

# Supplementary Material: Gaussian Splatting Feature Fields for (Privacy-Preserving) Visual Localization

Maxime Pietrantoni<sup>1,2,3</sup>

Gabriela Csurka<sup>3</sup>

Torsten Sattler<sup>2</sup>

<sup>1</sup> Faculty of Electrical Engineering, Czech Technical University in Prague

<sup>2</sup> Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague

<sup>3</sup> NAVER LABS Europe

{firstname.lastname}@cvut.cz, {firstname.lastname}@naverlabs.com

In this supplementary material, first in Sec. A, we provide additional explanations with regard to the contrastive losses, the optimal transport association procedure, and the volumetric feature encoding. In Sec. B, we detail the training and visual localization parameters. We also provide a pseudo-algorithm describing training in Alg. 1. Then, in Sec. C, we provide visual localization results on the 12Scenes dataset with ablation studies and in Appendix D we present a privacy attack experiment. Finally, in Sec. E we display more visualizations of rendered segmentation, while commenting on the attached supplementary videos.

## A. Technical details of GSFFs

### A.1. Contrastive losses and regularization terms

Here we provide more details regarding the derivation of the losses in Sec. 3.1 of the main paper. We recall that our is training the model in a self-supervised manner to align the feature maps  $\mathbf{F}^{3D}$  and  $\mathbf{F}^{2D}$ . Therefore, during training at each step we sample  $N$  pixels in these maps to be aligned. Let us denote the  $N$  corresponding pairs of pixel aligned extracted/rendered features by  $\{\mathbf{F}_n^{2D}, \mathbf{F}_n^{3D}\}_{n=1}^N$ . The contrastive loss has two terms. The first one is a term enforcing similarity of 2D extracted features with regard to 3D rendered features, and the second term is enforcing similarity of the 2D rendered features with regard to 3D extracted features:

$$L_{NCE} = -\frac{1}{N} \sum_{u=1}^N \log \left( \frac{\exp(\mathbf{F}_u^{3D} \cdot \mathbf{F}_u^{2D} / \tau)}{\sum_{j=1}^N \exp(\mathbf{F}_u^{3D} \cdot \mathbf{F}_j^{2D} / \tau)} \right) - \frac{1}{N} \sum_{u=1}^N \log \left( \frac{\exp(\mathbf{F}_u^{2D} \cdot \mathbf{F}_u^{3D} / \tau)}{\sum_{j=1}^N \exp(\mathbf{F}_u^{2D} \cdot \mathbf{F}_j^{3D} / \tau)} \right)$$

yielding  $L_{NCE}$  equal to:

$$-\frac{1}{N} \sum_{u=1}^N \log \left( \frac{\exp(\mathbf{F}_u^{3D} \cdot \mathbf{F}_u^{2D} / \tau) \exp(\mathbf{F}_u^{2D} \cdot \mathbf{F}_u^{3D} / \tau)}{\sum_{j=1}^N \exp(\mathbf{F}_u^{3D} \cdot \mathbf{F}_j^{2D} / \tau) \sum_{j=1}^N \exp(\mathbf{F}_u^{2D} \cdot \mathbf{F}_j^{3D} / \tau)} \right).$$

From this we can derive Eq. 2 in the main paper, where the normalization factor  $A$  is:

$$A = \left( \sum_{j=1}^N \exp(\mathbf{F}_u^{3D} \cdot \mathbf{F}_j^{2D} / \tau) \right) \left( \sum_{j=1}^N \exp(\mathbf{F}_u^{2D} \cdot \mathbf{F}_j^{3D} / \tau) \right)$$

Similarly, the prototypical contrastive loss has a term enforcing the similarity between 2D extracted features and the associated prototypes, and a second term enforcing similarity between 3D rendered features and the associated prototypes:

$$L_{PRO} = -\frac{1}{N} \sum_{n=1}^N \log \left( \frac{\exp((\mathbf{F}_n^{3D} \cdot \mathbf{p}_n) / \tau)}{\sum_{j=1}^K \exp(\mathbf{F}_n^{3D} \cdot \mathbf{p}_j / \tau)} \right) - \frac{1}{N} \sum_{n=1}^N \log \left( \frac{\exp((\mathbf{F}_n^{2D} \cdot \mathbf{p}_n) / \tau)}{\sum_{j=1}^K \exp(\mathbf{F}_n^{2D} \cdot \mathbf{p}_j / \tau)} \right)$$

yielding  $L_{PRO}$  equal to

$$-\frac{1}{N} \sum_{n=1}^N \log \left( \frac{\exp((\mathbf{F}_n^{3D} \cdot \mathbf{p}_n) / \tau) \exp((\mathbf{F}_n^{2D} \cdot \mathbf{p}_n) / \tau)}{(\sum_{j=1}^K \exp(\mathbf{F}_n^{3D} \cdot \mathbf{p}_j / \tau)) (\sum_{j=1}^K \exp(\mathbf{F}_n^{2D} \cdot \mathbf{p}_j / \tau))} \right)$$

which simplifies to Eq. 3 with the normalization factor:

$$B = \left( \sum_{j=1}^K \exp(\mathbf{F}_u^{3D} \cdot \mathbf{p}_j / \tau) \right) \left( \sum_{j=1}^K \exp(\mathbf{F}_u^{2D} \cdot \mathbf{p}_j / \tau) \right).$$

### A.2. Optimal transport (OT) associations

In this section, we detail the optimal transport association step from Sec. 3.2 of the main paper. Given a batch of  $N$  pairs of pixel aligned extracted/rendered features  $\{\mathbf{F}_n^{2D}, \mathbf{F}_n^{3D}\}_{n=1}^N$  and a set of  $K$  prototypes  $P \in \mathbb{R}^{K \times D}$ , we aim at associating a prototype per pair of pixel aligned features. We want this association operation to respect two criteria: 1) A single prototype must be associated per pair of features so that the extracted/rendered features are pushed toward the same "class" in the feature space. 2) Predictions must be as balanced as possible to avoid collapse.

**Data:**  $M$  posed training images,  $K$  target number of classes and a total number of training iterations  $N_{\text{iter}}$

Build SfM and pretrain a GoF [15] with the training images

Apply spectral clustering on the set of 3D Gaussian centers to initialize the spatial prototypes  $\{p_k\}_{k=1}^K$

Then, to train the GSFFs, iterate:

**for**  $iter$  in  $range(N_{\text{iter}})$  **do**

    Sample a random image  $I$  and viewpoint  $P$

    Extract 2D image-based features  $\mathbf{F}^{2D}$  and segmentation map  $\mathbf{S}^{2D}$

    For each 3D Gaussian  $\mathcal{G}_i$ , extract a scale aware volumetric feature  $\mathbf{g}_i$  from the triplane using the covariance kernel based encoding (cf. Sec. 3.1)

    Assign a 3D segmentation label  $s_i$  to each 3D Gaussian  $\mathcal{G}_i$  based on volumetric feature/prototypes similarities (cf. Sec. 4)

    From the pose  $P$  rasterize 3D segmentation labels and volumetric features to obtain  $\mathbf{S}^{3D}$  and  $\mathbf{F}^{3D}$

    Sample a novel viewpoint  $\hat{P}$ , find correspondences between image  $I$  and  $\hat{I}$  (cf. Sec. 3.2).

    Render 3D features/segmentation maps from  $\hat{P}$  and replace features/segmentation in  $\mathbf{F}^{3D}$ ,  $\mathbf{S}^{3D}$  with  $\hat{\mathbf{F}}^{3D}$ ,  $\hat{\mathbf{S}}^{3D}$  at correspondence location

    Repeat the replacements process for 2D features/segmentations

    Compute the contrastive loss  $L_{NCE}$  from  $\mathbf{F}^{2D}$  and  $\mathbf{F}^{3D}$

    Associate a prototype  $p_k$  to each pair  $\{\mathbf{F}_i^{2D}, \mathbf{F}_i^{3D}\}$  using the OT labeling procedure (cf. Appendix A.2)

    Compute  $L_{PRO}$ ,  $L_{CE}$  from these associations as well as from  $\mathbf{F}^{2D}$ ,  $\mathbf{F}^{3D}$ ,  $\mathbf{S}^{2D}$ ,  $\mathbf{S}^{3D}$  (cf. Sec. 3.2)

    Compute regularization losses  $L_{TVL}$ ,  $L_{Depth}$  and photometric loss  $L_{PHO}$  from the rendered image

    Jointly update the image encoder, the triplane field and 3D Gaussians by minimizing

$$L = L_{PHO} + .5L_{NCE} + .5L_{PRO} + .5L_{CE} + .1L_{TVL} + 0.05L_{Depth}$$

    Update the prototypes  $p_k$  with an EMA scheme based on volumetric features  $\mathbf{g}_i$  and the spectral clustering assignments

**end**

**Algorithm 1:** Pseudo algorithm describing the training process of GSFFs.

To solve these constraints, we resort to using optimal transport, where we frame this problem as finding a mapping  $Q \in \mathbb{R}^{N \times K}$  between pixels and prototypes that maximizes the feature similarity between the pairs of features and the prototypes. We define the joint feature/prototypes similarities  $S \in \mathbb{R}^{K \times N}$  as:

$$S_{kn} = \exp((\mathbf{F}_n^{2D} \cdot \mathbf{p}_k + \mathbf{F}_n^{3D} \cdot \mathbf{p}_k)/\tau)/C,$$

with

$$C = (\sum_k \exp(\mathbf{F}_n^{2D} \cdot \mathbf{p}_k/\tau))(\sum_k \exp(\mathbf{F}_n^{3D} \cdot \mathbf{p}_k/\tau)).$$

We introduce the following objective where  $Q$  maximizes the joint feature/prototypes similarities  $S$ :

$$\max_{Q \in U(\frac{1}{N}, \frac{1}{K})} \text{Tr}(Q(-\log S)^t) + \lambda h(Q),$$

where the entropy term  $h(Q)$  encourages balanced predictions while using joint extracted/rendered feature prototypes similarities yields a single association per feature pair. If we relax  $Q$  such that it belongs to the transportation polytope  $U(\frac{1}{N}, \frac{1}{K})$  [3],  $Q$  can be efficiently computed with the iterative Sinkhorn-Knopp algorithm [3]. The final associations  $\Gamma \in \mathbb{R}^N$  are obtained with  $\Gamma = \text{argmax}_k(Q)$ .

### A.3. Projecting 3D Gaussians onto the Triplane

In this subsection, we explain how a volumetric feature for each 3D Gaussian is derived from the triplane grid (Sec. 3.1

of the main paper). The triplane grid is centered at the origin of the world coordinate space and it is composed of three orthogonal 2D planes  $H_{xy}, H_{xz}, H_{yz} \in \mathbb{R}^{D \times R \times R}$ ,  $D$  being the triplane feature dimension and  $R$  the resolution of the grid. We project each 3D Gaussian  $\mathcal{G}_i$  onto these planes and derive three Gaussian kernels  $\mathcal{G}_i^{xy}, \mathcal{G}_i^{xz}, \mathcal{G}_i^{yz}$  from the projections, which we use to obtain scale aware volumetric features  $\mathbf{g}_i^{3D}$ .

To obtain the  $xy$  feature, we proceed as follows ( $xz$  and  $yz$  features are obtained similarly). Let  $m_i$  be the center of  $\mathcal{G}_i$  and  $\Sigma_i$  its covariance matrix. We first project the center on the plane yielding  $m_i^{xy}$ . We perform orthographic projection of the covariance matrix to obtain  $\Sigma_i^{xy}$ . We define a grid of dimension 5 by 5 centered on  $m_i^{xy}$ . On the plane  $xy$ , we use the coordinates of the points in the grid  $u$  to define the following Gaussian kernel:

$$\mathcal{G}_i^{xy}(\mathbf{u}) = \frac{1}{Z} \exp(-\frac{\mathbf{u}(\Sigma_i^{xy})^{-1}\mathbf{u}^t}{2}),$$

where  $Z$  is a normalization constant. We query the feature plane  $H_{xy}$  for each point  $u_k$  in the grid and apply the Gaussian kernel on the queried features. This yields a  $D$ -dimensional feature  $\mathbf{g}_i^{xy}$  associated to  $\mathcal{G}_i^{xy}$ . We repeat this operation for  $H_{xz}$ ,  $\mathcal{G}_i^{xz}$  and  $H_{yz}$ ,  $\mathcal{G}_i^{yz}$ . The resulting features  $\mathbf{g}_i^{xy}, \mathbf{g}_i^{xz}, \mathbf{g}_i^{yz}$  are summed to obtain the volumetric feature  $\mathbf{g}_i^{3D}$  of  $\mathcal{G}_i$ .

## B. Implementation details

### B.1. Additional training details

Similar to GoF [15], densification and pruning operations based on image space gradients are applied until iteration 15000. The densification interval is set to 600 iterations until iteration 7500, and reduced to 400 iterations afterward. To facilitate the convergence of the 3D Gaussian model, we train on images downsampled by a factor 4 until iteration 7500. From iteration 15000, the geometric regularization losses from [15] are applied until the end of the training. The triplane learning rate is set to  $7e-3$ , while the encoder learning rate is set to  $1e-4$ . The learning rates for 3D Gaussian primitives are identical to GoF [15].

GSFFs is optimized with the Adam optimizer [4]. The prototypes are updated based on an exponential moving average (EMA) scheme with  $\alpha = 0.9995$  after each training iteration. The temperature  $\tau$  for the contrastive losses is set to 0.05. In the *Multi-view consistency* paragraph from Sec. 3.2, the pixel reprojection threshold is set to 2 for the fine level, and to 4 for the coarse level. The coarse encoder uses a Dinov2 [6] pretrained backbone, followed by projection convolutional layers (convolutional layer with kernel size 1 to reduce the dimension, while maintaining the resolution) and a ConvNeXt block [5]. The fine encoder is composed of a shallow convolutional layers followed by a ConvNeXt block [5]. The segmentation heads contain convolutional layers with ReLU activations and GroupNorm [14] normalization.

We provide in Alg. 1 the pseudo algorithm describing the training process of the GSFFs.

### B.2. Data pre-processing

To reduce running time and the memory footprint, images are rescaled such that image width is 1024 pixels for Cambridge Landmarks and 480 pixels for Indoor6. The original image resolution of 640 by 480 pixels is used on 7Scenes. During visual localization, we use the same image resolution as the one used during training.

Cambridge Landmarks and Indoor6 contain images with illumination changes, as such we learn an embedding per training image to capture these illumination changes during the training. These embeddings are only used during training to learn the scene representation. On Cambridge Landmarks during training, we mask out the sky and pedestrians. The masks are extracted using the semantic segmentation model from [9]. As Indoor6 contain day/night images with extreme illumination changes we further apply CLAHE normalization on images.

### B.3. Visual localization setup

For the pose refinement, we use the Adam optimizer [4] with coarse/fine learning rates of 0.5/0.2 on Cambridge

	N Classes	KingsCollege	OldHospital	ShopFacade	StMarysChurch
FP	34	0.034	0.032	0.024	0.028
	84	0.046	0.043	0.027	0.031
BP	34	0.065	0.065	0.071	0.076
	84	0.462	0.401	0.287	0.392

Table 1. Computation time (s) for a forward pass FP (rendering) and a backward pass BP (pose optimization) on Cambridge Landmark for GSFFs trained with 34 classes and 84 classes.

	Training time (h)	KC	OH PSNR / SSIM / LPIPS	SF PSNR / SSIM / LPIPS	SMC
GoF [81]	0.7	16.95/0.69/0.27	16.98/0.63/0.30	20.04/0.74/0.21	18.49/0.71/0.25
GSFFs	10.8	16.92/0.69/0.27	16.94/0.61/0.30	20.02/0.74/0.21	18.47/0.71/0.26

Table 2. Evaluating novel view rendering.

Landmarks, 0.3/0.2 on Indoor6, and 0.2/0.1 on 7Scenes respectively. The number of refinement steps for the coarse and fine level is set to 150/300 on Cambridge Landmarks and Indoor6, and 75/150 on 7Scenes. Rendered areas with high distortion (see [15] for the definition of distortion) are masked out during refinement. Additionally, the sky is masked out on Cambridge Landmarks.

### B.4. Training time and rendering quality

In Tab. 2 we report novel view rendering quality evaluated with PSNR, SSIM [13] and LPIPS [18] image metrics as well as the corresponding model training time for the Cambridge Landmark scenes. From these results in Tab. 2, we observe that GSFFs-Feature, compared to GoF [15], adds computational overhead (the cost of training the feature fields), but it does not decrease the novel view synthesis quality. Note that GSFFs-Privacy cannot render RGB images for privacy reasons.

We also provide rendering time and backward pass time for the pose optimization in Tab. 1 for 34 and for 84 classes. We can observe that while the forward pass is only slightly increased, the cost of backward pass increases significantly with the increase the number of classes. Globally, through our experiments, we found that in general using 34 classes is a good compromise between rendering speed and pose accuracy (see the accuracies in Tab. 4 in the main paper and Tab. 1 for the training times).

## C. Additional Localization Experiments

### C.1. 12Scenes Dataset

In Tab. C.1, we report localization results on the 12Scenes [12] dataset for both the SfM pseudo ground truth (pGT) and the DSLAM pGT. We compare our GSFFs-Feature model against three non privacy preserving feature rendering based-approaches methods NeRF-SCR [2], PNeRFLoc [19], NeuraLoc [17] and GSplatloc [16]. GSFFs is more accurate than all baselines and clearly outperforms the rendering based-approaches [2, 19], while providing accuracy

Model	kitchen	living	bed	kitchen	living	lute	gates362	gates381	lounge	manolis	office2 5a	office2 5b
GSFFs-PR Feature (34 Classes) (NV)	0.3/0.2/99	0.3/0.18/100	0.4/0.17/100	0.7/0.42/91	0.4/0.21/96	0.6/0.27/97	0.5/0.23/100	0.5/0.27/99	0.8/0.29/97	0.5/0.22/99	0.9/0.41/99	1.1/0.41/94
GSFFs-PR Privacy (34 Classes) (NV)	0.6/0.32/97	0.6/0.29/100	0.5/0.23/100	0.9/0.51/75	0.7/0.30/99	0.8/0.30/96	0.8/0.31/98	0.7/0.36/98	1.6/0.54/91	0.7/0.31/97	1.3/0.65/95	1.8/0.83/69
NeRF-SCR [2]	0.9/0.5	2.1/0.6	1.6/0.7	1.2/0.5	2.0/0.8	2.6/1.0	2.0/0.8	2.7/1.2	1.8/0.6	1.6/0.7	2.5/0.9	2.6/0.8
PNeRF-LOC [19]	1.0/0.6	1.5/0.5	1.2/0.5	0.8/0.4	1.4/0.5	8.1/3.3	1.6/0.7	8.7/3.2	2.3/0.8	1.1/0.5	X	2.8/0.9
GSPlatLoc [16]	0.8/0.4	1.1/0.4	1.2/0.5	1.0/0.5	1.2/0.5	1.5/0.6	1.1/0.5	1.2/0.5	1.6/0.5	1.1/0.5	1.4/0.6	1.5/0.5
NeuralLoc [17]	0.9/0.5	1.1/0.4	1.3/0.6	1.0/0.6	1.2/0.5	1.4/0.7	1.1/0.5	1.1/0.5	1.7/0.6	1.0/0.5	1.3/0.6	1.5/0.5
GSFFs-PR Feature (34 Classes) (NV)	0.7/0.4	1.1/0.4	1.1/0.4	0.8/0.4	1.0/0.5	1.3/0.5	1.1/0.4	1.2/0.5	1.4/0.5	0.8/0.4	1.2/0.5	1.7/0.6

Table 3. Localization results on 12Scenes (SfM pGT top rows / DSiam pGT bottom rows). Median pose error (cm.) ( $\downarrow$ ) / Median angle error ( $^\circ$ ) ( $\downarrow$ ) / Recall at 5cm/5 $^\circ$  (%) ( $\uparrow$ ).

improvement compared to [16, 17] on most of the scenes.

## C.2. Varying the number of segmentation classes

In this section, we study the influence of the number of classes/prototypes on both the feature- and the segmentation-based variants of our approach. All experiments in the main paper use a feature dimension of 16 with 34 segmentation classes. As we render both features and segmentations in the same variable (each channel is independently rendered), we compile the rasterizer with a fixed rendering dimension of 50 (16 + 34) yielding a good compromise between rendering speed and discriminative power.

In this ablation, we maintain a feature dimension of 16 and vary the number of classes. We compile the rasterizer with a map dimension of 100 and then train and evaluate GSFFs-PR with 84 classes, 59 classes, and 15 classes. In Fig. 1, we plot the median translation and rotation errors on Cambridge Landmarks against the number of classes of each model. In Tab. 4 (main paper) we compare 34 classes versus 84 on the Indoor 6 dataset). From these results, we derive the following observations. Using only a few classes is not sufficient because the lack of discriminative power. Increasing the number of classes first yields a significant gain, however above a certain limit we observe a drop in accuracy. We suspect that the reason is over-clustering and hence more difficulty for the representation to converge during the training. Overall, the number of classes must be high enough to make the segmentation discriminative enough for localization, but not too high to ensure the convergence. Naturally, larger scenes with diverse viewpoints will require more classes, while for simpler scenes it is better to consider less classes.

Furthermore, we can see from Tab. 4 (main paper) that GSFFs-PR Feature localization pipeline also benefits from the increased number of classes as, during training, gradients are backpropagated from segmentation and feature maps and  $L_{PRO}$  implicitly uses the prototypes.

## C.3. Effect of the initialization

In Tab. 4 we provide pose refinement results on Cambridge Landmarks with different initial poses. Note that poses estimated with ACE [1] and HLoc [10] are much more accurate than DenseVLAD(DV) [11]. We can observe that, as expected, improving the initial pose results in higher final

	Init.	Image Res.	Coarse Fine Steps	Runtime Query (s)	KC	OH	SF	SMC
					cm / deg			
[64]	[64]	?	350	3	27/0.46	20/0.71	5/0.36	16/0.61
[40]	Ace	512	x	0.2	25/0.29	26/0.38	5/0.23	13/0.41
DV	-	-	-	-	280/5.7	401/7.1	111/7.6	231/8
Vanilla-GS	DV	1024	150-300	45	21.9/0.34	22.3/0.42	4.5/0.27	11.8/0.35
GSFFs-PR	DV	1024	150-300	45	17.9/0.27	21.4/0.41	4.1/0.26	10.4/0.30
GSFFs-PR	DV	480	150-300	8.1	19.7/0.27	21.7/0.36	4.7/0.23	8.7/0.29
GSFFs-PR	DV	480	50-50	1.8	74.6/1.26	162/2.81	16.4/0.73	90.2/2.68
ACE	-	-	-	-	28.0/0.4	31.0/0.6	5.0/0.3	18.0/0.6
GSFFs-PR	ACE	1024	150-300	45	16.1/0.22	17.9/0.34	3.9/0.18	7.6/0.23
GSFFs-PR	ACE	480	150-300	8.1	17.8/0.22	18.7/0.33	4.7/0.21	8.5/0.25
GSFFs-PR	ACE	480	75-150	4	18.2/0.24	21.1/0.35	4.8/0.22	8.5/0.25
GSFFs-PR	ACE	480	50-50	1.8	19.1/0.27	25.4/0.49	5.1/0.26	11.1/0.35
GSFFs-PR	ACE	480	25-25	0.9	19.8/0.44	25.6/0.66	5.8/0.45	13.2/0.53
Hloc	-	-	-	-	11.1/0.20	15.5/0.31	4.4/0.2	7.1/0.24
GSFFs-PR	Hloc	1024	0-50	5	10.9/0.18	14.1/0.29	3.9/0.19	6.4/0.21
GSFFs-PR	Hloc	480	0-50	0.9	11.0/0.19	14.2/0.30	3.9/0.18	6.4/0.22

Table 4. Varying initialization and image resolution.

GSFFs-PR localization accuracy. It also requires less refinement steps to converge. Starting from a wide baseline such as DenseVlad [11] the model needs more refinement steps but the optimization ultimately reaches high accuracy (especially for high resolution images) showing the robustness of GSFFs-PR.

## C.4. Varying resolution and refinement steps

In Tab. 4 we also provide results with different coarse/fine number of refinement steps, image resolution and runtime per query. We can see that performances on high resolution images is slightly better than on low resolution images, but this comes at a higher inference cost. We can further decrease the running time by decreasing the refinement steps. The loss in performance is relatively small conditioned that we start from a good initialization. This suggest that we could further increase the localization speed by combining low and high resolution based refinement and stopping the optimization earlier.

## D. Privacy attack

To visualize and assess the degree of privacy of GSFFs-PR-Privacy, we train an inversion model [7, 8] to reconstruct images from rendered segmentations. As a baseline, we train another inversion model to reconstruct images from feature maps rendered from GSFFs-PR-Feature. Both inversions model are trained on 6 scenes (*fire*, *heads*, *office*, *pumpkin*, *redkitchen*, *stairs*) from the 7Scenes dataset and evaluated the remaining scene *chess*. Example reconstructed images from both GSFFs-PR-Feature and GSFFs-PR-Privacy and displayed in Fig. 2. We can observe that



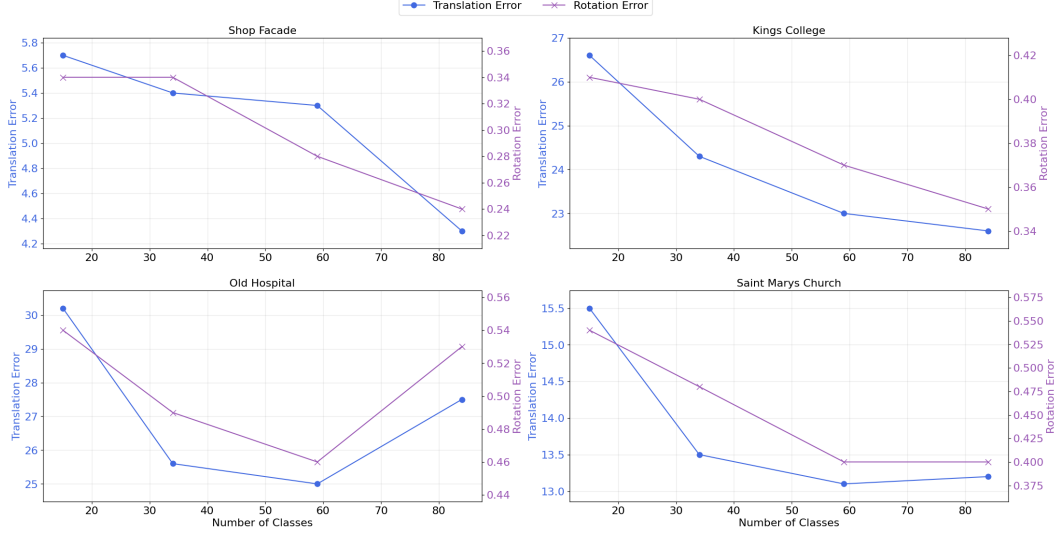


Figure 1. Median pose error (cm.) ( $\downarrow$ ) and Median angle error ( $^{\circ}$ ) ( $\downarrow$ ) on Cambridge Landmarks for models trained with 15/34/59/84 classes.



Figure 2. Left to right: original image, image inversion attack from rendering our features (middle, GSFFs-PR Feature) or rendering our segmentation (right, GSFFs-PR Privacy).

images reconstructed from GSFFs-PR-Feature reveal a lot of scenes details, while images reconstructed from GSFFs-PR-Privacy totally obfuscate privacy sensitive information.

## E. Visualizations

We show in Fig. 3 pairs of 2D extracted / 3D rendered segmentations for the coarse and fine levels while comparing segmentations from models trained with 34 and 84 classes. The segmentation classes of the coarse level capture larger and less sharply defined segments in the image compared to the fine level segmentations. As shown in the ablation in Tab. 3 of the main paper, this allows us to increase the convergence basin of the pose refinement, while improving fine accuracy. Using 84 classes instead of 34 results in visually finer-grained segmentations in turn allows for more

accurate pose refinement.

We also attach to this Supplementary three videos called *Shop\_Facade\_renderings.mp4*, *Kings\_College\_renderings.mp4* and *Chess\_renderings.mp4* where we render images, feature maps and segmentation maps for trajectories in *Shop Facade*, *KingsCollege* and *Chess*. Features are visualized through PCA down-projection. Both GSFFs’s feature and segmentation maps are very stable under wide viewpoint changes, which is a critical property for accurate pose refinement. We can furthermore see that the segmentation classes do not capture, in general, any semantic concepts showing a high degree of privacy of our model.

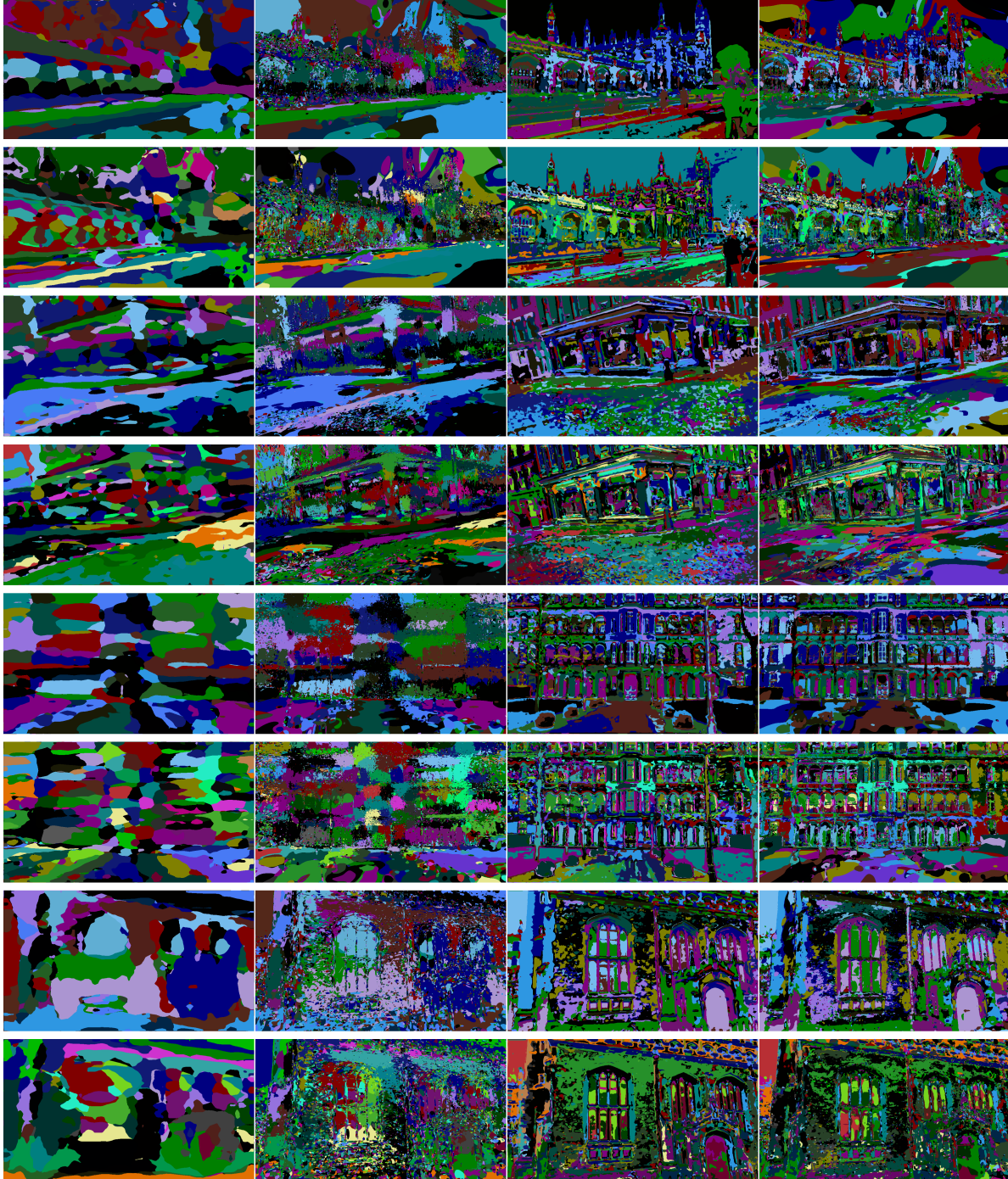


Figure 3. From left to right, coarse encoder/segmentation, fine encoder/segmentation. Comparison between models trained with 34 classes (line 1/3/5/7) and models trained with 84 classes (line 2/4/6/8). 2D extracted and 3D rendered segmentations are well aligned which allows for accurate privacy preserving visual localization.

## References

- [1] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated Coordinate Encoding: Learning to Relocalize in Minutes using RGB and Poses. In *CVPR*, 2023.
- [2] Le Chen, Weirong Chen, Rui Wang, and Marc Pollefeys. Leveraging neural radiance fields for uncertainty-aware visual localization. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6298–6305. IEEE, 2024.



2024. 3, 4
- [3] Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *NeurIPS*, 2013. 2
  - [4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3
  - [5] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 3
  - [6] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3
  - [7] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. SegLoc: Learning Segmentation-Based Representations for Privacy-Preserving Visual Localization. In *CVPR*, 2023. 4
  - [8] Francesco Pittaluga, Sanjeev J. Koppal, Sing Bing Kang, and Sudipta N. Sinha. Revealing Scenes by Inverting Structure from Motion Reconstructions. In *CVPR*, 2019. 4
  - [9] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction. In *ICCV*, 2021. 3
  - [10] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019. 4
  - [11] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomáš Pajdla. 24/7 Place Recognition by View Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(2):257–271, 2018. 4
  - [12] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 323–332. IEEE, 2016. 3
  - [13] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612, 2004. 3
  - [14] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3
  - [15] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024. 2, 3
  - [16] Hongjia Zhai, Xiyu Zhang, Boming Zhao, Hai Li, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. SplatLoc: 3D Gaussian Splatting-based Visual Localization for Augmented Reality. *arXiv preprint arXiv:2409.14067*, 2024. 3, 4
  - [17] Hongjia Zhai, Boming Zhao, Hai Li, Xiaokun Pan, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Neuraloc: Visual localization in neural implicit map with dual complementary features. *arXiv preprint arXiv:2503.06117*, 2025. 3, 4
  - [18] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018. 3
  - [19] Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. Pnerfloc: Visual localization with point-based neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7450–7459, 2024. 3, 4