8ms, and RRT path planning, averaging 5.77 ms. In conclusion, our analysis demonstrates that NARUTO offers real-time capabilities, particularly due to its efficient planning module.

| Method | Time (ms) | Node Num. | Step Num. |
|---|---|---|---|
| RRT | $19 \times 10^3$ | $19 \times 10^3$ | $28 \times 10^3$ |
| w/o direct line | $17 \times 10^3$ | $20 \times 10^3$ | $21 \times 10^3$ |
| w/o fast tree | 16.00 | 44.17 | 2.56 |
| Ours (E-RRT) | 5.77 | 16.70 | 1.19 |

Table 3. **RRT runtime analysis on Replica-room0.** We conducted a runtime analysis of RRT variants, revealing that our optimized RRT implementation significantly outpaces traditional RRT in planning speed, achieving real-time planning capabilities.

### 8.2. RRT Runtime Analysis

In this section, we delve deeper into our optimized RRT implementation, as outlined in Sec. 7.2. We have engineered a customized version of RRT that enhances planning speed through several strategies:

- Direct Line: Actively identifying straight paths that link the RRT tree to the goal.
- Fast Tree: Speeding up the expansion of the tree.
- Parallel Computing: Utilizing GPU processing for increased efficiency.

These innovations significantly reduce the time required for path planning, making our RRT variant highly suitable for real-time applications. We present an ablation study on the runtime performance of our RRT approach in Tab. 3. To maintain consistency, all experiments were conducted using parallel processing for nearest-neighbor searches during tree expansion.

**Evaluation** Our evaluation of the methods encompasses three key metrics: the average time taken for each path planning request, the average number of nodes generated within the RRT tree, and the average number of steps taken in the RRT process.

**Analysis** Compared to traditional RRT, our efficient RRT implementation is markedly faster, both in average planning time and iteration count. It also generates fewer nodes and uses less memory, as shown by the reduced average number of nodes required per planning request. The ablation

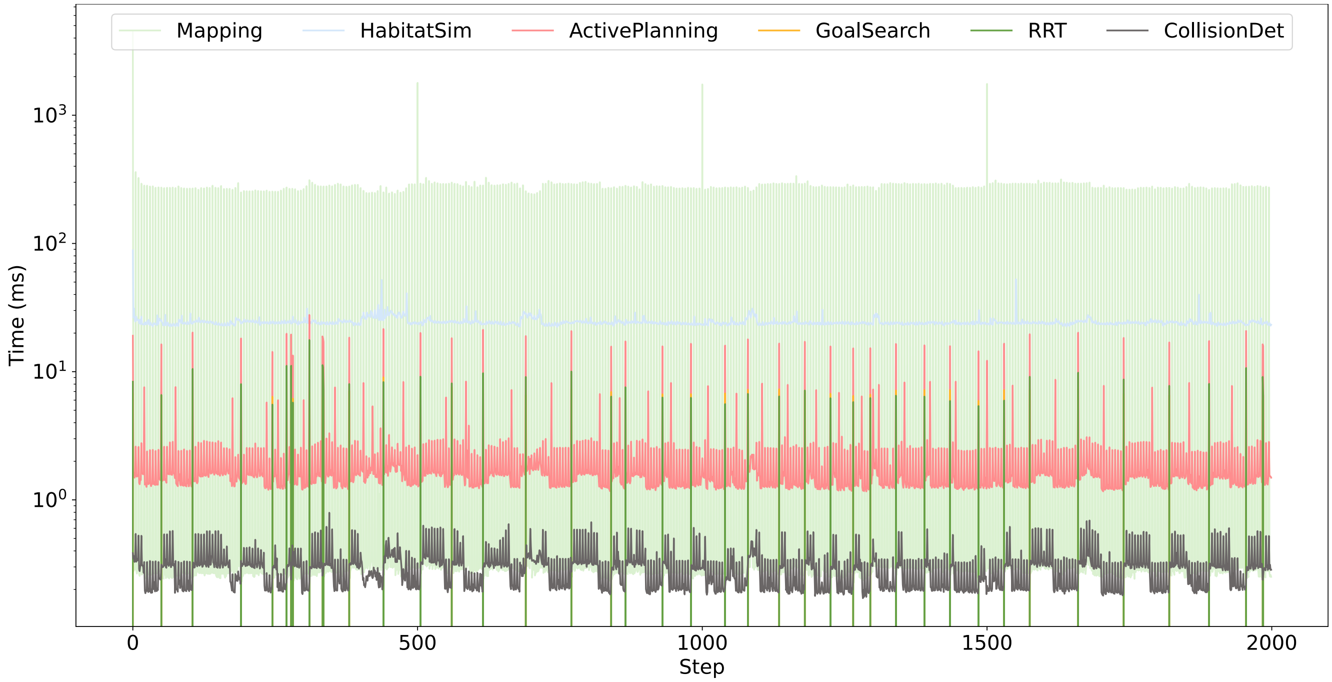

Figure 8. **Runtime Analysis in the Replica-room0 Environment** This figure illustrates the runtime analysis of each module within the *Replica-room0* environment. A notable runtime impulse is observed during goal-searching iterations. The analysis encompasses three principal modules: Habitat Simulator for data generation, Active Planning for path planning, and Mapping for mapping optimization. In the Active Planning module, further runtime analysis includes its submodules: Uncertainty-aware Goal Searching, RRT Path Planning, and Collision Detection.

study detailed in Tab. 3 highlights that our primary strategy for improvement involves identifying potential straight paths, drawing inspiration from RRT-Connect [34]. This approach, along with quicker tree growth, not only accelerates the planning process but also decreases memory usage.

# 9. Additional Experimental Results

## 9.1. Detailed results on MP3D and Replica

In this section, we present more comprehensive results for the various scenes included in the Matterport3D [8] and Replica dataset [67]. Detailed, scene-specific quantitative results are provided in Tab. 5 and Tab. 4. For the qualitative visualization, the reconstructed meshes undergo a culling process as delineated in Neural RGB-D [2] and GoSURF [74], ensuring that only the most relevant data is presented.

**MP3D** In Tab. 5, we present a comparative analysis of our method against the state-of-the-art Active Neural Mapping (ANM) [79]. The results demonstrate that our method outperforms ANM across all evaluated metrics. Most notably, our method exhibits a significant advancement in terms of reconstruction quality and completeness, surpassing the existing benchmarks set by previous art. This consistent superiority in performance underscores the effectiveness of our approach in challenging reconstruction scenarios.

In Fig. 9, we conduct a qualitative evaluation of our 3D reconstruction method against the ground truth for various scenes in the Matterport3D dataset. Ground truth meshes are presented in the odd-numbered rows, while the even-numbered rows showcase our method's reconstructed meshes. Each scene is identified by a unique code (*e.g.*, “Gdvg”, “gZ6f”) on the left. We offer a tripartite comparison for each: the first and second columns depict the exterior surfaces; the third and fourth columns reveal the interior surfaces; and the final two columns provide close-up views of the intricate internal reconstructions. This format delineates a comprehensive visual assessment, contrasting both the textural and geometric dimensions of the meshes.

In Fig. 11 through Fig. 15, we present per-scene trajectory visualizations on the Matterport3D dataset. For enhanced visual clarity, we focus exclusively on illustrating the trajectory formed by keyframe camera poses and the reconstructed texture mesh. To provide a thorough perspective of each scene, we include a bird's eye view alongside two distinct side views. This tri-view presentation facilitates a comprehensive understanding of the spatial dynamics in each scene. It is important to note that the “black regions” visible in the mesh represent areas lacking ground truth data, which were consequently excluded from the

| Method | Metrics | office0 | office1 | office2 | office3 | office4 | room0 | room1 | room2 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| **Neural SLAM** | | | | | | | | | | |
| Co-SLAM [73] | Acc. [cm] ↓ | 1.68 | 1.46 | 2.98 | 3.07 | 2.44 | 2.14 | 2.64 | 2.02 | **2.30** |
| | Comp. [cm] ↓ | 1.68 | 1.82 | 2.70 | 2.83 | 2.64 | 2.25 | 2.84 | 2.02 | **2.35** |
| | Comp. Ratio ↑ | 96.25 | 94.44 | 89.80 | 90.82 | 91.59 | 94.61 | 90.32 | 94.09 | **92.74** |
| [73] **w/ ActRay** | Acc. (cm) ↓ | 1.61 | 1.48 | 2.96 | 3.12 | 2.43 | 2.17 | 2.58 | 2.00 | **2.30** |
| | Comp. (cm) ↓ | 1.61 | 1.85 | 2.67 | 2.96 | 2.67 | 2.26 | 2.78 | 2.03 | **2.35** |
| | Comp. Ratio ↑ | 96.24 | 94.44 | 90.61 | 89.85 | 91.51 | 94.66 | 90.23 | 94.08 | 92.70 |
| **Neural Mapping**: Tracking is disabled. | | | | | | | | | | |
| Co-SLAM [73] | Acc. [cm] ↓ | 1.50 | 1.28 | 2.56 | 2.69 | 2.25 | 2.01 | 1.55 | 1.87 | **1.96** |
| | Comp. [cm] ↓ | 1.48 | 1.61 | 2.17 | 2.52 | 2.47 | 2.13 | 1.71 | 1.88 | 2.00 |
| | Comp. Ratio ↑ | 96.33 | 94.65 | 92.47 | 91.43 | 91.34 | 94.67 | 95.45 | 93.95 | 93.79 |
| [73] **w/ ActRay** | Acc. (cm) ↓ | 1.47 | 1.27 | 2.55 | 2.71 | 2.26 | 2.02 | 1.57 | 1.87 | **1.96** |
| | Comp. (cm) ↓ | 1.47 | 1.59 | 2.13 | 2.55 | 2.49 | 2.07 | 1.71 | 1.85 | **1.98** |
| | Comp. Ratio ↑ | 96.44 | 94.80 | 92.90 | 91.32 | 91.32 | 94.92 | 95.40 | 94.12 | **93.90** |
| **Neural Active Mapping** | | | | | | | | | | |
| w/o ActiveRay | Acc. (cm) ↓ | 1.29 | 1.05 | 2.17 | 2.86 | 1.72 | 1.56 | 1.24 | 1.46 | 1.67 |
| | Comp. (cm) ↓ | 1.40 | 1.50 | 1.66 | 3.14 | 1.76 | 1.67 | 1.45 | 1.47 | 1.76 |
| | Comp. Ratio ↑ | 97.92 | 95.87 | 98.04 | 90.68 | 98.09 | 98.31 | 97.62 | 98.55 | 96.89 |
| Uncertainty Net | Acc. (cm) ↓ | 1.32 | 1.05 | 2.04 | 3.13 | 1.70 | 1.58 | 1.26 | 1.45 | 1.69 |
| | Comp. (cm) ↓ | 2.12 | 2.01 | 2.73 | 2.50 | 2.07 | 1.90 | 1.58 | 1.56 | 2.06 |
| | Comp. Ratio ↑ | 94.21 | 93.22 | 92.62 | 92.12 | 94.24 | 96.36 | 96.65 | 97.54 | 94.62 |
| **Full** | Acc. (cm) ↓ | 1.30 | 1.03 | 2.25 | 2.29 | 1.75 | 1.56 | 1.25 | 1.47 | **1.61** |
| | Comp. (cm) ↓ | 1.39 | 1.53 | 1.69 | 2.27 | 1.79 | 1.68 | 1.43 | 1.48 | **1.66** |
| | Comp. Ratio ↑ | 98.17 | 95.26 | 97.54 | 93.91 | 97.93 | 98.28 | 98.04 | 98.47 | **97.20** |

Table 4. Per-scene quantitative results on Replica[67] dataset

| Method | Metric | Gdvg | gZ6f | HxpK | pLe4 | YmJk | **Avg.** |
|---|---|---|---|---|---|---|---|
| ANM [79] | MAD (cm) ↓ | 3.77 | 3.18 | 7.03 | 3.25 | 4.22 | 4.29 |
| | Acc. (cm) ↓ | 5.09 | 4.15 | 15.60 | 5.56 | 8.61 | 7.80 |
| | Comp. (cm) ↓ | 5.69 | 7.43 | 15.96 | 8.03 | 8.46 | 9.11 |
| | Comp. Ratio ↑ | 80.99 | 80.68 | 48.34 | 76.41 | 79.35 | 73.15 |
| Ours | MAD (cm) ↓ | 1.60 | 1.23 | 1.53 | 1.37 | 1.45 | **1.44** |
| | Acc. (cm) ↓ | 3.78 | 3.36 | 9.24 | 5.15 | 10.04 | **6.31** |
| | Comp. (cm) ↓ | 2.91 | 2.31 | 2.67 | 3.24 | 3.86 | **3.00** |
| | Comp. Ratio ↑ | 91.15 | 95.63 | 91.62 | 87.76 | 84.74 | **90.18** |

Table 5. **Per-scene quantitative results on Matterport3D [8] dataset**. Our method achieves consistently better reconstruction than the state-of-the-art method ANM [79].

mapping optimization process. Our observations indicate that while our method demonstrates high completeness in fully exploring the environment, it tends to allocate a considerable number of steps to survey these "black regions". This behavior can be attributed to our selective exclusion of these regions during mapping optimization, which in turn, prevents effective reduction of uncertainty in these areas. Our method, prioritizing observation of uncertain regions, thus allocates more attention to these parts. This phenomenon is a reflection of the challenges posed by the imperfect simulation of real-world environments.

**Replica** We present per-scene ablation studies on Replica in Tab. 4. These results demonstrate that Active Ray Sampling enhances the performance of CoSLAM [73], particularly in scenarios where tracking is disabled. Additionally, our ablation studies reveal that employing the Uncertainty Grid (Full) approach yields superior results compared to the Uncertainty Net across most scenes.

In Fig. 10, we conduct a qualitative evaluation of our 3D reconstruction method against the ground truth for various scenes in the Replica dataset. Ground truth meshes are presented in the odd-numbered rows, while the even-numbered rows showcase our method's reconstructed meshes. Our results show a high level of quality and completeness, closely mirroring the ground truths.

In Fig. 16 - Fig. 23, we present trajectory visualization for each scene. Given that five trials were conducted for each scene, we selectively showcase the most illustrative visualization result for demonstration purposes. In our qualitative analysis, we present two key elements for each scene: the texture mesh visualization and the corresponding planned trajectory. Similarly, we only illustrate the trajectory formed by keyframe camera poses and the reconstructed texture mesh for better clarity.

| Method | Metrics | office0 | office1 | office2 | office3 | office4 | room0 | room1 | room2 |
|---|---|---|---|---|---|---|---|---|---|
| **CoSLAM** | Comp. Ratio ↑ | 96.33 | 94.65 | 92.47 | 91.43 | 91.34 | 94.67 | 95.45 | 93.95 |
| (no tracking) | Traj. (m) ↑ | 18.20 | 11.56 | 23.16 | 29.16 | 25.22 | 24.69 | 16.21 | 23.07 |
| **Ours** | Comp. Ratio ↑ | 98.17 | 95.26 | 97.54 | 93.91 | 97.93 | 98.28 | 98.04 | 98.47 |
|  | Traj. (m) ↑ | 81.27 | 30.02 | 90.20 | 88.59 | 96.36 | 73.91 | 96.99 | 41.31 |

Table 6. Per-scene trajectory length evaluation on Replica[67] dataset

## 9.2. More qualitative comparison on MP3D

For the completeness of the study, we provide more comparison between ground truth, ANM baseline [79], and our method in Matterport3D dataset, as shown in Fig. 24. We trim the meshes for a better visualization purpose.

## 9.3. Comparison against passive mapping methods

In traditional mapping methods, typically involving environments scanned by human-operated sensing devices, the trajectory of scanning significantly impacts the reconstruction's quality and completeness. Such approaches are termed passive mapping methods, characterized by the absence of a planning or guidance module. In Tab. 2, we present a quantitative comparison between Passive Neural Mapping and Active Neural Mapping, utilizing Co-SLAM as the backbone. Here, we aim to offer additional qualitative comparisons in Fig. 25 to highlight differences in reconstruction details more vividly. In passive Co-SLAM (with tracking disabled), regions may be missed or poorly reconstructed if not adequately covered by the scanning trajectory. Conversely, our active reconstruction method ensures a more comprehensive and accurate reconstruction, effectively addressing these limitations.

We compared the trajectory lengths of passive versus active scanning on the Replica dataset, with the results detailed in Tab. 6. Under the same conditions (2000 frames with 400 keyframes), passive scanning may result in redundant observations due to the lack of guided scanning. Active scanning, on the other hand, enables more extensive coverage and yields superior reconstruction quality. However, this approach typically results in longer trajectories, as the agent continuously moves to ensure comprehensive scanning of the environment.

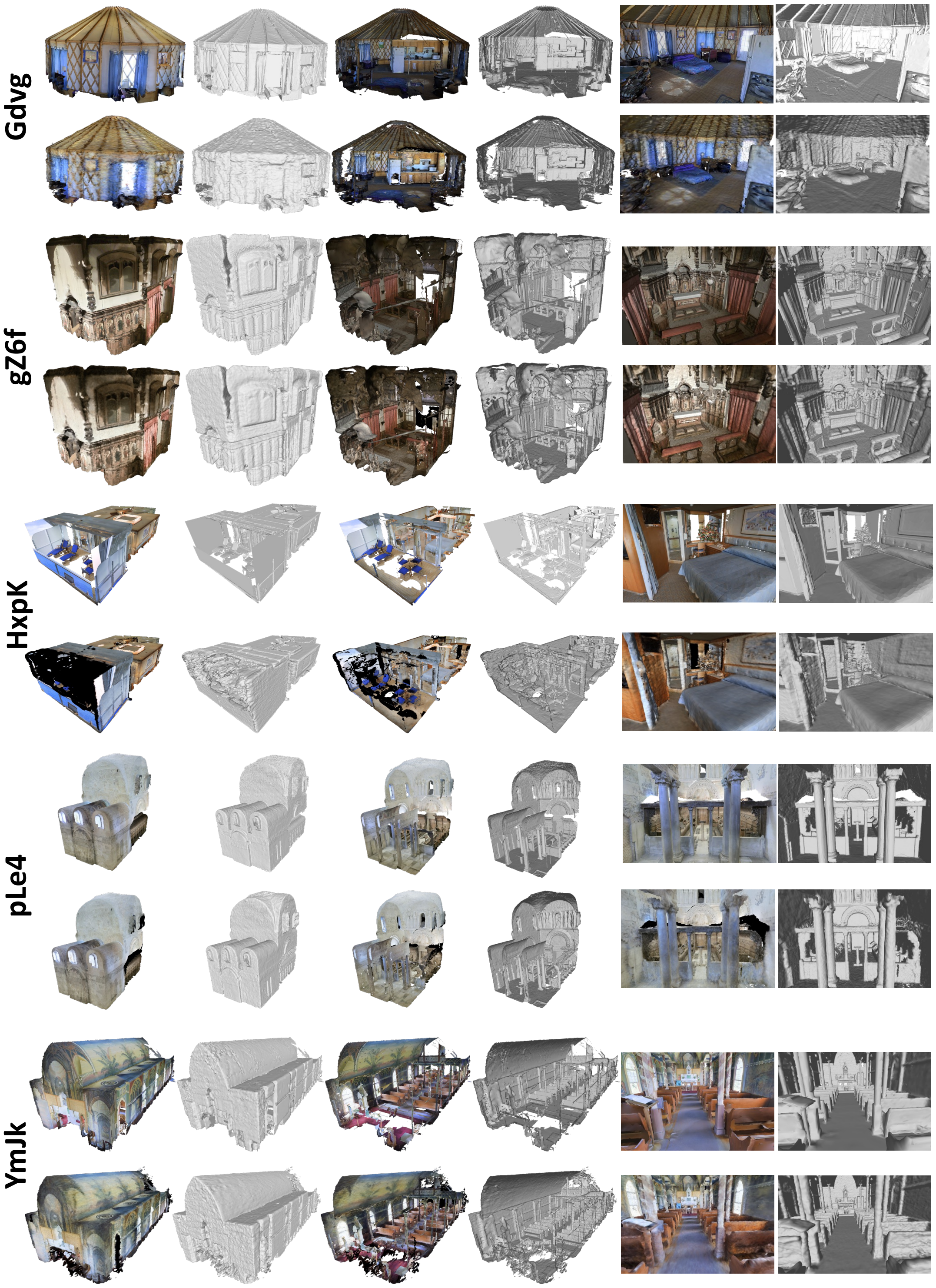

Figure 9. **MP3D Reconstruction Results** This presents a side-by-side comparison of the reconstruction results with the Matterport3D dataset. The odd-numbered rows display the ground truth meshes, while the even-numbered rows feature the meshes reconstructed by our method. Our results show a high level of quality and completeness, closely mirroring the ground truths. This alignment underscores the efficacy of our method in accurately exploring and reconstructing complex spatial geometries.

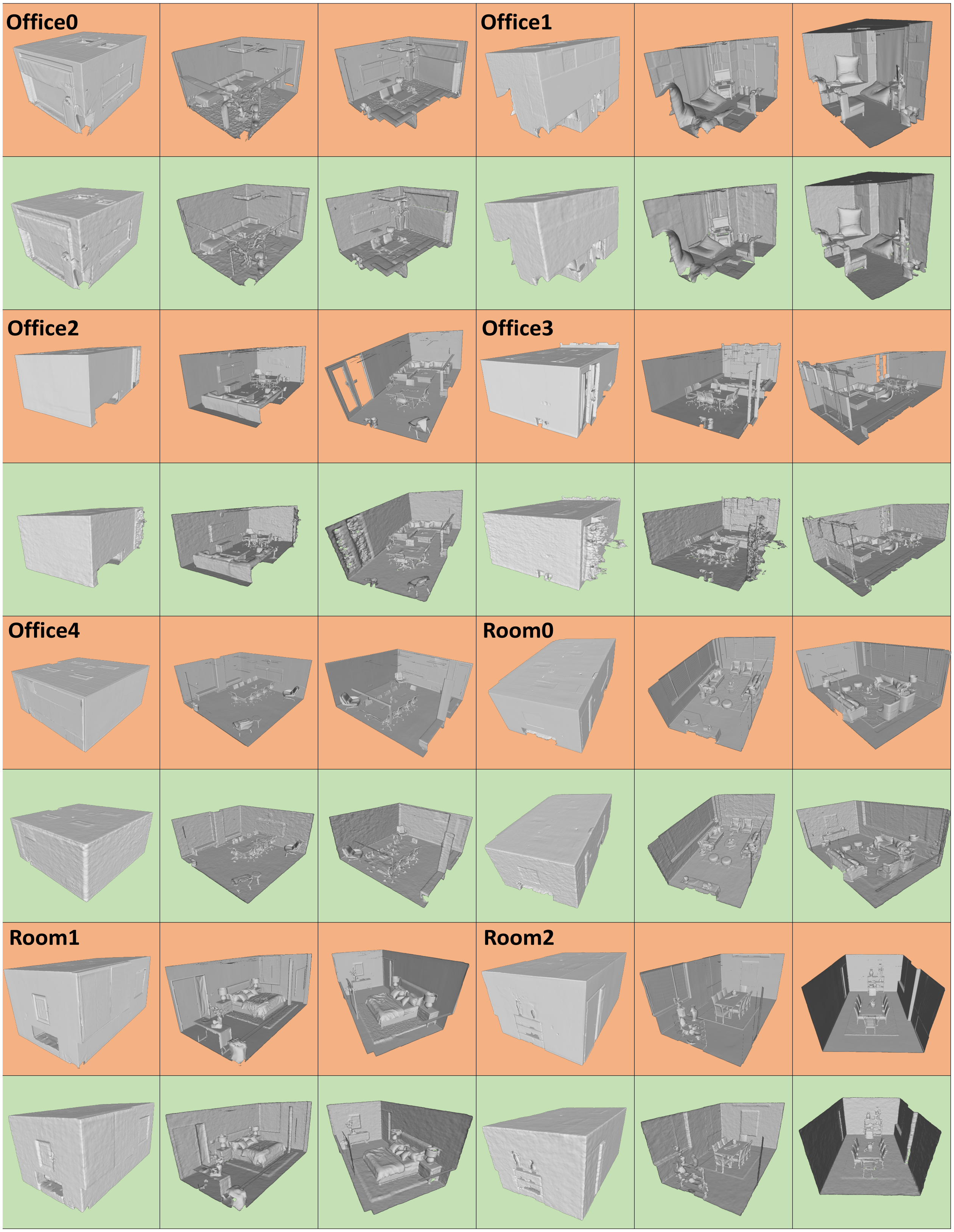

Figure 10. **Replica Reconstruction Results** This presents a side-by-side comparison of the reconstruction results with the Replica dataset. The odd-numbered rows display the ground truth meshes, while the even-numbered rows feature the meshes reconstructed by our method. Our results show a high level of quality and completeness, closely mirroring the ground truths.

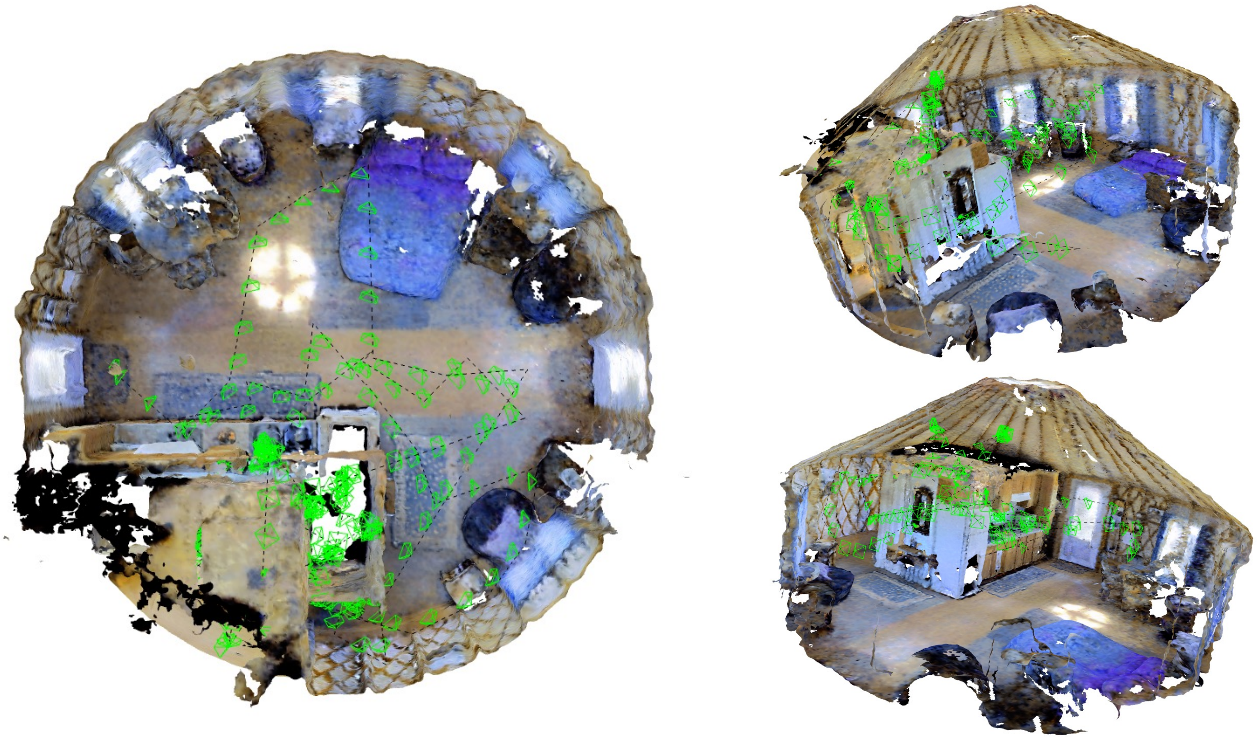

Figure 11. **Matterport3D (Gdvg)** Reconstructed Mesh and planned trajectory.

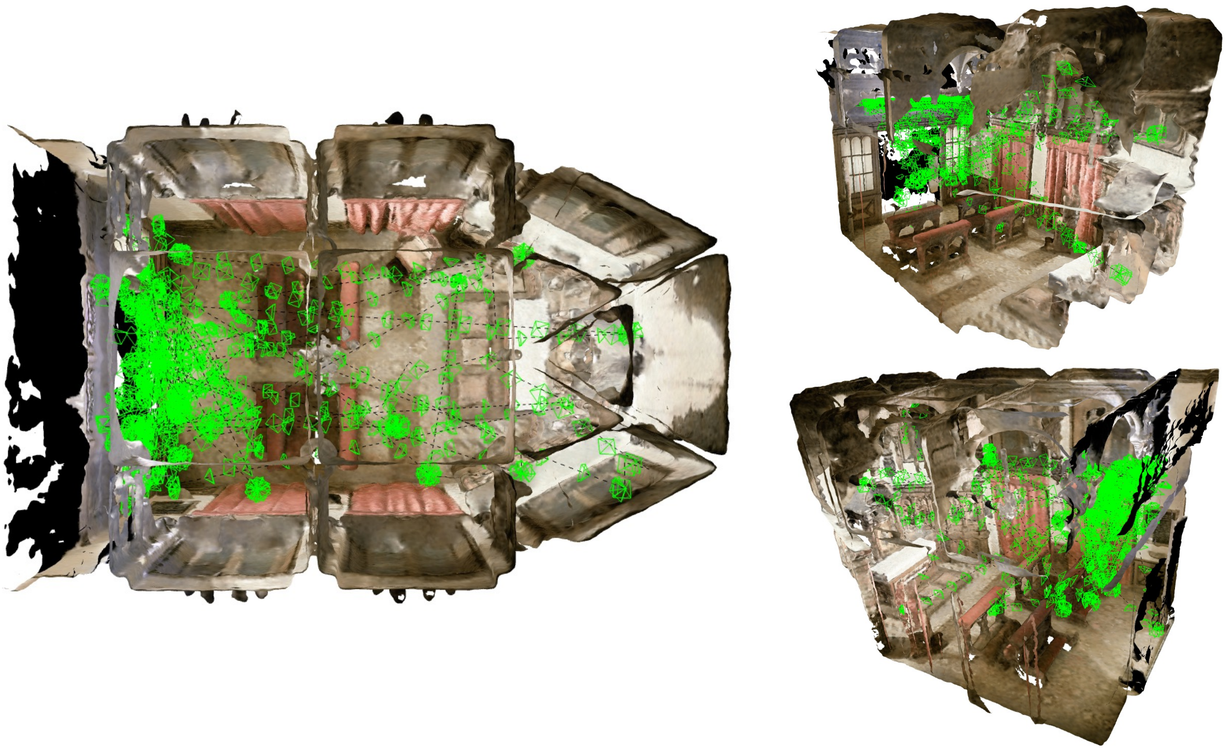

Figure 12. **Matterport3D (gZ6f)** Reconstructed Mesh and planned trajectory.

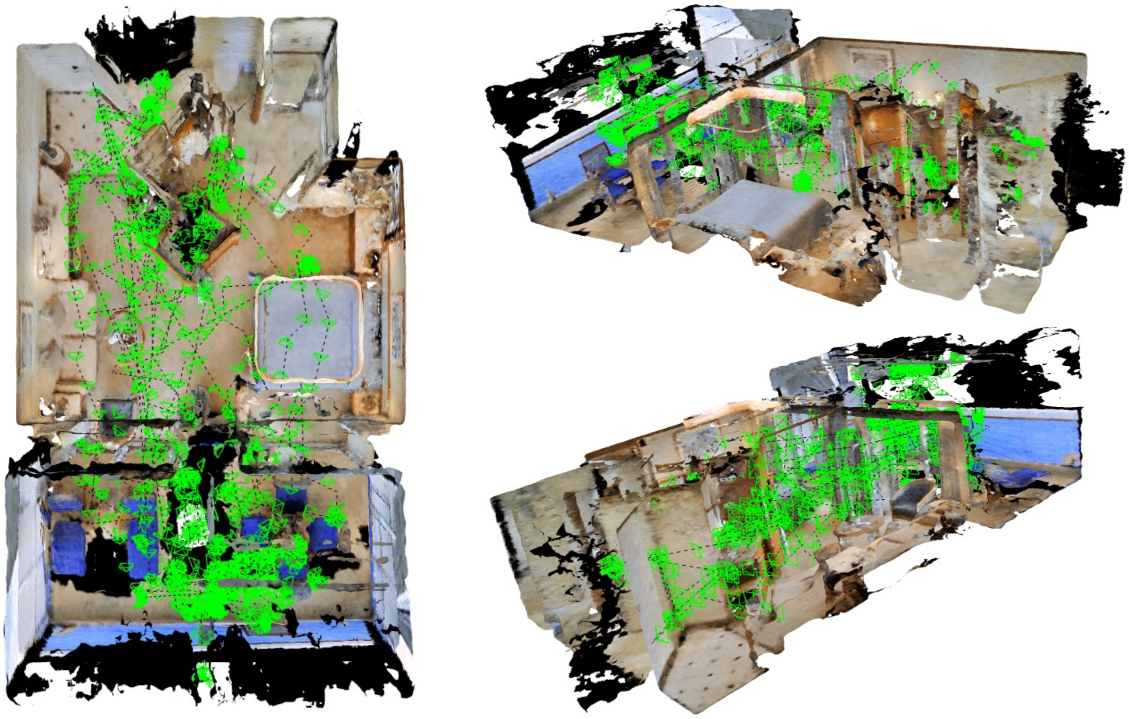

Figure 13. **Matterport3D (HxpK)** Reconstructed Mesh and planned trajectory.

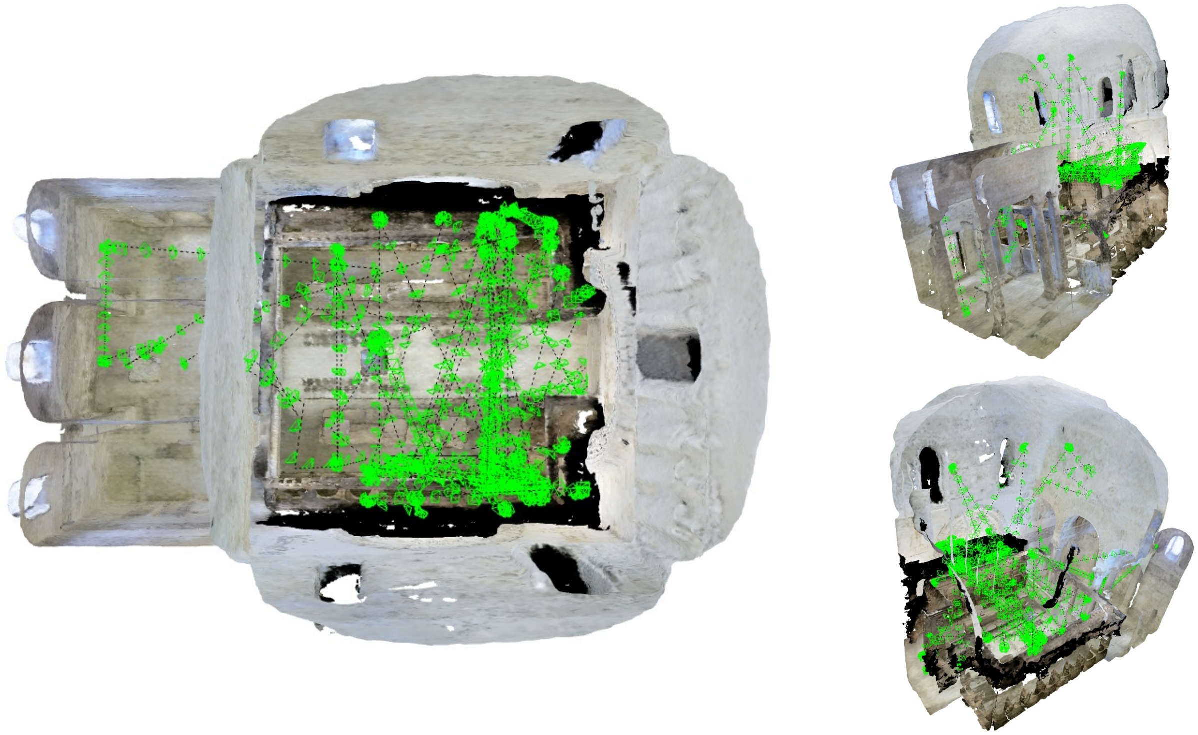

Figure 14. **Matterport3D (pLe4)** Reconstructed Mesh and planned trajectory.

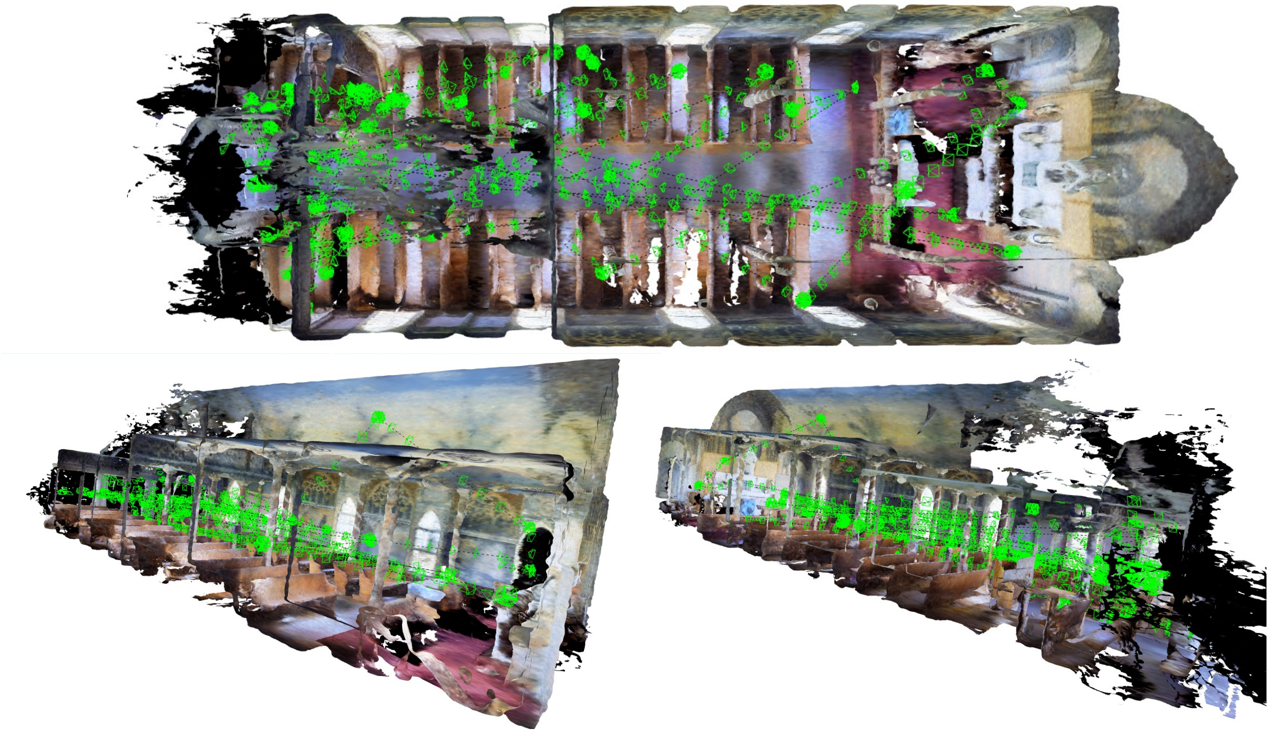

Figure 15. **Matterport3D (YmJk)** Reconstructed Mesh and planned trajectory.

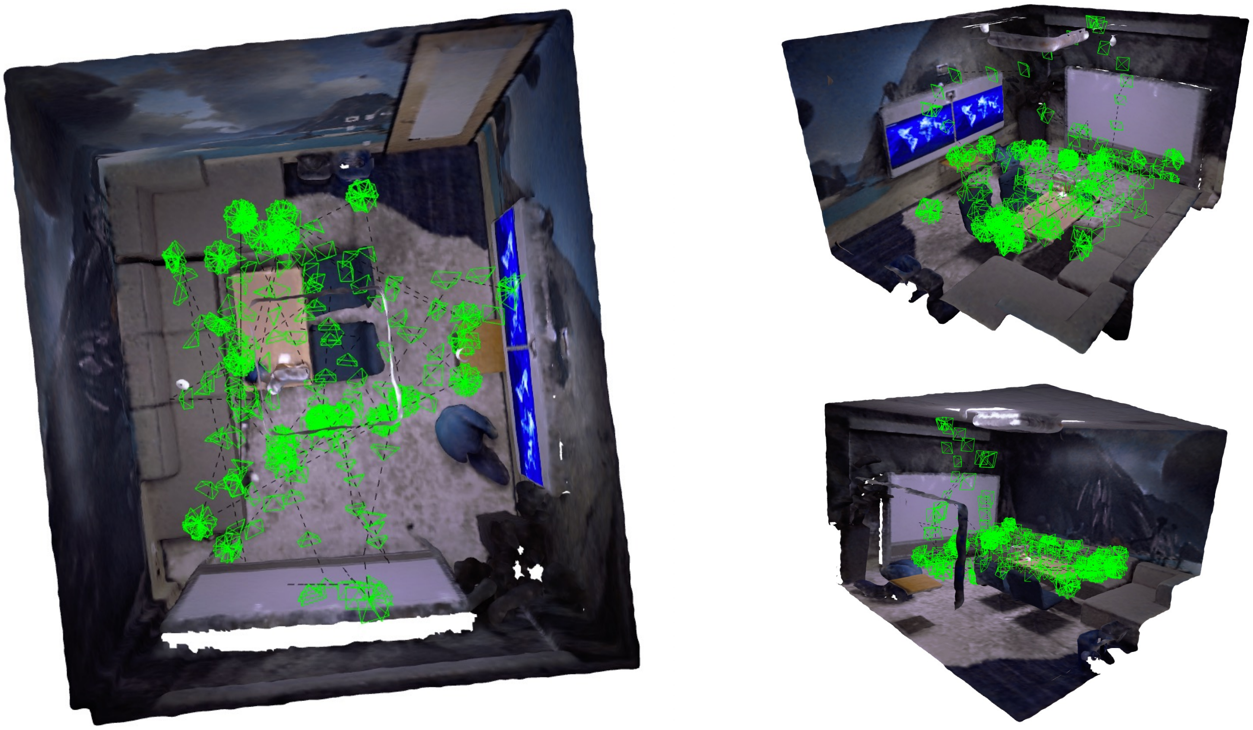

Figure 16. **Replica (office0)** Reconstructed Mesh and planned trajectory.

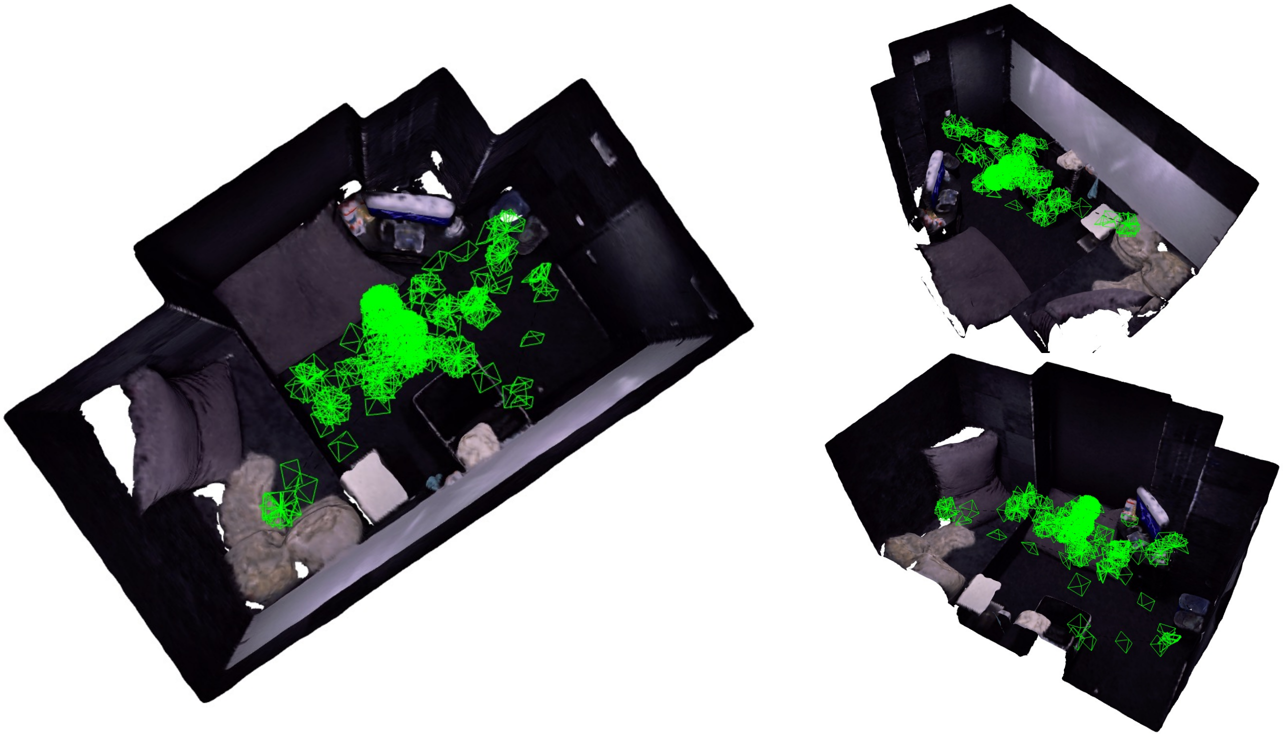

Figure 17. **Replica (office1)** Reconstructed Mesh and planned trajectory.

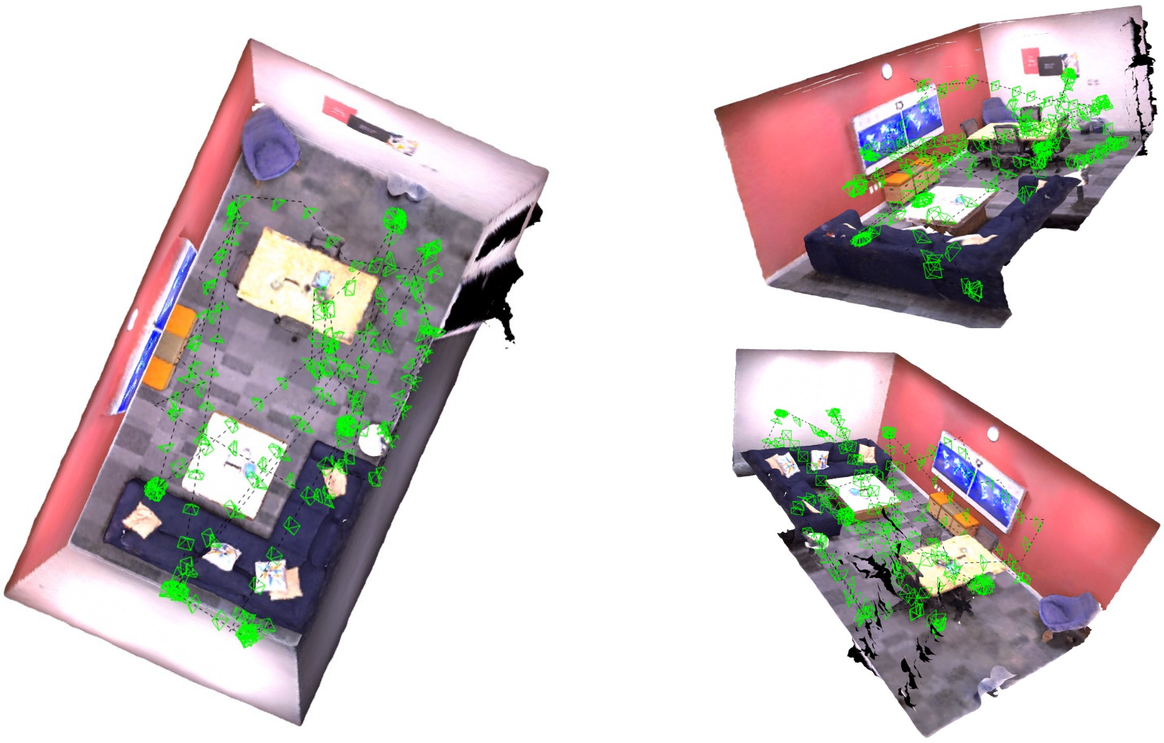

Figure 18. **Replica (office2)** Reconstructed Mesh and planned trajectory.

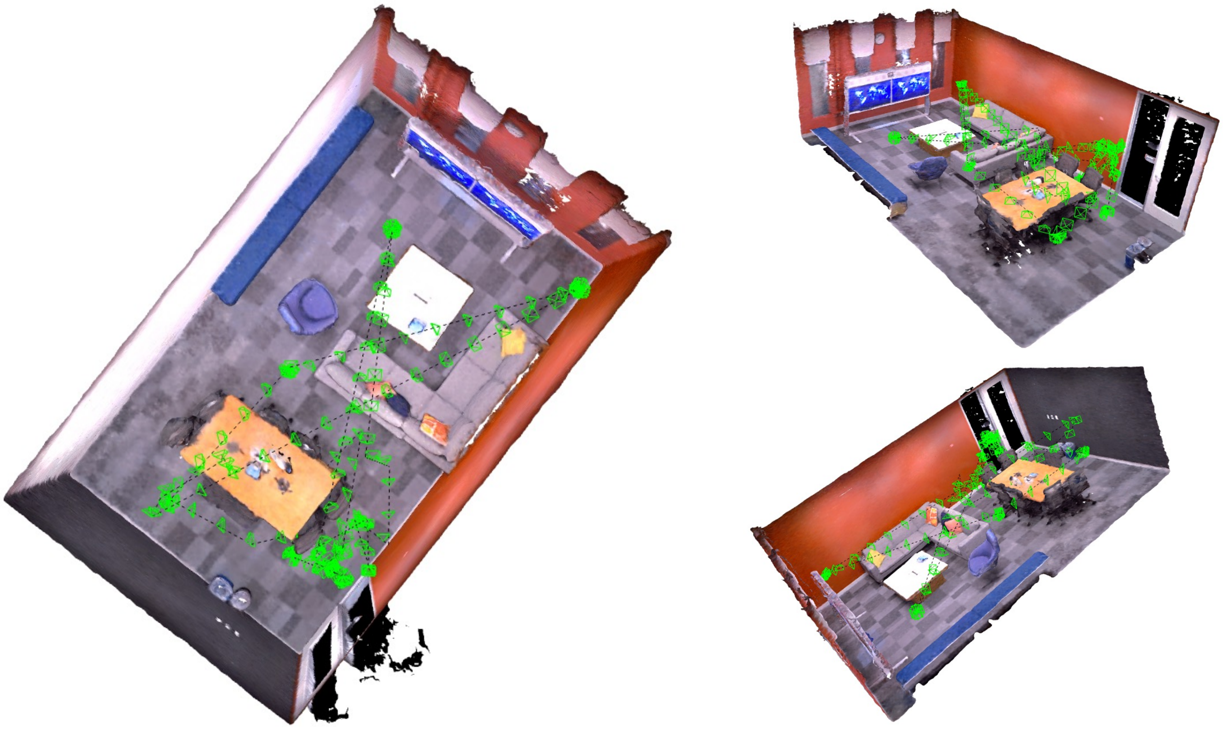

Figure 19. **Replica (office3)** Reconstructed Mesh and planned trajectory.

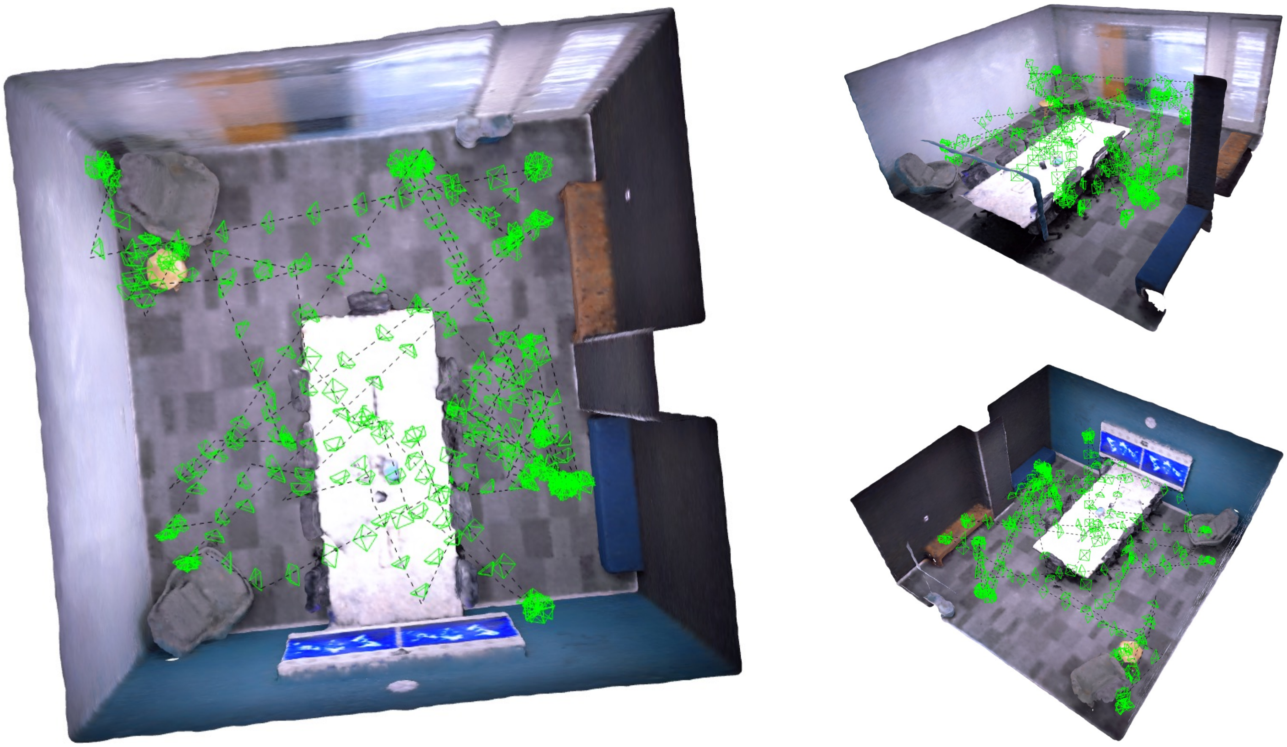

Figure 20. **Replica (office4)** Reconstructed Mesh and planned trajectory.

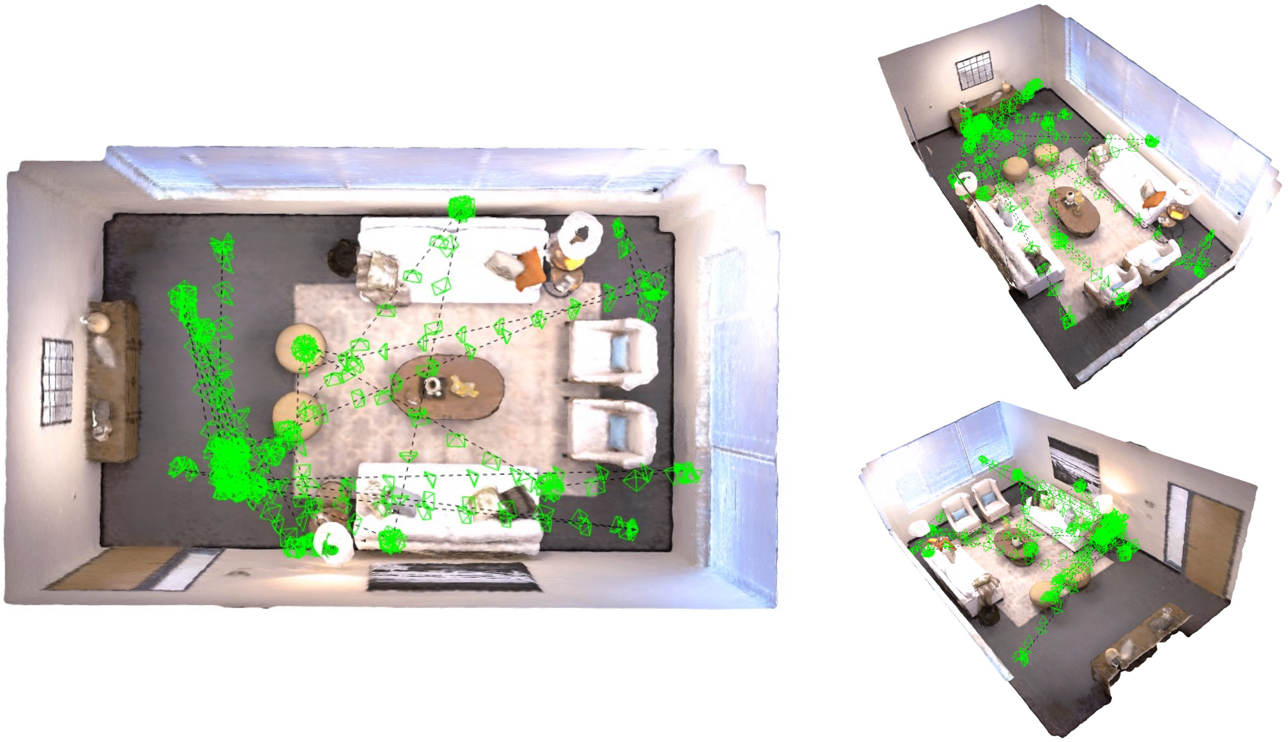

Figure 21. **Replica (room0)** Reconstructed Mesh and planned trajectory.

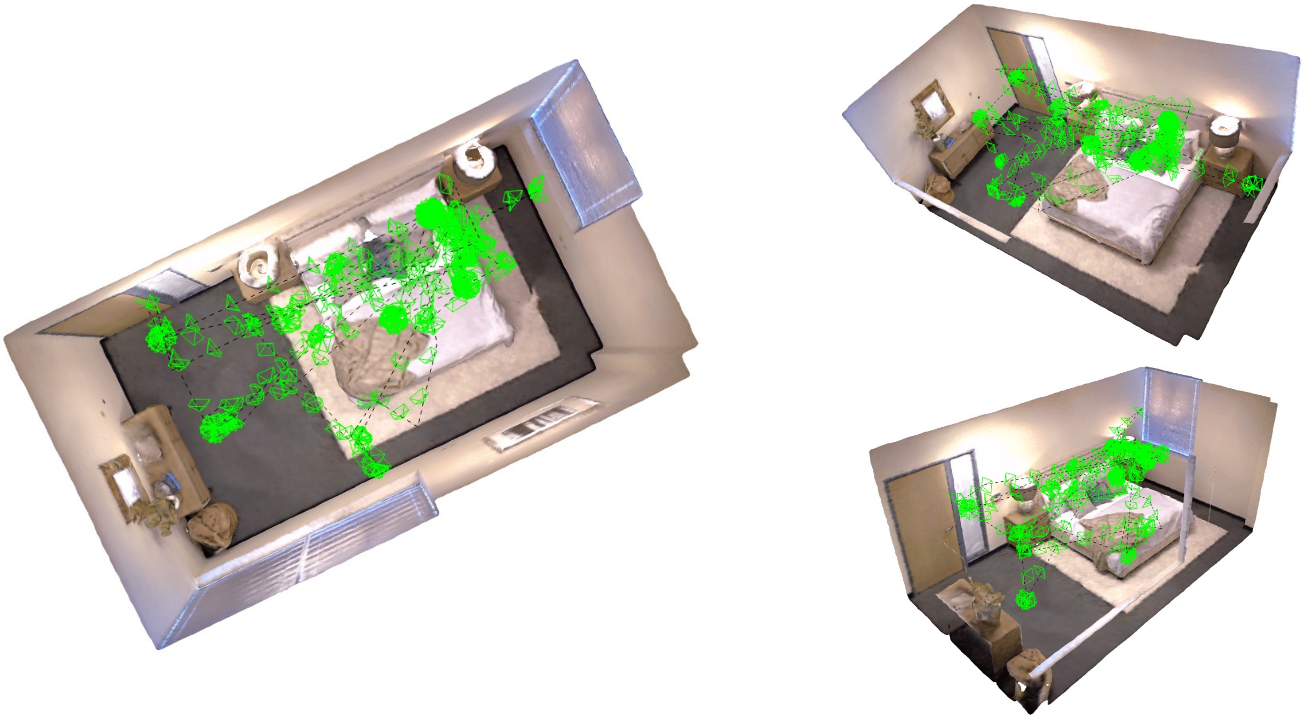

Figure 22. **Replica (room1)** Reconstructed Mesh and planned trajectory.

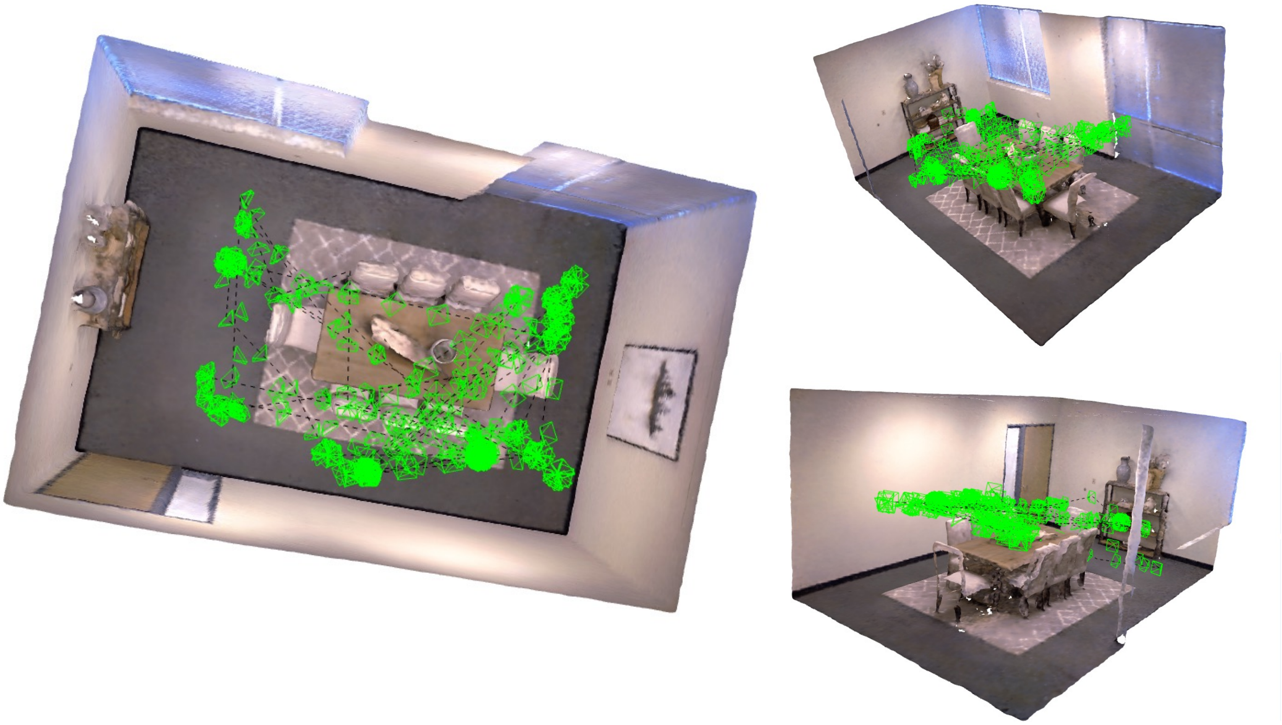

Figure 23. **Replica (room2)** Reconstructed Mesh and planned trajectory.

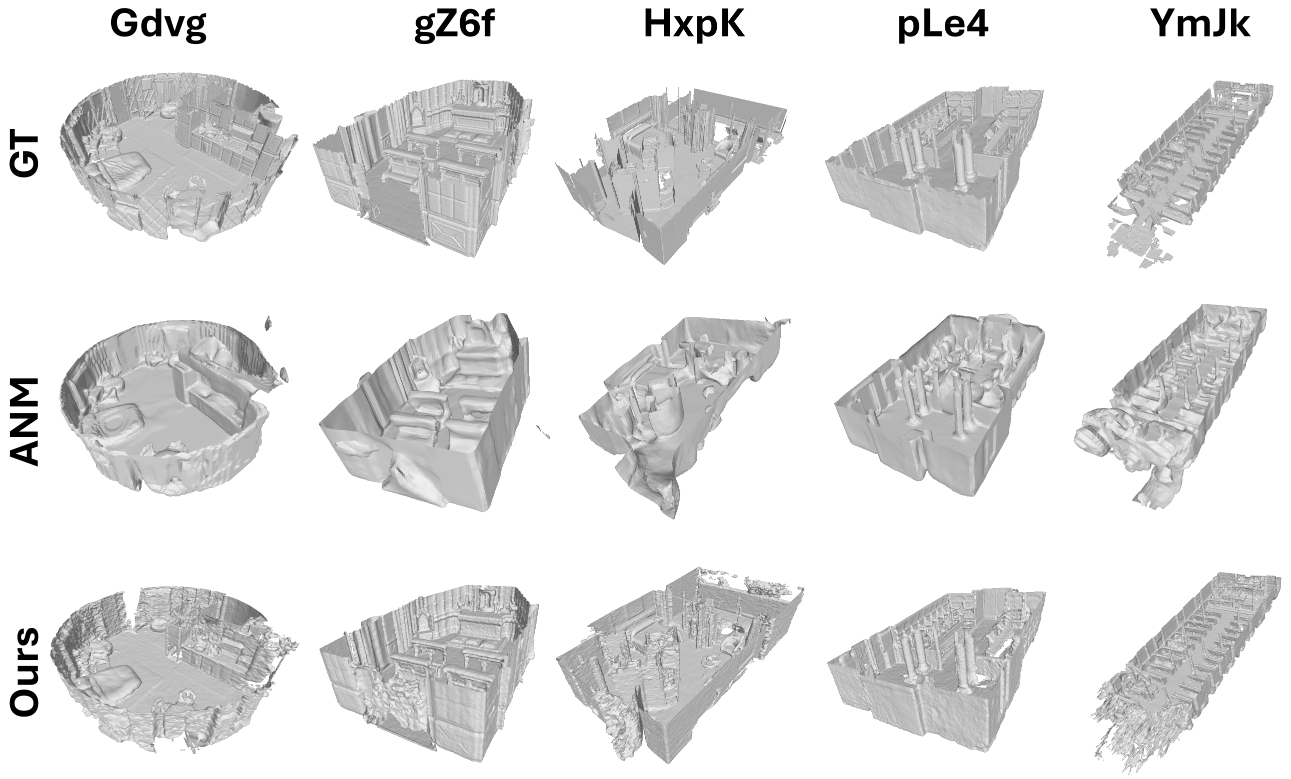

Figure 24. **More Matterport3D results** We trim the reconstruction results for a better comparison. Compared to the baseline method, ANM [79], our method shows more precise and complete reconstructions.

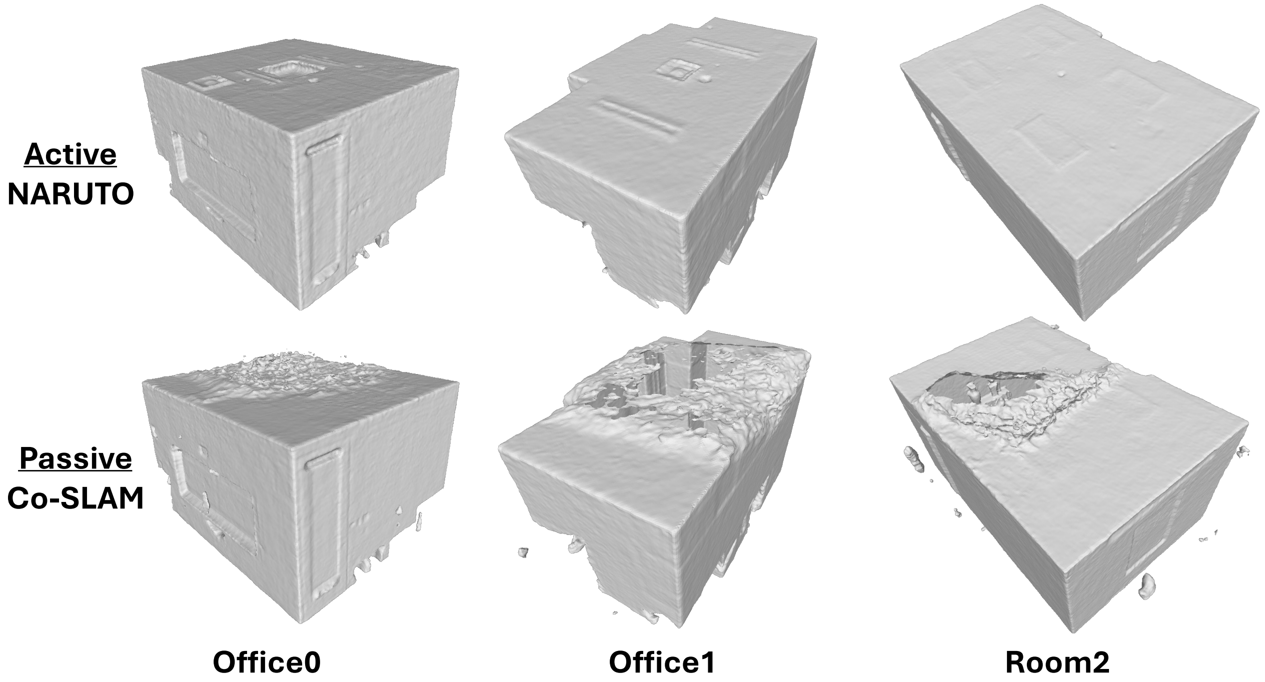

Figure 25. **Qualitative comparison between active and passive mapping methods.** For Co-SLAM [73], we disable the tracking thread and run the reconstruction using a pre-defined trajectory. Active NARUTO shows a more complete and precise reconstruction, especially for the regions that have not been adequately covered by the passive scanning.