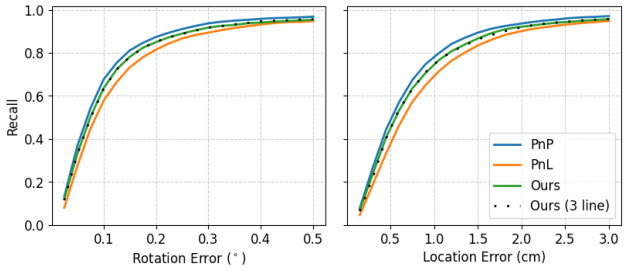
Supplementary Material: Privacy Preserving Localization via Coordinate Permutations

A. Private Map without Known Coordinate

As described in the main paper, exposing the information of one of the three coordinates can largely accelerate the process of localization due to the reduced number of configurations (2^6 versus 3^6). To show the effect of this, we conducted an experiment on the HoloLens dataset [7] in both scenarios of with and without exposing one coordinate. The results are summarized in Figure 1. From the table, one can observe similar median and AUC scores for both rotation and location errors in these two settings. However, the running times differ drastically by almost an order of magnitude. As all privacy attack results shown in this work are performed on the line clouds formed with one coordinate exposed, we do not observe any evident privacy concerns related to this setting.



	MED (°)	AUC@1°	MED (cm)	AUC@10cm	(1°, 10 cm)	Time (s)
PnL [6]	0.08	84.71	0.64	88.66	97.71	1.16
Proposed (w/)	0.07	86.31	0.56	89.71	97.92	6.61
Proposed (w/o)	0.08	86.32	0.55	89.78	98.02	59.55
PnP	0.07	87.85	0.51	90.84	98.83	0.07

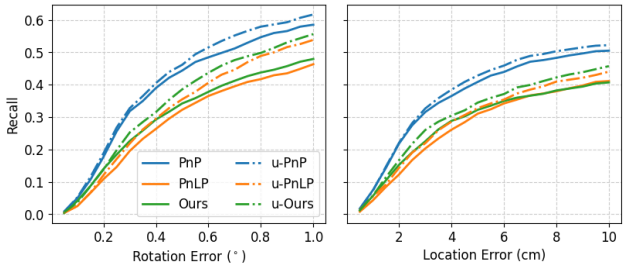
Figure 1: Localization performance on HoloLens dataset [7] in the private map scenario with and without exposing one coordinate. The accuracy in both cases are similar, however, without exposing one coordinate, the runtime increases by about an order of magnitude.

B. Localization with Gravity Prior

LaMAR [5]. This dataset is a recent large-scale augmented reality benchmark in unconstrained and challenging environments. The dataset provides image streams with realistic motions together with other heterogeneous sensor information and is well-suited for the task of localization and mapping. In our experiments, we obtained the reference reconstructions and image correspondences using the toolbox provided with the dataset [3, 4]. Furthermore, thanks to the onboard inertial measurement unit, gravity information is provided for both the reference reconstruction and the query sequences. We evaluate our method for private query localization on the HoloLens validation sequences with available ground truth trajectories. For this experiment, we restricted ourselves to the validation sequences in scene CAB, because the rest of the dataset is not yet fully publicly available. There are in total 1172 query images for this experiment, and each image has around 500 correspondences.

Cumulative distributions for rotation and location errors on scene CAB are summarized in Figure 2. For 6 DoF pose estimation, we can observe an improvement of recall by 12% within the tight threshold. This is achieved at the cost of about 10 times of the running time. Note that however, if the loose bound is considered, the recall of our method is slightly worse than that of the random line method. We attribute this loss in recall to the extremely low inlier ratios of correspondences in these hard to localize query images. Our method can benefit from p^2 extra geometric constraints as shown in the main paper where p is the inlier ratio. When the inlier ratio is below 20%, which was a frequent case for this dataset, the expected ratio of recoverable 2D point constraints is less than 4%. Such a low level of additional constraints does not add a sufficient improvement in the result. Additionally, since each point is associated with 2 lines, there are more false inliers considered in the optimization process, which can potentially diminish the accuracy.

A similar trend in accuracy can be observed in the results on 4 DoF camera pose estimation. For the tighter bound, the proposed method shows improvements in recall while for the loose bound, the recall of our method is lower. In the 4 DoF case, the runtime difference between our method and the random line method is significantly reduced as compared to the 6 DoF case. Since minimal solvers in this case only need 4 points, the total number of possible configurations is $2^4 = 16$. As such, the runtime of our method is less than 3 times of the random line method.



	(1°, 10 cm)	(5°, 1 m)	Time (s)
PnLP [7]	45.81	64.25	0.21
Proposed	46.63	63.49	2.50
PnP	52.68	67.93	0.09
u-PnLP [7]	53.24	73.63	0.11
u-Proposed	55.12	72.44	0.28
u-PnP	60.67	75.60	0.06

Figure 2: Localization performance on LaMAR [5] dataset with and without gravity information. Note that with gravity information, the runtime overhead of the proposed method drops from 12 times to less than 3 times.

C. Analysis of Potential Privacy Attacks

C.1. Private Queries

The point recovery mechanism described in the paper finds pairs by identifying rectangles where the corners are given by the projections and permuted points. There is, however, a potential for data leakage, as it is theoretically possible to identify the rectangle using only three of the corners. This means that for a swapped pair where only one of the points is an inlier (belonging to be map and is non-sensitive), we can potentially recover the other point position despite it being an outlier and thus not present in the map. For each inlier, the set of potential pairings is given by (see equation (7) in the main paper)

$$C = \{k \mid |x_{kj'} - \tilde{x}_{j'}| < \varepsilon, (x_{kj'}, k) \in S_{j'}\}, \qquad (1)$$

As also mentioned in the main paper, this set likely contains multiple candidates from points having similar x or y-coordinates as the projection, which is why we proposed to use the symmetric check.

In this section, we show that this symmetric check is actually necessary, and we experimentally investigate the effect of this type of inlier-outlier pair leakage. Our experiments show both that few outlier points can be recovered and that the success rate is low where many incorrect pairs are found. In our experiment, we use the ground-truth camera pose, which should give an upper-bound on what can be recovered in real localization problems. Using the given pose, we identify the inliers and run the original point recovery method (find the inlier-inlier pairs). For each inlier which was not paired, we then try to form pairs using the outlier points. We compare two approaches: a) for each inlier where we have a unique match, *i.e.* $|C| = 1$, we form the pair. b) For each inlier where have potential pairs $|C| > 0$, we take the one with the smallest residual.

Quantitative result for both approaches on the dataset we considered are summarized in Figure 3. We consider a point as correctly recovered if the recovered position is within a threshold to the original position. For the first approach, considering the recovered pairs consisting of an inlier and an outlier, the ratio of correctly recovered is about . However, the absolute ratio of such pairs is low, and is insignificant compared with the inlier-inlier ratio. As for the second approach, among all recovered inlier-outlier pairs, the correct points only account for about 50%. While the absolute ratio of such point is not high in this case, it is also hard to distinguish correctly recovered points with incorrect ones.

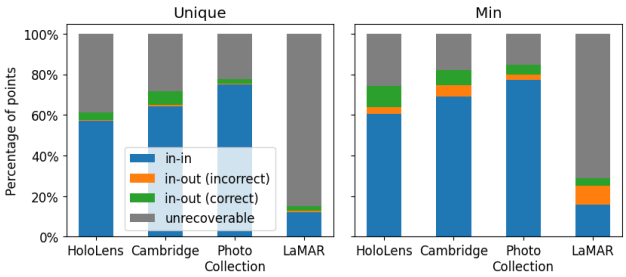
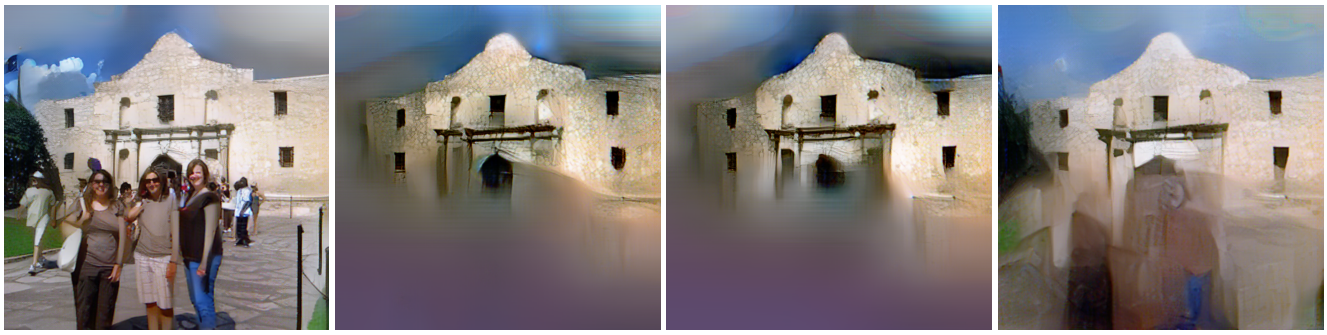


Figure 3: Privacy attack on query images with two different approaches. Percentage of recovered points of different While only a small portion of outliers can be recovered, a large proportion of which are incorrect.

To qualitatively show the impact of these additionally recovered points, image inversion attacks are summarized in Figure 4. In this experiment, we run the method by Pittaluga *et al*. [2]. The original keypoints with their descriptors are not privacy-preserving as realistic details can be observed in the synthesized image (Figure 4a). Compared with the results from points recovered with symmetric error validation (Figure 4b), the results from both attacks contain extra information due to the additionally recovered points. For the unique attack approach, since the absolute number of additionally recovered points is very small, there is no perceivable difference in the quality of the inverted image. In fact, it appears as if the incorrectly recovered points introduce extra blur to the synthesized result. As for the min attack approach, a larger difference between results can be observed, yet the large ratio of incorrectly recovered points makes the synthesized image preserve the privacy of any recognizable details.

C.2. Private Maps

In this paper, we proposed to lift points along axis-aligned lines, which poses a potential information leakage by orthogonal projection and image inversion along the major coordinate axes. To analyze the potential privacy risk



(a) Raw Features (b) Symmetric Error Recovery (c) Unique Attack (d) Min Attack

Figure 4: Inversion attacks for localization with private queries. Recovered image using: *(a)* the complete set of features, *(b)* points recovered from the proposed method which includes the symmetric check, *(c)* additional points recovered from inlier-outlier selecting points with unique matches, *(d)* taking the closest candidate.

to our method under such an attack, we synthesized images from the orthogonal projection using the method by Pittaluga *et al*. [2].

The point cloud we consider here has its major axes aligned with the coordinate system as shown in Figure 6a. A synthesized example for the case that an attacker has additional knowledge of which coordinates are correct can be found in Figure 6b. Note that this is a scenario that can be ruled out in practice, as it would require an attacker to intercept the coordinate swapping procedure. In such a unrealistic scenario, the synthesized image indeed leaks information, since every point projects to the correct position. This works especially well in the top-down view, where mostly only a single point projects to the same position in the inverted image. In the other directions, multiple structures in the scene project at the same locations in the image and thus confuse the inversion network.

The inversion result for the realistic scenario, where no swapping information is revealed, is shown in Figure 6c. In this case, the large number of corrupted points make the image blurry and unrecognizable.

To further protect the privacy of the permuted point cloud from the orthogonal projection attack, one can apply a rotation to the original point cloud and then permute the coordinates, as proposed in the main paper. This rotation can be random or specifically chosen to avoid *e.g*. a top down view of the scene. The rotation can then be applied to the estimated camera pose before returning the result to the localization client. In a scenario, where the private map is shared with clients for on-device localization, the rotation can be safely shared, as it cannot be used to "unrotate" the permuted point cloud. The inversion results on such rotated maps are illustrated in Figure 6d and Figure 6e. Under this setting, even if the attacker is provided with the correct coordinate, simple orthogonal projection does not reveal meaningful information.

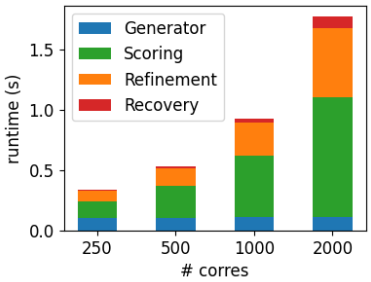
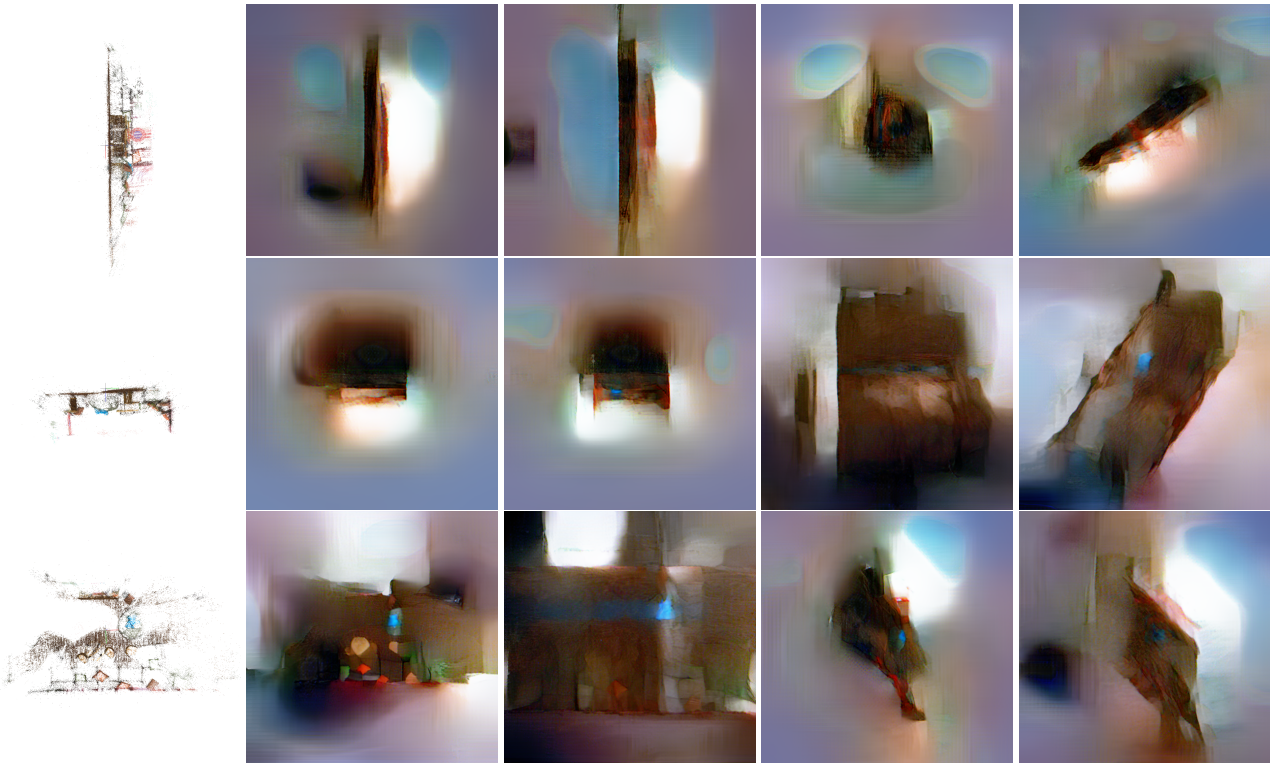


Figure 5: Runtimes for different stages in our RANSAC procedure.

D. Complexity Analysis of Point Recovery

The complexity of the point recovery mechanism is amortized linear on average for every pose query. For every point, we first hash their coordinates into chunked bins. By this, we can construct the S_1, S_2 in linear time on average. For the point recovery with a certain camera pose, the processes can be divided into the following steps: calculate the reprojection point position for each point, establish the candidate sets for each inlier, and calculate the symmetric error to determine whether each candidate is valid. It is easy to see that the first step is linear with respect to the number of points. For the second step, we only need to query neighboring chunks for points within small distance from reprojected point, which can be done in constant time using hashing. Furthermore, since the average number of points falling in these bins is constant, a constant number of additional checks will be conducted. Adding up the runtime of these steps gives us the amortized linear runtime.

In practice, we only trigger the process of point recovery when the number of inliers to the single-line constraint is promising ($> 80\%$ of the current best inlier count). And as Fig. 5 shows, the runtime overhead of point recovery is comparatively very small.



(a) Original (b) Oracle (w/o rot) (c) Proposed (w/o rot) (d) Oracle (w/ rot) (e) Proposed (w/ rot)

Figure 6: Orthogonal projection of the permuted point cloud. *(a)* The three view orthogonal projection on the original point cloud. *(b)* Results when the incorrect coordinate of each point is known. *(c)* The three view orthogonal projection results on the permuted point cloud. *(c)* The projection results of the rotated point cloud when the incorrect coordinate of each point is known. *(d)* Projection results of the rotated point cloud with permuted coordinate.

E. Comparison with Partial Localization

Geppert et al. [1] focus on on a different setting where the query and the map are in 3D. Their method is, in practice, expected to provide lower accuracy in case of image-based queries due to sub-optimal 3D optimization cost (compared to our image-based reprojection costs) and client-side composition of the orthogonal results. Table 1 shows a comparison on the 7 scenes dataset.

References

[1] Marcel Geppert, Viktor Larsson, Johannes Lutz Schönberger, and Marc Pollefeys. Privacy Preserving Partial Localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 4, 5

[2] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[3] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[4] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1

[5] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. Lamar: Benchmarking localization and mapping for augmented reality. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 686–704. Springer, 2022. 1, 2

[6] Pablo Speciale, Johannes L. Schönberger, Sing Bing Kang, Sudipta Sinha, and Marc Pollefeys. Privacy Preserving Image-Based Localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[7] Pablo Speciale, Johannes L. Schönberger, Sudipta N. Sinha, and Marc Pollefeys. Privacy preserving image queries for camera localization. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2

	PnP	PnLP	Ours		Partial Loc [1]
	cm / °	cm / °	cm / °	$\tau_1/\tau_2/\tau_3$	$\tau_1/\tau_2/\tau_3$
chess	2.52 / 0.87	2.57 / 0.89	2.56 / 0.88	23.5 / 100.0 / 100.0	24.0 / 97.5 / 99.5
fire	2.31 / 0.94	2.41 / 1.0	2.34 / 0.96	58.5 / 100.0 / 100.0	35.0 / 97.0 / 99.0
heads	1.01 / 0.78	1.11 / 0.84	1.04 / 0.79	90.0 / 93.0 / 96.0	39.0 / 77.0 / 82.0
office	3.23 / 0.95	3.51 / 1.01	3.37 / 0.98	7.0 / 72.2 / 100.0	16.8 / 59.0 / 95.2
pumpkin	5.31 / 1.4	5.59 / 1.44	5.38 / 1.43	1.0 / 86.0 / 97.5	0.0 / 51.0 / 90.5
kitchen	4.46 / 1.42	4.56 / 1.47	4.61 / 1.48	11.6 / 81.8 / 100.0	26.6 / 87.0 / 99.6
stairs	5.1 / 1.41	5.75 / 1.64	5.55 / 1.61	7.0 / 74.0 / 99.0	0.0 / 11.0 / 48.0

Table 1: Result for 7scenes. (cm / °) are median errors in camera location / rotation and $\tau_1/\tau_2/\tau_3$ are kept the same as [1]. Our method closes the accuracy/recall gap from Special *et al*. (PnLP) to the non-privacy preserving method (PnP), and also performs better to Geppert *et al*. (Partial Loc [1]).