A APPENDIX

A.1 IMPLEMENTATION DETAILS

The 2D occupancy map's resolution is 5cm. For each single frontier pixel on the 2D map, we add 200 3D Gaussians, which are uniformly distributed in the 3D cube above it. Other parameters like color, opacity, and scale are generated uniformly between 0 and 1. When there are frontiers on 2D map, we choose the next frontier to be explored by the area of each frontier divided by the distance. When no frontier exists, we select the top 20% of Gaussians with the highest score. These Gaussians are grouped using DBSCAN [9]. The largest cluster is selected for candidate pose generation. Candidates are uniformly sampled in the range between 0.3m to 1m, facing towards the selected position. Only the poses in free space are kept for path-level selection. The importance factor η in Eq. 17 is set to 5 across all experiments. The source code for this project will be made public no later than the publication of this paper.

We compute the Expected Information Gain (EIG) for each global candidate and use A* to plan a path for each of them. In order to prevent a twisted path, we consider locations 0.15m (3 pixels) away from the current robot position as neighbors and set the robot width to 3 pixels for collision check. However, the path planned by A* might have redundant waypoints, causing unnecessary turns for the robot. Therefore, we smooth the path by finding shortcuts. Specifically, for each waypoint w_i, if the path between waypoint w_{i+2} and w_i is collision-free, then we remove the intermediate waypoint w_{i+1} from the path. Finally, we use a greedy follower for motion planning. If the angle between the heading direction of the robot and the relative next waypoint is larger than $5°$, then we turn left or right to decrease the angle. Otherwise, we choose the forward action to approach the next waypoint. In such a way, we get a sequence of actions $\{a_i\}_{i=1}^{T}$ for each path.

Given a sequence of actions $\{a_i\}_{i=1}^{T}$ for each path, we use forward dynamics to compute the future camera poses $\{c_i\}_{i=1}^{T}$. Initially, we use an intermediate variable $\mathbf{H}''_{obs} \triangleq \mathbf{H}''[\mathbf{w}^*]$ to help compute expected information gain along the path. For each camera pose x_i, we compute its pose Hessian $\mathbf{H}''_{pose}$ and the current model Hessian matrix $\mathbf{H}''_{cur} \triangleq \mathbf{H}''[\mathbf{y}|x_i, \mathbf{w}^*]$. $\mathbf{H}''_{cur}$ is then accumulated, and we update $\mathbf{H}''_{obs}$ to evaluate the remaining poses on the path. We select the path that minimizes the objective given by Eq. 17 for execution.

A.2 RESULTS FOR EACH SCENE IN GIBSON AND HM3D DATASET

Following previous literature [61], we use the following scenes for Gibson Dataset: `Greigsville`, `Denmark`, `Cantwell`, `Eudora`, `Pablo`, `Ribera`, `Swormville`, `Eastville`, `Elmira`. For HM3D we use the following scenes: `DBjEcHFg4oq`, `mscxX4KEBcB`, `QKGMrurUVbk`, `oPj9qMxrDEa`, `CETmJJqkhcK`. The detailed results for each scene on each evaluation metric are presented as bar plots in Fig. 6 for Gibson and Fig. 7 for HM3D. We also present qualitative comparisons on testing views from the Gibson dataset in Fig. 8.
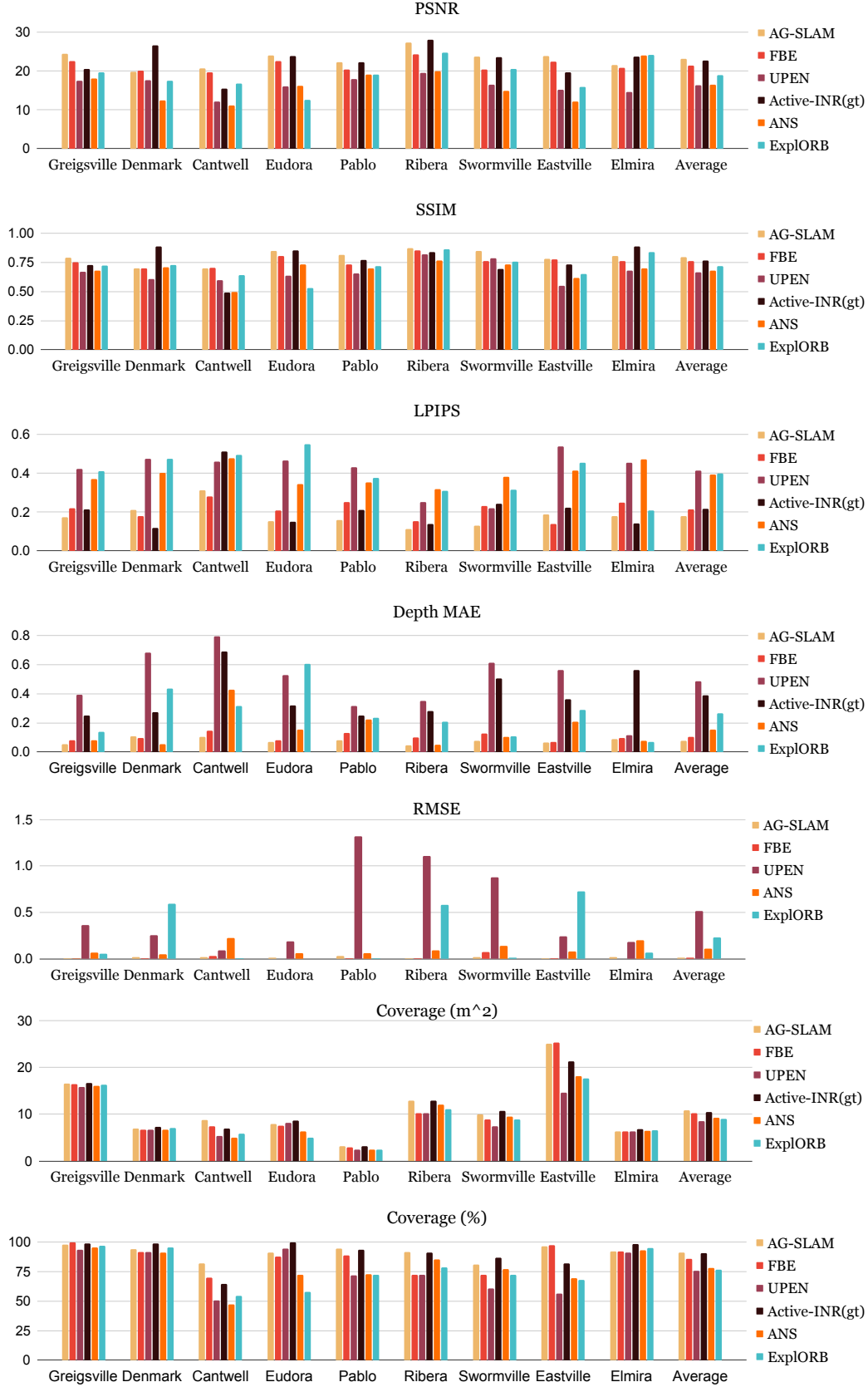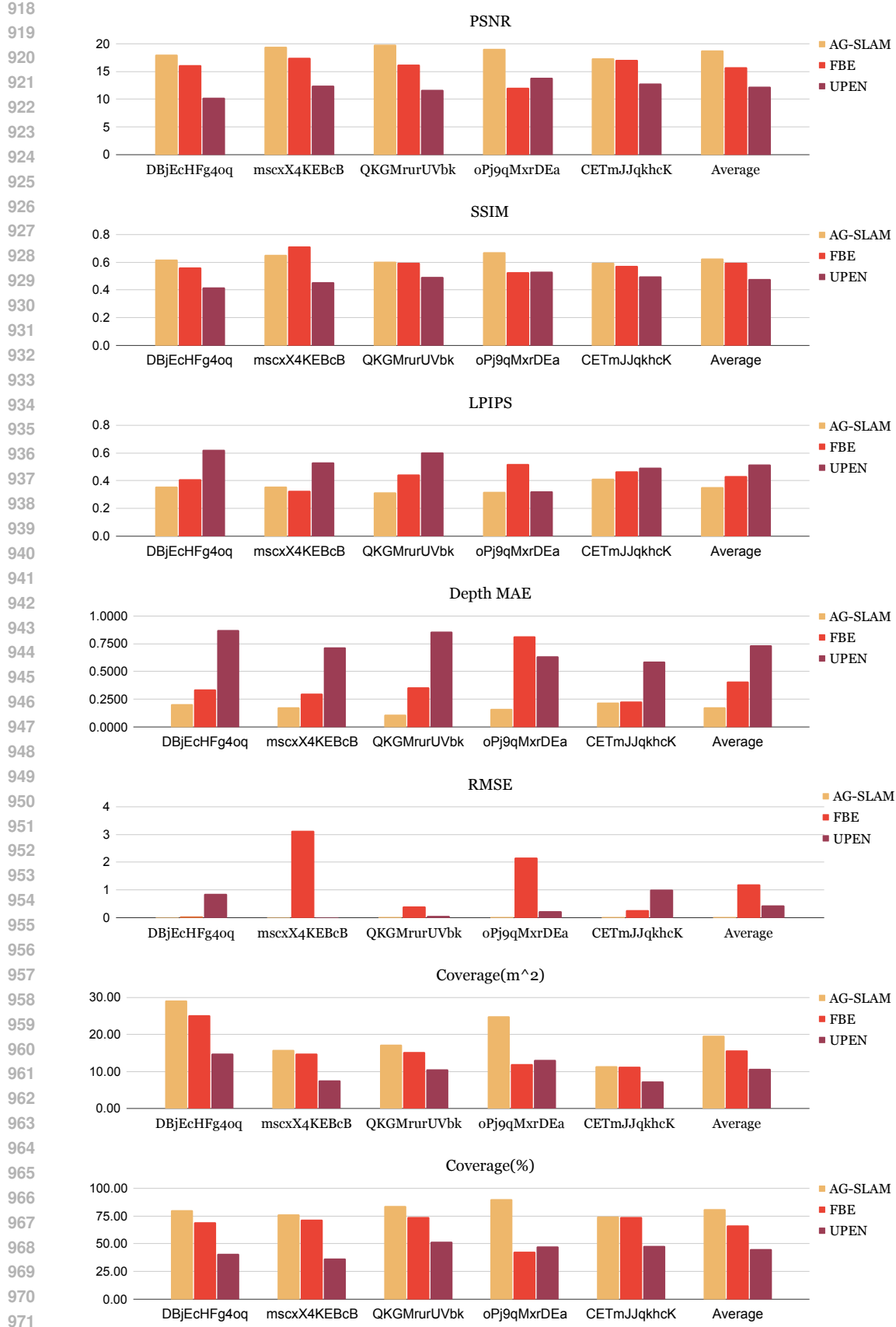
Figure 6: **Per-scene results on Gibson Dataset**

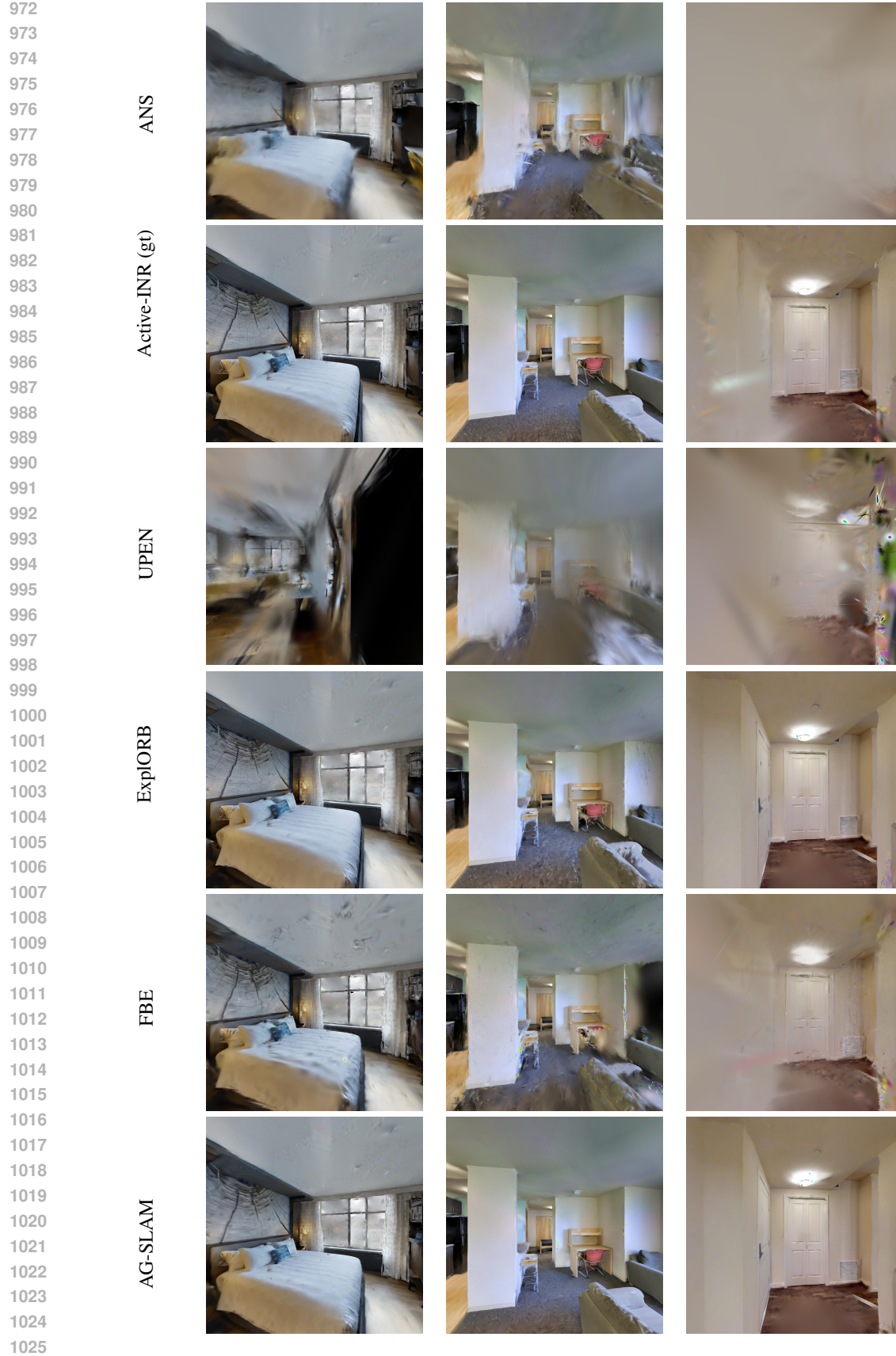Figure 7: **Per-scene results on HM3D Dataset**

Figure 8: **Test Rendering Qualitative Visualization on Gibson Dataset** All the renderings are from the test view of the Gibson dataset.