MATCHED, BUT WHERE? QUERY-CONDITIONED POINT PREDICTION FOR 3DGS-BASED VISUAL LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) has emerged as a powerful scene representation for visual localization, enabling direct matching between 2D image keypoints and 3D Gaussian primitives as well as efficient rendering for pose refinement. Existing direct matching methods typically assign the center of a matched Gaussian to the query keypoint. However, a Gaussian represents a volumetric region, and its center generally differs from the observed scene point. That is to say, a correct Gaussian match can therefore yield an inaccurate 3D coordinate, compromising PnP-based pose estimation. To address this limitation, we propose ReCoLoc, a region-to-coordinate framework that explicitly separates region retrieval from point prediction by equipping Gaussian landmarks with query-conditioned coordinate functions. Specifically, contribution-weighted identity encoding aggregates multi-view descriptors using Gaussian rendering contributions, yielding reliable regional identities for retrieval. Query-conditioned coordinate decoding then predicts the corresponding 3D position within the retrieved Gaussian region from the query descriptor, thereby reducing the geometric error introduced by center-based assignment. Considering that retrieval and coordinate prediction both determine correspondence quality, correspondence-aware landmark selection jointly evaluates their reliability when constructing the localization map. Also, we introduce shared-subspace map compression, which jointly encodes identity descriptors and coordinate functions in a common low-dimensional basis, reducing direct-matching map storage substantially with little loss of localization accuracy. Experiments on Cambridge Landmarks, 7-Scenes, and 12-Scenes demonstrate that ReCoLoc outperforms current state-of-the-arts in terms of localization accuracy and recall.

1 INTRODUCTION

Visual localization estimates the 6-DoF camera pose of a query image within a pre-built scene map for applications such as augmented reality, autonomous driving, and robotics. The choice of scene representation determines how image observations are linked to the mapped environment. Structure-based methods establish correspondences with reconstructed 3D points (Schönberger & Frahm, 2016; Sarlin et al., 2019), whereas scene coordinate regression methods encode scene geometry in neural networks and predict 3D coordinates from image observations (Brachmann & Rother, 2022; Brachmann et al., 2023).

More recently, advances in neural rendering have introduced scene representations that jointly model geometry and appearance (Mildenhall et al., 2020). Among them, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) combines explicit 3D primitives with efficient rendering, making it particularly attractive for visual localization. By associating visual descriptors with Gaussian primitives, recent methods directly match 2D query keypoints to the 3D scene and recover the camera pose using a robust Perspective-n-Point (PnP) solver. The same representation also enables rendering-based pose refinement. For instance, STDLoc (Huang et al., 2025) obtains an initial pose from sparse 2D-3D matches and refines it through dense matching against rendered feature maps. Subsequent work improves the Gaussian landmarks used in the direct matching stage. ULF-Loc (Gu et al., 2026) constructs more reliable landmark descriptors through geometry-weighted multiview feature fusion,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

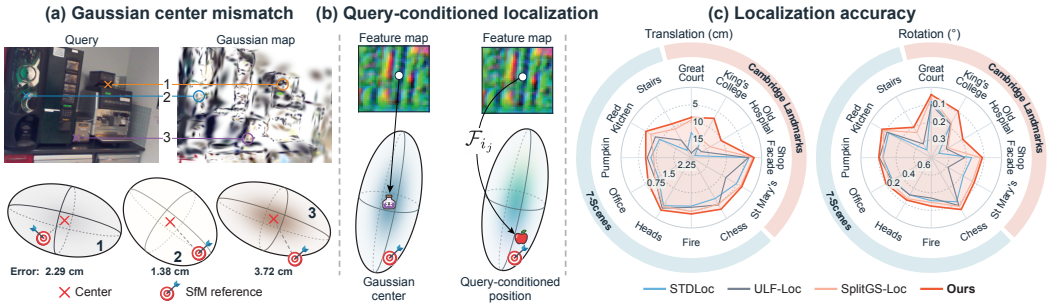

Figure 1: **Motivation and localization without pose refinement.** (a) Matched Gaussian centers deviate from the SfM reference points of the query keypoints. (b) ReCoLoc replaces the center with a query-conditioned position. (c) Per-scene median errors on Cambridge and 7-Scenes (outer is better).

while SplitGS-Loc (Lee et al., 2026) splits spatially extended Gaussians into smaller primitives to reduce matching ambiguity.

Nevertheless, these methods still assign the center of a matched Gaussian as the 3D correspondence of a query keypoint. A Gaussian, however, represents a volumetric region rather than a single scene point: its center summarizes the region's spatial distribution but generally differs from the point observed by the keypoint, as shown in Fig. 1a. Consequently, even when the correct Gaussian is identified, center-based assignment yields an inaccurate 2D-3D correspondence and introduces geometric error into PnP. Subdivision reduces this discrepancy by shrinking each primitive, but still retains center-based assignment, only at a finer scale. This motivates a landmark representation that retains the Gaussian's regional extent while determining the corresponding 3D position from the query observation, as illustrated in Fig. 1b.

To this end, we propose ReCoLoc, a *region-to-coordinate* framework for 3DGS-based visual localization. Our key idea is to treat each Gaussian as a retrievable 3D region and endow it with a function that predicts a query-specific 3D position. This separates identifying the relevant Gaussian from determining the point needed for pose estimation. Specifically, to associate 2D keypoints reliably with 3D regions, we introduce *contribution-weighted identity encoding*, which aggregates multiview descriptors according to the Gaussian's rendering contribution at their observed pixels. The resulting identity descriptor favors observations strongly associated with that region, facilitating retrieval across viewpoints. To determine the corresponding point within the retrieved 3D region, we develop *query-conditioned coordinate decoding*. By associating individual keypoint descriptors with local coordinates derived from their camera rays, we fit a coordinate function that maps local appearance to position within the region. The query descriptor thus determines the 3D point returned by the matched Gaussian for PnP-based pose estimation. Correspondence accuracy depends on both region retrieval and coordinate prediction. We therefore devise *correspondence-aware landmark selection*, which scores both steps on a separate subset of mapping views and retains spatially distributed regions that produce reliable 2D-3D matches. Equipping landmarks with coordinate functions also increases their storage requirements. To reduce this cost, *shared-subspace map compression* represents the identity descriptors and coordinate functions in a common low-dimensional basis that accounts for both retrieval and coordinate prediction. The resulting Tiny-ReCoLoc reduces direct-matching map storage by 85.1% with little loss of localization accuracy.

Thorough experimentation across three public benchmarks affirms ReCoLoc's promising 2D-3D matching capability and its benefits for accurate visual localization. By treating region retrieval and point prediction as distinct steps when forming 2D-3D correspondences from Gaussian landmarks, ReCoLoc opens up exciting prospects for 3DGS-based visual localization.

2 Related Work

Structure-based localization. Structure-based methods estimate camera poses from correspondences between 2D query features and reconstructed 3D points. Research in this direction improves both map search and feature matching. To search large point clouds efficiently, Active Search (Sattler et al., 2012) combines 2D-to-3D matching with an active 3D-to-2D search around established

correspondences. HLoc (Sarlin et al., 2019) adopts a hierarchical pipeline, using image retrieval to restrict local feature matching to relevant parts of the scene. Alongside these search strategies, learned features and matchers improve the reliability of correspondences. SuperPoint (DeTone et al., 2018) jointly learns keypoint detection and description through self-supervision, while SuperGlue (Sarlin et al., 2020) models feature interactions with attention and matches them through optimal transport.

Scene coordinate regression-based localization. Scene coordinate regression establishes 2D-3D correspondences by predicting scene coordinates from image observations. Shotton et al. (2013) introduce this formulation using regression forests for RGB-D relocalization. DSAC* (Brachmann & Rother, 2022) combines neural coordinate prediction with differentiable robust pose estimation, enabling end-to-end training with RGB images and camera poses. More recent work improves the efficiency of learning and storing scene-specific representations. For instance, ACE (Brachmann et al., 2023) accelerates mapping by training a coordinate head on shuffled features from a pretrained encoder, using a reprojection-loss curriculum. NeuMap (Tang et al., 2023) addresses map compactness by encoding scene information in spatially organized latent codes and decoding coordinates with a shared Transformer. Its code-pruning mechanism controls the storage required by the representation.

NeRF-based localization. Neural radiance fields enable localization using synthesized images, features, and geometry. For pose regression and refinement, DFNet (Chen et al., 2022) uses exposure-adaptive view synthesis and direct feature alignment to improve pose regression. NeFeS (Chen et al., 2024) learns a dense feature field and refines initial poses by minimizing the discrepancy between synthesized and query features. Another line uses radiance fields to establish explicit correspondences for geometric pose estimation. CROSSFIRE (Moreau et al., 2023) jointly learns an image encoder and a scene-specific descriptor field, then combines rendered descriptors and depth to construct 2D-3D matches near a pose prior. PNeRFLoc (Zhao et al., 2024) incorporates localization features into a point-based radiance field through feature adaptation, then uses point matching for initial localization and photometric warping for refinement. NeRFMatch (Zhou et al., 2024) instead learns to match query features to the intermediate features of a pretrained NeRF, obtaining correspondences with rendered 3D points for initial pose estimation and iterative refinement.

3DGS-based localization. The explicit Gaussian primitives in 3DGS (Kerbl et al., 2023) can be equipped with descriptors for direct 2D-3D matching. GSplatLoc (Sidorov et al., 2025) distills local descriptors into Gaussians, while STDLoc (Huang et al., 2025) improves sparse matching through landmark sampling and a scene-specific detector. ULF-Loc (Gu et al., 2026) selects landmarks by keypoint consensus and removes alpha-blending bias through geometry-weighted multiview feature fusion. SplitGS-Loc (Lee et al., 2026) uses rendering contributions for landmark selection and feature aggregation, and splits Gaussians to reduce many-to-one pixel-Gaussian ambiguity. These methods improve landmark features and geometry but still assign a single coordinate to each matched Gaussian. ReCoLoc revisits this landmark representation by retaining the Gaussian as a 3D region and predicting the correspondence coordinate from the query descriptor, making point determination an explicit part of 2D-3D matching. Efficient rendering also enables pose refinement from RGB or feature renderings. GSplatLoc uses photometric warping, while GS-CPR (Liu et al., 2025) matches the query with a view rendered at the initial pose and lifts the matches into 3D using rendered depth. STDLoc performs dense matching against rendered feature maps, whereas ULF-Loc matches the query with a rendered RGB image and filters coarse matches by local geometric consistency.

3 METHODOLOGY

ReCoLoc constructs Gaussian landmarks for query-dependent point correspondences from posed mapping images and an RGB 3DGS reconstruction. Let $\mathcal{G}$ be the set of Gaussian indices and $\mathcal{V}$ the mapping views with known camera intrinsics and poses. A pretrained local feature encoder ϕ extracts keypoints and computes unit-length D-dimensional descriptors. As shown in Fig. 2, we first collect multiview observations to encode regional identities and fit coordinate decoders, then select landmarks by their correspondence quality. For the compact variant, a shared subspace compresses the map for storage-efficient localization.

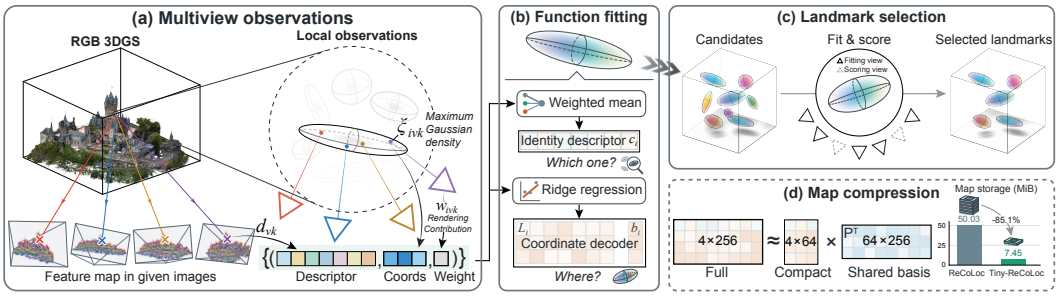

Figure 2: **Overview of ReCoLoc.** (a) Each mapping keypoint is associated with Gaussians through rendering contributions w_{ivk} and provides its descriptor $\boldsymbol{d}_{vk}$ and a local coordinate $\boldsymbol{\xi}_{ivk}$ at the maximum Gaussian density along its ray. (b) A contribution-weighted mean gives the identity descriptor $\boldsymbol{c}_i$ that retrieves the Gaussian, and ridge regression fits the decoder $(\boldsymbol{L}_i, \boldsymbol{b}_i)$ that predicts the 3D position within it. (c) Candidates are selected by their correspondence quality on scoring views. (d) Tiny-ReCoLoc projects descriptors and decoders onto a shared basis $\boldsymbol{P}$.

3.1 Query-conditioned Gaussian landmarks

To recover a point after identifying its Gaussian region, we associate each Gaussian indexed by i with a unit identity descriptor $\boldsymbol{c}_i \in \mathbb{R}^D$ and a coordinate decoder $\mathcal{F}_i : \mathbb{R}^D \to \mathbb{R}^3$. Given a query keypoint $\boldsymbol{u}_j \in \mathbb{R}^2$ with unit descriptor $\boldsymbol{q}_j \in \mathbb{R}^D$, where j indexes query keypoints, we retrieve the Gaussian with the highest descriptor similarity and apply its decoder to predict the 3D coordinate. For the selected index set $\mathcal{L} \subseteq \mathcal{G}$ of N landmarks, the correspondence is obtained as

$$i_j = \arg\max_{i \in \mathcal{L}} \boldsymbol{q}_j^\top \boldsymbol{c}_i, \qquad \boldsymbol{x}_j = \mathcal{F}_{i_j}(\boldsymbol{q}_j). \tag{1}$$

To account for the shape and scale of each region, we parameterize its decoder in the Gaussian's local frame. For center $\boldsymbol{\mu}_i \in \mathbb{R}^3$ and covariance $\boldsymbol{\Sigma}_i = \boldsymbol{H}_i \boldsymbol{H}_i^\top$, the matrix $\boldsymbol{H}_i = \boldsymbol{R}_i \operatorname{diag}(\boldsymbol{s}_i)$ contains the scaled principal axes: $\boldsymbol{R}_i \in \mathrm{SO}(3)$ is the Gaussian rotation and $\boldsymbol{s}_i \in \mathbb{R}^3_{>0}$ contains its axis scales. Here $\mathrm{SO}(3)$ denotes the three-dimensional rotation group. A local coordinate $\boldsymbol{\xi}$ maps to the world point $\boldsymbol{\mu}_i + \boldsymbol{H}_i \boldsymbol{\xi}$. We restrict local coordinates to $\|\boldsymbol{\xi}\|_2 \le \rho$, defining an ellipsoidal region around the Gaussian center, with radius $\rho > 0$ in the local frame.

3.2 Multiview landmark construction

Ray-guided observation construction. To describe a region and resolve positions within it, we collect keypoint observations across the mapping views. In view v, a local feature extractor ϕ extracts up to M keypoints $\boldsymbol{u}_{vk}$, indexed by k, with unit descriptors $\boldsymbol{d}_{vk} \in \mathbb{R}^D$. We use SuperPoint (DeTone et al., 2018) by default. To associate these image features with Gaussian i, we compute its front-to-back rendering contribution at each keypoint pixel,

$$w_{ivk} = \alpha_{iv}(\boldsymbol{u}_{vk}) \prod_{\ell \prec_v i} \big(1 - \alpha_{\ell v}(\boldsymbol{u}_{vk})\big), \tag{2}$$

where α_{iv} is the opacity-weighted projected Gaussian in view v, and $\ell \prec_v i$ denotes Gaussians preceding i in the front-to-back rendering order.

We derive a local coordinate target for each association by back-projecting the keypoint into a camera ray and finding the point of maximum Gaussian density along it. Let $\boldsymbol{C}_v$ be the camera center and $\boldsymbol{r}_{vk}$ the unit world-space bearing of keypoint k. For ray distance t, the ray in the Gaussian frame is $\boldsymbol{o} + t\boldsymbol{h}$, where $\boldsymbol{o} = \boldsymbol{H}_i^{-1}(\boldsymbol{C}_v - \boldsymbol{\mu}_i)$ and $\boldsymbol{h} = \boldsymbol{H}_i^{-1}\boldsymbol{r}_{vk}$. The target is

$$t^*_{ivk} = -\frac{\boldsymbol{h}^\top \boldsymbol{o}}{\boldsymbol{h}^\top \boldsymbol{h}}, \qquad \boldsymbol{\xi}_{ivk} = \boldsymbol{o} + t^*_{ivk}\boldsymbol{h}. \tag{3}$$

This construction places the target on the observed ray while accounting for the Gaussian's anisotropic geometry. We retain an association when its contribution satisfies $w_{ivk} \ge \tau$, where τ is the association threshold, its depth is positive ($t^*_{ivk} > 0$), and $\|\boldsymbol{\xi}_{ivk}\|_2 \le \rho$. We denote the retained index pairs by $\mathcal{O}_i$ and store $(\boldsymbol{d}_{vk}, \boldsymbol{\xi}_{ivk}, w_{ivk})$ for each $(v, k) \in \mathcal{O}_i$.

Contribution-weighted identity encoding. The identity descriptor should allow observations of the same region to retrieve a common landmark across views. To obtain this regional representation, we aggregate the descriptors in $\mathcal{O}_i$, using their rendering contributions to weight each observation,

$$\bar{w}_{ivk} = \frac{w_{ivk}}{\sum_{(v',k')\in\mathcal{O}_i} w_{iv'k'}}, \qquad \bar{\boldsymbol{d}}_i = \sum_{(v,k)\in\mathcal{O}_i} \bar{w}_{ivk}\boldsymbol{d}_{vk}, \qquad \boldsymbol{c}_i = \frac{\bar{\boldsymbol{d}}_i}{\|\bar{\boldsymbol{d}}_i\|_2}. \tag{4}$$

Normalizing the weighted mean yields $\boldsymbol{c}_i$, which is used for cosine-similarity retrieval in Eq. (1).

Query-conditioned coordinate decoding. The aggregated descriptor identifies the region, but the individual observations carry the appearance variations needed to distinguish positions within it. To preserve this information, we fit an affine decoder to the descriptor-coordinate pairs in $\mathcal{O}_i$,

$$(\boldsymbol{L}_i, \boldsymbol{b}_i) = \arg\min_{\boldsymbol{L},\boldsymbol{b}} \sum_{(v,k)\in\mathcal{O}_i} \bar{w}_{ivk}\|\boldsymbol{\xi}_{ivk} - \boldsymbol{L}\boldsymbol{d}_{vk} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{L}\|_F^2, \quad \boldsymbol{L}_i \in \mathbb{R}^{3\times D}. \tag{5}$$

Here $\lambda > 0$ regularizes the descriptor-dependent coefficients $\boldsymbol{L}_i$, while the intercept $\boldsymbol{b}_i \in \mathbb{R}^3$ is unpenalized; $\|\cdot\|_F$ denotes the Frobenius norm. The weighted ridge problem has a closed-form solution (Appendix A.5). Writing the prediction for a query descriptor $\boldsymbol{q}$ around the weighted target mean $\bar{\boldsymbol{\xi}}_i = \sum_{(v,k)\in\mathcal{O}_i} \bar{w}_{ivk}\boldsymbol{\xi}_{ivk}$ gives

$$\boldsymbol{L}_i\boldsymbol{q} + \boldsymbol{b}_i = \bar{\boldsymbol{\xi}}_i + \boldsymbol{L}_i(\boldsymbol{q} - \bar{\boldsymbol{d}}_i), \qquad \mathcal{F}_i(\boldsymbol{q}) = \boldsymbol{\mu}_i + \boldsymbol{H}_i\Pi_\rho(\boldsymbol{L}_i\boldsymbol{q} + \boldsymbol{b}_i), \tag{6}$$

where $\Pi_\rho(\boldsymbol{y}) = \boldsymbol{y}/\max(1, \|\boldsymbol{y}\|_2/\rho)$ projects any $\boldsymbol{y} \in \mathbb{R}^3$ onto the local ball of radius ρ. The query-dependent term $\boldsymbol{L}_i(\boldsymbol{q} - \bar{\boldsymbol{d}}_i)$ predicts a displacement from the regional mean coordinate.

3.3 Correspondence-aware landmark selection

We select landmarks by jointly evaluating region retrieval and coordinate prediction on a separate set of mapping views. We partition the mapping views $\mathcal{V}$ into disjoint fitting views $\mathcal{V}_{\text{fit}}$ and scoring views $\mathcal{V}_{\text{score}}$. Among Gaussians with enough observations across mapping views, we sample a spatially distributed candidate pool $\mathcal{C}$ of at most C regions and fit their descriptors and decoders on $\mathcal{V}_{\text{fit}}$. Appendix A.2 gives the observation-count thresholds and sampling procedure.

For each scoring view, we retrieve one candidate per keypoint and predict its coordinate, following Eq. (1). Matches whose descriptor similarity exceeds the acceptance threshold η_{sim} are retained. For candidate i in view v, let $\mathcal{A}_{iv}$ contain its associated keypoints from $\mathcal{O}_i$, and let $\mathcal{B}_{iv}$ contain the keypoints that retrieve it. We weight each keypoint in $\mathcal{A}_{iv} \cap \mathcal{B}_{iv}$, which is both associated with the Gaussian and retrieved by its descriptor, by its coordinate accuracy, measured through the reprojection error $e_{ivk} = \|\pi_v(\mathcal{F}_i(\boldsymbol{d}_{vk})) - \boldsymbol{u}_{vk}\|_2$ where π_v projects a world point to pixels using the known intrinsics and pose of mapping view v. For a nonempty region-view pair, the score is

$$g_{iv} = \sum_{k\in\mathcal{A}_{iv}\cap\mathcal{B}_{iv}} \exp\left(-\frac{e_{ivk}^2}{2\sigma^2}\right), \qquad r_{iv} = \frac{2g_{iv}}{|\mathcal{A}_{iv}| + |\mathcal{B}_{iv}|}. \tag{7}$$

Here $\sigma > 0$, measured in pixels, controls the reprojection tolerance. Predictions with invalid projection or nonpositive depth contribute zero to g_{iv}. The denominator accounts for missed associations and incorrect retrievals, while the numerator rewards correctly retrieved keypoints whose predicted coordinates reproject accurately. Consequently, a region receives a high score when its descriptor and decoder together produce reliable 2D-3D correspondences.

To stabilize scores for regions observed in few scoring views, we smooth the per-region average with a shared prior. Let $\mathcal{V}_i$ contain the views with $|\mathcal{A}_{iv}| + |\mathcal{B}_{iv}| > 0$, and let r_0 be the mean score over all observed region-view pairs. We use

$$Q_i = \frac{\sum_{v\in\mathcal{V}_i} r_{iv} + \kappa r_0}{|\mathcal{V}_i| + \kappa}, \tag{8}$$

where $\kappa \geq 0$ controls the prior strength. To preserve scene coverage, we select N landmarks by ranking Q_i within spatial neighborhoods, following STDLoc (Huang et al., 2025). We then refit the selected descriptors and decoders on all mapping views. Appendix A.2 gives the complete algorithm.

3.4 SHARED-SUBSPACE MAP COMPRESSION

The identity descriptors and decoder matrices account for most of the map storage. To obtain a compact representation for both retrieval and coordinate prediction, we exploit their common descriptor space and project them onto a shared low-dimensional basis. Let $\boldsymbol{C} \in \mathbb{R}^{N \times D}$ stack the identity descriptors and $\boldsymbol{B} \in \mathbb{R}^{3N \times D}$ stack the matrices $\boldsymbol{L}_i$. We choose a shared basis $\boldsymbol{P} \in \mathbb{R}^{D \times d_c}$ with orthonormal columns, where $d_c < D$ is the compressed dimension, from the leading eigenvectors of the joint scatter matrix

$$\boldsymbol{G} = \frac{\boldsymbol{C}^\top \boldsymbol{C}}{\|\boldsymbol{C}\|_F^2} + \frac{\boldsymbol{B}^\top \boldsymbol{B}}{\|\boldsymbol{B}\|_F^2}. \tag{9}$$

Including both components preserves directions used to identify regions and to predict coordinates; the normalization balances their relative reconstruction errors. A projected query $\boldsymbol{z} = \boldsymbol{P}^\top \boldsymbol{q}$ is matched against $\boldsymbol{a}_i = \boldsymbol{P}^\top \boldsymbol{c}_i$ using $\boldsymbol{z}^\top \boldsymbol{a}_i$, then mapped to local coordinates with $\boldsymbol{A}_i = \boldsymbol{L}_i \boldsymbol{P}$.

To further reduce storage, we quantize each row of $\boldsymbol{A}_i$ with its own scale. Let $\widehat{\boldsymbol{A}}_i$ be the dequantized matrix and $\widetilde{\boldsymbol{b}}_i$ the corrected intercept that preserves the full prediction at $\bar{\boldsymbol{d}}_i$. The compact coordinate function is

$$\widetilde{\mathcal{F}}_i(\boldsymbol{q}) = \boldsymbol{\mu}_i + \boldsymbol{H}_i \Pi_\rho(\widehat{\boldsymbol{A}}_i \boldsymbol{z} + \widetilde{\boldsymbol{b}}_i). \tag{10}$$

Appendix A.3 specifies the quantization and mean-preserving correction.

3.5 CAMERA RE-LOCALIZATION

Direct 2D-3D matching for pose estimation. Given a query image, we extract local features, retrieve a Gaussian region for each keypoint, and decode its 3D coordinate using Eq. (1). Correspondences above the descriptor similarity threshold η_{sim} are passed to PoseLib RANSAC-PnP (Larsson & contributors, 2020) to estimate the camera pose. The compact variant projects query descriptors into the shared subspace and follows the same procedure.

Pose refinement. To improve the initial estimate, we use the rendering capability of the same RGB 3DGS reconstruction to establish correspondences near the estimated viewpoint (Liu et al., 2025). At each iteration, we render an RGB image and its expected depth, and match the query to the rendered image using the semi-dense matcher EDM (Li et al., 2025). Back-projecting the rendered match locations with their depths gives 3D points for the matched query pixels. RANSAC-PnP estimates an updated pose, which provides the viewpoint for the next refinement iteration.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate ReCoLoc on Cambridge Landmarks (Kendall et al., 2015) for outdoor localization under viewpoint and illumination changes, and on 7-Scenes (Shotton et al., 2013) and 12-Scenes (Valentin et al., 2016) for indoor localization in texture-poor and repetitive environments. For 7-Scenes and 12-Scenes, we use the SfM pseudo ground-truth poses provided by Brachmann et al. (2021). For Cambridge Landmarks, we use the SfM reference poses of Kendall et al. (2015).

Metrics. We report the median translation (cm) and rotation (°) errors for each scene and their averages across scenes. Localization recall measures the percentage of query images whose translation and rotation errors are both below the specified thresholds. Strict recall uses 15 cm/5° on Cambridge and 1 cm/1° on the indoor datasets. To assess correspondence quality, we also report correspondence accuracy, the fraction of 2D-3D matches with positive depth and a reprojection error below 5 pixels under the ground-truth pose.

Baselines. We compare ReCoLoc with four families of visual localizers, including structure-based methods: AS (SIFT) (Sattler et al., 2012), HLoc (SP+SG) (Sarlin et al., 2019; 2020), and DVLAD+R2D2 (Brachmann et al., 2021); scene coordinate regression-based methods: DSAC* (Brachmann & Rother, 2022), ACE (Brachmann et al., 2023), and NeuMap (Tang et al., 2023); NeRF-based methods: DFNet+NeFeS$_{50}$ (Chen et al., 2022; 2024), CROSSFIRE (Moreau

et al., 2023), PNeRFLoc (Zhao et al., 2024), and NeRFMatch (Zhou et al., 2024); and 3DGS-based methods: GSplatLoc (Sidorov et al., 2025), ACE+GS-CPR (Liu et al., 2025), STDLoc (Huang et al., 2025), ULF-Loc (Gu et al., 2026), and SplitGS-Loc (Lee et al., 2026).

Implementation details. We train an RGB 3DGS reconstruction for each scene and freeze its geometry before landmark construction; training details are given in Appendix A.1. We use SuperPoint descriptors with $D = 256$, extracting up to $M = 2{,}048$ keypoints per image and retaining $N = 20{,}000$ Gaussian landmarks per scene. We set the contribution threshold to $\tau = 0.05$, the local coordinate radius to $\rho = 3$, the ridge coefficient to $\lambda = 10^{-3}$, the reprojection scale to $\sigma = 5$ pixels, and the score prior strength to $\kappa = 4$. Appendix B.1 analyzes their sensitivity. Tiny-ReCoLoc uses a shared dimension of $d_c = 64$, with 8-bit coordinate coefficients and 16-bit floating-point descriptors and auxiliary parameters. We estimate camera poses using PoseLib RANSAC-PnP and apply refinement for two iterations on 7-Scenes and one on 12-Scenes, rendering at 640×480 resolution. All experiments run on an NVIDIA RTX 4090 GPU.

4.2 Visual Localization Benchmark

Results on Cambridge Landmarks. Table 1 compares localization results on the five outdoor scenes. Without pose refinement, ReCoLoc achieves an average scene-median error of 6.3 cm/0.11°, reducing the translation error of SplitGS-Loc (8.7 cm/0.15°) by 27.6%. It also surpasses the refined results of STDLoc and ULF-Loc (8.7 cm/0.13° and 8.7 cm/0.14°) and, as reported in Table 4, raises strict recall to 81.8%, compared with 72.3% for refined STDLoc. These results show the benefit of recovering query-dependent points within the retrieved Gaussian regions for outdoor localization.

Table 1: **Localization results on Cambridge Landmarks.** Median translation and rotation errors (cm/°) are reported. The last column reports the average across scenes. PR denotes pose refinement for 3DGS-based methods. The best, second-best, and third-best results are highlighted.

	Method	Great Court	King's College	Old Hospital	Shop Facade	St Mary's	Avg.↓ [cm/°]
Str	AS (SIFT)	24/0.13	13/0.22	20/0.36	4/0.21	8/0.25	13.8/0.23
	HLoc (SP+SG)	17.7/0.11	11.0/0.20	15.1/0.31	4.2/0.20	7.0/0.22	11.0/0.21
SCR	DSAC*	33.0/0.21	17.9/0.31	21.1/0.40	5.2/0.24	15.4/0.51	18.5/0.33
	ACE	27.9/0.14	18.6/0.33	25.7/0.51	5.1/0.26	9.5/0.33	17.4/0.31
	NeuMap	6/0.10	14/0.19	19/0.36	6/0.25	17/0.53	12.4/0.29
NeRF	DFNet+NeFeS$_{50}$	-	37/0.54	52/0.88	15/0.53	37/1.14	35.3/0.77
	CROSSFIRE	-	47/0.7	43/0.7	20/1.2	39/1.4	37.3/1.00
	PNeRFLoc	81/0.25	24/0.29	28/0.37	6/0.27	40/0.55	35.8/0.35
	NeRFMatch	19.6/0.09	12.5/0.23	20.9/0.38	8.4/0.40	10.9/0.35	14.5/0.29
3DGS	GSplatLoc	-	31/0.49	16/0.68	4/0.34	14/0.42	16/0.48
	ACE+GS-CPR	-	20/0.29	21/0.40	5/0.24	13/0.40	15/0.33
	STDLoc	12.9/0.08	15.3/0.18	18.9/0.35	3.6/0.17	5.2/0.19	11.2/0.19
	STDLoc (PR)	11.5/0.06	14.8/0.16	11.3/0.21	2.5/0.12	3.6/0.12	8.7/0.13
	ULF-Loc	18.7/0.10	16.8/0.18	16.6/0.30	3.8/0.21	8.7/0.29	12.9/0.22
	ULF-Loc (PR)	9.6/0.05	17.4/0.19	10.1/0.19	2.7/0.14	3.8/0.12	8.7/0.14
	SplitGS-Loc	9.3/0.06	14.1/0.17	13.8/0.29	2.5/0.12	3.8/0.13	8.7/0.15
	Tiny-ReCoLoc	8.6/0.04	7.1/0.10	9.8/0.21	2.1/0.11	3.4/0.12	6.2/0.11
	ReCoLoc	8.6/0.04	7.1/0.09	10.2/0.19	2.1/0.11	3.4/0.12	6.3/0.11

Results on 7-Scenes. As shown in Table 2, ReCoLoc achieves an average error of 0.78 cm/0.22° through direct matching, compared with 1.32 cm/0.33° for STDLoc, 1.11 cm/0.31° for ULF-Loc, and 1.0 cm/0.26° for SplitGS-Loc; the per-scene errors on both datasets are shown in Fig. 1c. Refinement further reduces the error to 0.55 cm/0.18°, compared with 0.6 cm/0.19° for refined SplitGS-Loc. As reported in Table 4, the improvement is also reflected in strict recall, which rises from 68.3% to 80.3%, compared with 73.3% for refined ULF-Loc.

Results on 12-Scenes. As shown in Table 3, ReCoLoc achieves an average error of 0.33 cm/0.14° before refinement, matching refined STDLoc. One refinement iteration reduces the translation error to 0.31 cm while retaining a rotation error of 0.14°, and raises the strict recall in Table 4 from 94.9% to 96.7%, compared with 93.2% for refined STDLoc. Appendix D.1 provides per-scene results.

Compact representation. Tiny-ReCoLoc projects the identity descriptors and coordinate functions into a shared 64-dimensional space. It requires 7.45 MiB per scene, compared with 50.03 MiB for the full map, an 85.1% reduction. Across Cambridge, 7-Scenes, and 12-Scenes, strict recall

Table 2: **Localization results on 7-Scenes.**

	Method	Chess	Fire	Heads	Office	Pumpkin	Red Kitchen	Stairs	Avg.↓ [cm/°]
Str	AS (SIFT)	3/0.87	2/1.01	1/0.82	4/1.15	7/1.69	5/1.72	4/1.01	3.71/1.18
Str	HLoc (SP+SG)	2.39/0.84	2.29/0.91	1.13/0.77	3.14/0.92	4.92/1.30	4.22/1.39	5.05/1.41	3.31/1.08
Str	DVLAD+R2D2	2.56/0.88	2.21/0.86	0.98/0.75	3.48/1.00	4.79/1.28	4.21/1.44	4.60/1.27	3.26/1.07
SCR	DSAC*	0.50/0.17	0.78/0.29	0.50/0.34	1.19/0.35	1.19/0.29	0.72/0.21	2.65/0.78	1.07/0.35
SCR	ACE	0.55/0.18	0.83/0.33	0.53/0.33	1.05/0.29	1.06/0.22	0.77/0.21	2.89/0.81	1.10/0.34
SCR	NeuMap	2/0.81	3/1.11	2/1.17	3/0.98	4/1.11	4/1.33	4/1.12	3.14/1.09
NeRF	DFNet+NeFeS$_{50}$	2/0.57	2/0.74	2/1.28	2/0.56	2/0.55	2/0.57	5/1.28	2.43/0.79
NeRF	CROSSFIRE	1/0.4	5/1.9	3/2.3	5/1.6	3/0.8	2/0.8	12/1.9	4.43/1.39
NeRF	PNeRFLoc	2/0.80	2/0.88	1/0.83	3/1.05	6/1.51	5/1.54	32/5.73	7.29/1.76
NeRF	NeRFMatch	0.95/0.30	1.11/0.41	1.34/0.92	3.09/0.87	2.21/0.60	1.03/0.28	9.26/1.74	2.71/0.73
3DGS	GSplatLoc	0.39/0.13	0.91/0.29	0.94/0.50	1.41/0.32	1.41/0.26	1.32/0.29	3.44/0.82	1.40/0.37
3DGS	ACE+GS-CPR	0.5/0.15	0.6/0.25	0.4/0.28	0.9/0.26	1.0/0.23	0.7/0.17	1.4/0.42	0.8/0.25
3DGS	STDLoc	0.98/0.22	0.93/0.31	0.63/0.31	1.36/0.29	1.27/0.25	1.43/0.24	2.64/0.70	1.32/0.33
3DGS	STDLoc (PR)	0.42/0.13	0.52/0.21	0.41/0.26	0.74/0.21	0.92/0.24	0.59/0.14	1.20/0.35	0.68/0.22
3DGS	ULF-Loc	0.67/0.17	0.88/0.34	0.51/0.31	1.21/0.28	1.12/0.21	1.23/0.22	2.14/0.62	1.11/0.31
3DGS	ULF-Loc (PR)	0.42/0.11	0.44/0.19	0.37/0.23	0.75/0.20	0.99/0.21	0.64/0.13	1.34/0.35	0.71/0.20
3DGS	SplitGS-Loc	0.7/0.16	0.7/0.29	0.6/0.26	1.2/0.25	1.4/0.25	1.0/0.18	1.6/0.47	1.0/0.26
3DGS	SplitGS-Loc (PR)	0.3/0.11	0.4/0.18	0.4/0.27	0.6/0.16	0.9/0.20	0.4/0.12	0.9/0.28	0.6/0.19
3DGS	**Tiny-ReCoLoc**	0.46/0.12	0.59/0.25	0.41/0.24	0.88/0.21	1.05/0.21	0.76/0.15	1.34/0.40	0.79/0.22
3DGS	**ReCoLoc**	0.46/0.12	0.60/0.25	0.41/0.24	0.87/0.21	1.05/0.20	0.76/0.15	1.34/0.40	0.78/0.22
3DGS	**ReCoLoc (PR)**	0.27/0.08	0.44/0.18	0.29/0.20	0.68/0.19	0.77/0.18	0.48/0.12	0.94/0.27	0.55/0.18

Table 3: **Localization results on 12-Scenes.**

	Method	Avg.↓ [cm/°]
SCR	DSAC*	0.5/0.25
SCR	ACE	0.7/0.26
NeRF	PNeRFLoc	2.8/1.08
3DGS	GSplatLoc	1.4/0.26
3DGS	ACE+GS-CPR	0.5/0.21
3DGS	STDLoc	0.56/0.25
3DGS	STDLoc (PR)	0.33/0.14
3DGS	ULF-Loc	0.50/0.21
3DGS	ULF-Loc (PR)	0.36/0.16
3DGS	**Tiny-ReCoLoc**	0.33/0.14
3DGS	**ReCoLoc**	0.33/0.14
3DGS	**ReCoLoc (PR)**	0.31/0.14

Table 4: **Localization recall.** Mean scene recall (%) ↑ at translation/rotation thresholds [cm/°].

Method	7-Scenes			12-Scenes			Cambridge	
	[5/5]	[2/2]	[1/1]	[5/5]	[2/2]	[1/1]	[50/5]	[15/5]
DSAC*	97.8	80.7	-	99.8	96.7	-	-	-
ACE	97.1	83.3	56.3	100	97.2	75.0	78.7	43.1
GSplatLoc	78.0	-	34.6	91.9	70.3	45.4	-	-
ACE+GS-CPR	99.9	92.9	66	100	98.7	87.1	84.6	56.8
STDLoc	96.9	79.5	40.1	98.7	96.2	82.4	92.4	65.5
STDLoc (PR)	99.2	93.5	73.0	98.3	97.8	93.2	94.1	72.3
ULF-Loc	98.0	84.4	50.0	98.8	97.5	87.6	88.2	58.6
ULF-Loc (PR)	98.5	91.1	73.3	97.3	96.7	92.2	92.8	69.1
Tiny-ReCoLoc	99.1	92.1	68.3	99.7	99.2	94.8	95.9	81.0
ReCoLoc	99.1	92.2	68.3	99.7	99.2	94.9	96.0	81.8
ReCoLoc (PR)	99.4	95.2	80.3	99.9	99.5	96.7	-	-

decreases only by 0.75, 0.07, and 0.10 percentage points, respectively. These results show that the shared representation substantially reduces storage while preserving localization accuracy. Tables 1-4 report its results without refinement; Appendix B.3 examines the storage-accuracy trade-off across descriptor dimensions.

4.3 Understanding ReCoLoc

From region retrieval to point coordinates. To isolate the effect of query-conditioned coordinate prediction, we fix Gaussian retrieval and compare three coordinate outputs in Table 5: the Gaussian center, the contribution-weighted mean of its fitting targets, and the decoder prediction. We evaluate both joint-selected maps and observation-selected maps, whose landmarks are ranked by effective observation count. Within each map, the three coordinate outputs use identical Gaussian matches.

Table 5: **Coordinate prediction from the same Gaussian matches.** Error is the scene-averaged median (cm/°). R is recall (%) at 15 cm/5° on Cambridge and 1 cm/1° indoors; C is correspondence accuracy (%) at 5 pixels.

Dataset	Map selection	Center			Weighted mean			Decoder		
		Error↓	R ↑	C ↑	Error↓	R ↑	C ↑	Error↓	R ↑	C ↑
Cambridge	Observation-selected	11.0/0.20	67.26	8.09	8.2/0.14	75.16	13.80	**6.6/0.12**	**80.37**	**19.52**
	Joint-selected	9.1/0.17	73.68	10.09	7.8/0.13	75.55	17.62	**6.3/0.11**	**81.78**	**22.82**
7-Scenes	Observation-selected	1.52/0.33	23.94	23.84	1.03/0.27	52.31	31.92	**0.82/0.23**	**65.18**	**43.70**
	Joint-selected	1.42/0.32	28.30	26.45	0.95/0.25	58.09	35.02	**0.78/0.22**	**68.33**	**45.65**
12-Scenes	Observation-selected	0.65/0.29	75.62	10.68	0.42/0.18	90.41	18.11	**0.35/0.15**	**93.21**	**29.32**
	Joint-selected	0.54/0.24	83.85	13.93	0.38/0.17	93.03	22.27	**0.33/0.14**	**94.94**	**32.57**

On the full method's joint-selected maps, the decoder consistently improves both strict pose recall and correspondence accuracy over the two query-independent coordinate baselines across all three datasets. Relative to the weighted mean, strict recall increases by 6.23, 10.24, and 1.91 percentage points on Cambridge, 7-Scenes, and 12-Scenes, respectively. Similar improvements are observed on observation-selected maps. These consistent gains across both map-selection strategies demonstrate the benefit of query-conditioned coordinate prediction.

Retrieving the Gaussian region. The decoder relies on retrieving the correct region. To evaluate retrieval, we compare our identity descriptor with the feature-registration rules of ULF-Loc and SplitGS-Loc on Cambridge. We construct descriptors on the same Gaussian geometry using fitting views, then measure Top-1 association accuracy on the scoring views. A Gaussian contributing to the query pixel is a positive association; queries with at least one such association in the common valid map are evaluated. As shown in Table 6, our descriptor achieves 21.26% average accuracy, compared with 9.75% for ULF-Loc registration and 15.39% for SplitGS-Loc registration.

Table 6: **Identity retrieval on Cambridge.** Top-1 Gaussian association accuracy (%) on the scoring views.

Registration rule	Great Court	King's	Hospital	Shop	St Mary's	Avg.↑
ULF-Loc	15.79	12.33	7.22	7.54	5.87	9.75
SplitGS-Loc	20.80	19.60	9.03	13.71	13.78	15.39
ReCoLoc	**25.35**	**30.89**	**12.21**	**18.19**	**19.66**	**21.26**

Multiple keypoints within one Gaussian. To examine how the decoder handles keypoints that retrieve the same region, we group correspondences by Gaussian identity within each query image. A single identity is retrieved once; a shared identity is retrieved by two or more keypoints. Shared identities account for 32.5%, 48.4%, and 54.6% of correspondences on Cambridge, 7-Scenes, and 12-Scenes, respectively. Table 7 reports correspondence accuracy in both groups and, for keypoint pairs sharing an identity, the fraction for which both correspondences are correct.

The decoder improves correspondence accuracy for both single and shared identities. For example, on 7-Scenes, accuracy increases from 31.38% to 48.35% for single identities and from 21.18% to 42.76% for shared identities. Among pairs assigned to the same Gaussian, the fraction with both correspondences correct increases from 4.82% to 20.97%. That is to say, a query-conditioned function can place different keypoints at different locations within a shared region, while also improving point coordinates when the Gaussian is retrieved only once.

Table 7: **Correspondence accuracy for single and shared Gaussian identities.** Single and Shared give the percentage of correct correspondences; Both gives the percentage of shared-identity pairs with two correct correspondences.

Output	Single	Shared	Both
Center	12.73	4.61	0.04
Mean	22.16	8.23	0.08
ReCoLoc	**27.80**	**12.50**	**1.18**

(a) Cambridge. 32.5% shared.

Output	Single	Shared	Both
Center	31.38	21.18	4.82
Mean	40.72	28.93	7.44
ReCoLoc	**48.35**	**42.76**	**20.97**

(b) 7-Scenes. 48.4% shared.

Output	Single	Shared	Both
Center	15.76	12.42	1.68
Mean	25.22	19.82	3.09
ReCoLoc	**33.50**	**31.80**	**10.71**

(c) 12-Scenes. 54.6% shared.

5 Conclusion

This paper presents ReCoLoc, a region-to-coordinate framework for 3DGS-based visual localization. To overcome the coordinate discrepancy caused by center-based assignment, we endow Gaussian landmarks with query-conditioned coordinate functions. Contribution-weighted identity encoding retrieves the relevant region, and coordinate decoding determines the corresponding point, making point determination an explicit part of 2D-3D matching. Considering that correspondence quality depends on both operations, we devise a joint criterion to select reliable landmarks. To reduce their storage cost, we further develop a shared-subspace representation that compresses descriptors and coordinate functions together. Extensive evaluations conducted across three localization benchmarks showcase ReCoLoc's superiority for accurate 3DGS-based visual localization.

AI USE STATEMENT

In this work, generative AI tools were used for language editing, including rephrasing and grammar checking, for checking bibliographic metadata, and for assisting in diagnosing software issues during development. All AI-assisted revisions, bibliographic checks, and diagnostic suggestions were carefully reviewed by the authors. Generative AI tools were not used to generate research ideas, design the proposed method or experiments, write or implement code, process or generate data, or interpret the experimental results. We take full responsibility for the final content of this work, including its text, code, claims, and experimental results.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithms 1-2 specify landmark construction, selection, and compression, and Section 4.1 lists the parameters used in all experiments. Appendix A provides the 3DGS reconstruction settings, the mapping-view split and candidate settings, the quantization procedure, the localization settings, and the closed-form coordinate fit. Appendix E describes the hardware and timing protocol. All datasets are public, and Section 4.1 identifies the reference poses used for evaluation.

REFERENCES

Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE TPAMI*, 44(9):5847–5865, 2022.

Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *ICCV*, pp. 6218–6228, 2021.

Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using RGB and poses. In *CVPR*, pp. 5044–5053, 2023.

Shuai Chen, Xinghui Li, Zirui Wang, and Victor Adrian Prisacariu. DFNet: Enhance absolute pose regression with direct feature matching. In *ECCV*, pp. 1–17, 2022.

Shuai Chen, Yash Bhalgat, Xinghui Li, Jia-Wang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Neural refinement for absolute pose regression with feature synthesis. In *CVPR*, pp. 20987–20996, 2024.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPRW*, pp. 224–236, 2018.

Yingdong Gu, Shaocheng Yan, Zhenjun Zhao, Yuan Kou, Jianxin Luo, Pengcheng Shi, and Jiayuan Li. ULF-Loc: Unbiased landmark feature for robust visual localization with 3D Gaussian splatting. In *CVPR*, pp. 19065–19075, 2026.

Zhiwei Huang, Hailin Yu, Yichun Shentu, Jin Yuan, and Guofeng Zhang. From sparse to dense: Camera relocalization with scene-specific detector from feature Gaussian splatting. In *CVPR*, pp. 27059–27069, 2025.

Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *ICCV*, pp. 2938–2946, 2015.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139:1–139:14, 2023.

Viktor Larsson and contributors. PoseLib - minimal solvers for camera pose estimation. https://github.com/vlarsson/PoseLib, 2020.

Miso Lee, Sangeek Hyun, Yerim Jeon, and Jae-Pil Heo. Disambiguating 2D-3D correspondences in Gaussian splatting-based feature fields for visual localization. *arXiv preprint arXiv:2605.07351*, 2026.

Xi Li, Tong Rao, and Cihui Pan. EDM: Efficient deep feature matching. In *ICCV*, pp. 26198–26208, 2025.

Changkun Liu, Shuai Chen, Yash Bhalgat, Siyan Hu, Ming Cheng, Zirui Wang, Victor Adrian Prisacariu, and Tristan Braud. GS-CPR: Efficient camera pose refinement via 3D Gaussian splatting. In *ICLR*, pp. 76761–76780, 2025.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pp. 405–421, 2020.

Arthur Moreau, Nathan Piasco, Moussab Bennehar, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. CROSSFIRE: Camera relocalization on self-supervised features from an implicit representation. In *ICCV*, pp. 252–262, 2023.

Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R. Nascimento. XFeat: Accelerated features for lightweight image matching. In *CVPR*, pp. 2682–2691, 2024.

Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2D2: Reliable and repeatable detector and descriptor. In *NeurIPS*, volume 32, pp. 12405–12415, 2019.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, pp. 12716–12725, 2019.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, pp. 4938–4947, 2020.

Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, pp. 752–765, 2012.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pp. 4104–4113, 2016.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, pp. 2930–2937, 2013.

Gennady Sidorov, Malik Mohrat, Denis Gridusov, Ruslan Rakhimov, and Sergey Kolyubin. GSplatLoc: Grounding keypoint descriptors into 3D Gaussian splatting for improved visual localization. In *IROS*, pp. 12601–12607, 2025.

Shitao Tang, Sicong Tang, Andrea Tagliasacchi, Ping Tan, and Yasutaka Furukawa. NeuMap: Neural coordinate mapping by auto-transdecoder for camera localization. In *CVPR*, pp. 929–939, 2023.

Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*, pp. 323–332, 2016.

Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. PNeRFLoc: Visual localization with point-based neural radiance fields. In *AAAI*, pp. 7450–7459, 2024.

Qunjie Zhou, Maxim Maximov, Or Litany, and Laura Leal-Taixé. The NeRFect match: Exploring NeRF features for visual localization. In *ECCV*, pp. 108–127, 2024.

A Additional Method Details

A.1 RGB 3D Gaussian scene reconstruction

Scene representation. We reconstruct each scene with RGB 3DGS (Kerbl et al., 2023) to obtain the geometry of regional landmarks and the appearance used for pose refinement. Each primitive has a 3D center, an anisotropic covariance, an opacity, and spherical harmonic (SH) coefficients for view-dependent color. Projecting the primitives into a camera and compositing them in depth order produces an RGB image. The Gaussian regions form ReCoLoc's landmarks, while their rendering contributions associate mapping keypoints with those regions.

Reconstruction objective. We initialize Gaussian centers and colors from the SfM point cloud and optimize the scene using posed mapping images. For mapping view v, let I_v be its RGB image and $\hat{I}_v$ the corresponding rendering. We optimize

$$\mathcal{L}_{\mathrm{RGB}} = (1 - \lambda_{\mathrm{SSIM}})\ell_1(\hat{I}_v, I_v) + \lambda_{\mathrm{SSIM}}\big(1 - \mathrm{SSIM}(\hat{I}_v, I_v)\big), \tag{11}$$

where ℓ_1 is the mean absolute pixel error, SSIM is structural similarity, and $\lambda_{\mathrm{SSIM}} = 0.2$ balances the two terms. The resulting geometry and appearance are learned from RGB supervision.

Training settings. We use Adam for 30,000 iterations, sampling one mapping image per iteration. Images are rendered with an antialiased rasterizer and a white background, limiting the longest edge to 640 pixels while preserving the aspect ratio. Cambridge uses undistorted images with object and invalid-border masks; sky pixels in the target are set to the background color. SH degree starts at zero and increases by one every 1,000 iterations, up to degree three. Table 8 lists the learning rates and densification thresholds. Position learning rates decay exponentially over training and are multiplied by the scene extent, following the 3DGS parameterization.

Table 8: **RGB 3DGS training settings.** Learning rates are shared across scenes within each dataset. Position rates are listed before scaling by the scene extent.

Setting	Cambridge	7-Scenes/12-Scenes
Initial position learning rate	1.6×10^{-5}	1.6×10^{-4}
Final position learning rate	1.6×10^{-6}	1.6×10^{-6}
Scale learning rate	1.0×10^{-3}	5.0×10^{-3}
Rotation learning rate	1.0×10^{-3}	1.0×10^{-3}
Opacity learning rate	5.0×10^{-2}	5.0×10^{-2}
Degree-zero SH learning rate	2.5×10^{-3}	2.5×10^{-3}
Higher-order SH learning rate	1.25×10^{-4}	1.25×10^{-4}
Densification gradient threshold	4.0×10^{-4}	2.0×10^{-4}

Adaptive densification runs every 100 iterations between iterations 500 and 15,000, cloning or splitting primitives according to their projected gradients and spatial scales. Primitives with opacity below 0.005 are pruned. Opacity is reset every 3,000 iterations during this stage, with an additional reset at iteration 500 for the white background.

A.2 Landmark selection

Algorithm 1 selects regions by the quality of their retrieved and decoded correspondences. Let $\mathcal{O}_i^{\mathrm{fit}}$ denote region i's observations in the fitting views. Its effective observation count is

$$s_i^{\mathrm{eff}} = \frac{\big(\sum_{(v,k)\in\mathcal{O}_i^{\mathrm{fit}}} w_{ivk}\big)^2}{\sum_{(v,k)\in\mathcal{O}_i^{\mathrm{fit}}} w_{ivk}^2}. \tag{12}$$

Eligible candidates have at least n_{obs} observations from n_{view} views and an effective observation count greater than $s_{\min}$. The spatial sampler in Algorithm 2 ranks candidates within neighborhoods to maintain scene coverage. Here $\mathrm{views}(\mathcal{O}_i^{\mathrm{fit}})$ returns the distinct view indices represented in the observations, and $\mathcal{E}$ denotes eligible Gaussian indices. We extract up to M keypoints per image, form

a pool of up to C candidate landmarks, and select N landmarks. The map $\mathcal{M}$ stores the geometry, descriptor, and coordinate function for each selected index in $\mathcal{L}$.

For the spatial sampler, $\mathcal{G}'$ is the input index set, n is the requested count, and S_i is the ranking score for Gaussian i. We draw neighborhood anchors $\mathcal{A}$ uniformly without replacement. For anchor a, $\mathcal{N}_a$ contains the indices of the k_{n} nearest candidate centers to that anchor, with already selected indices removed. In the final selection, S_i ranks Q_i, breaking ties by s_i^{eff}. The arrays S, Q, and s^{eff} collect these values over the input candidates.

Fitting and scoring views. We divide mapping views into fitting and scoring sets to evaluate candidates before final landmark selection. We sort mapping images by filename and assign the first 20 views in each consecutive block of 100 to scoring, using the remaining 80 for fitting. The final incomplete block follows the same rule, giving an approximately 80:20 fitting/scoring split. We initially compute effective observation counts, identity descriptors, and coordinate functions from the fitting views. The scoring views then evaluate correspondence quality for landmark selection. Finally, we refit the selected descriptors and coordinate functions using all mapping views.

Candidate selection settings. We require candidates to be observed across multiple views, setting $n_{\mathrm{obs}} = 8$, $n_{\mathrm{view}} = 8$, and $s_{\min} = 4$ and retaining Gaussians whose effective observation count exceeds $s_{\min}$. The candidate pool contains up to $C = 40{,}000$ Gaussians; when fewer are eligible, we use all of them. Both spatial sampling stages use $k_{\mathrm{n}} = 32$ neighbors. We retain $N = 20{,}000$ landmarks and extract up to $M = 2{,}048$ keypoints per mapping, scoring, or query image.

Algorithm 1 Retrieval and coordinate quality for Gaussian map selection

Require: Gaussians $\mathcal{G}$; mapping images $\mathcal{V}$ and cameras; encoder ϕ; split $\mathcal{V}_{\mathrm{fit}}, \mathcal{V}_{\mathrm{score}}$; keypoint limit M; candidate limit C; landmark count N; parameters $\tau, \rho, \lambda, \sigma, \kappa, \eta_{\mathrm{sim}}$; observation thresholds $n_{\mathrm{obs}}, n_{\mathrm{view}}, s_{\min}$; neighbors k_{n}
Ensure: Regional map $\mathcal{M} = \{(\boldsymbol{\mu}_i, \boldsymbol{H}_i, \boldsymbol{c}_i, \mathcal{F}_i) : i \in \mathcal{L}\}$

$\{\mathcal{O}_i\} \leftarrow$ RegionalObservations$(\mathcal{G}, \mathcal{V}, \phi, M, \tau, \rho)$
$\mathcal{E} \leftarrow \{i : |\mathcal{O}_i^{\mathrm{fit}}| \geq n_{\mathrm{obs}},\ |\mathrm{views}(\mathcal{O}_i^{\mathrm{fit}})| \geq n_{\mathrm{view}},\ s_i^{\mathrm{eff}} > s_{\min}\}$
$\mathcal{C} \leftarrow$ SpatialSelect$(\mathcal{E}, s^{\mathrm{eff}}, C, k_{\mathrm{n}})$
Fit $\{\boldsymbol{c}_i, \mathcal{F}_i\}_{i \in \mathcal{C}}$ on $\mathcal{V}_{\mathrm{fit}}$ using Eqs. (4)-(6)
for $v \in \mathcal{V}_{\mathrm{score}}$ **do**
    $\mathcal{A}_{iv} \leftarrow \{k : (v, k) \in \mathcal{O}_i\}$, $\mathcal{B}_{iv} \leftarrow \varnothing$, $g_{iv} \leftarrow 0$ for all $i \in \mathcal{C}$
    for each keypoint $(\boldsymbol{u}_{vk}, \boldsymbol{d}_{vk})$ in view v **do**
        $i^* \leftarrow \arg\max_{i \in \mathcal{C}} \boldsymbol{d}_{vk}^{\mathsf{T}} \boldsymbol{c}_i$
        if $\boldsymbol{d}_{vk}^{\mathsf{T}} \boldsymbol{c}_{i^*} > \eta_{\mathrm{sim}}$ **then**
            $\mathcal{B}_{i^*v} \leftarrow \mathcal{B}_{i^*v} \cup \{k\}$
            $\boldsymbol{x} \leftarrow \mathcal{F}_{i^*}(\boldsymbol{d}_{vk})$; $e \leftarrow \|\pi_v(\boldsymbol{x}) - \boldsymbol{u}_{vk}\|_2$
            if $k \in \mathcal{A}_{i^*v}$ and $\boldsymbol{x}$ has finite projection and positive depth **then**
                $g_{i^*v} \leftarrow g_{i^*v} + \exp(-e^2/(2\sigma^2))$
            end if
        end if
    end for
    $r_{iv} \leftarrow 2g_{iv}/(|\mathcal{A}_{iv}| + |\mathcal{B}_{iv}|)$ for every nonempty region-view pair
end for
$\mathcal{V}_i \leftarrow \{v : |\mathcal{A}_{iv}| + |\mathcal{B}_{iv}| > 0\}$; $r_0 \leftarrow \mathrm{mean}\{r_{iv} : i \in \mathcal{C},\ v \in \mathcal{V}_i\}$
$Q_i \leftarrow (\sum_{v \in \mathcal{V}_i} r_{iv} + \kappa r_0)/(|\mathcal{V}_i| + \kappa)$; use r_0 if the denominator is zero
$S \leftarrow$ Rank(Q, s^{eff})
$\mathcal{L} \leftarrow$ SpatialSelect$(\mathcal{C}, S, N, k_{\mathrm{n}})$
Refit $\{\boldsymbol{c}_i, \mathcal{F}_i\}_{i \in \mathcal{L}}$ using all observations $\mathcal{O}_i$
return $\{(\boldsymbol{\mu}_i, \boldsymbol{H}_i, \boldsymbol{c}_i, \mathcal{F}_i) : i \in \mathcal{L}\}$

Algorithm 2 Spatially distributed landmark selection

Require: Gaussian indices $\mathcal{G}'$ and centers; scores S; landmark count n; neighbors k_{n}
Ensure: Selected indices $\mathcal{L}$
$n \leftarrow \min(n, |\mathcal{G}'|)$; $\mathcal{L} \leftarrow \varnothing$
$\mathcal{A} \leftarrow \textsc{UniformSample}(\mathcal{G}', n)$
for $a \in \mathcal{A}$ **do**
  $\mathcal{N}_a \leftarrow \textsc{kNN}(\boldsymbol{\mu}_a, \{\boldsymbol{\mu}_i : i \in \mathcal{G}'\}, k_{\mathrm{n}}) \setminus \mathcal{L}$
  if $\mathcal{N}_a \neq \varnothing$ **then**
    $i^* \leftarrow \arg\max_{i \in \mathcal{N}_a} S_i$; $\mathcal{L} \leftarrow \mathcal{L} \cup \{i^*\}$
  end if
end for
Add the $n - |\mathcal{L}|$ highest-scoring indices in $\mathcal{G}' \setminus \mathcal{L}$ to $\mathcal{L}$
return $\mathcal{L}$

RegionalObservations extracts up to M descriptors per mapping image, computes Gaussian rendering contributions, and retains the associations defined in Section 3.2. Rank orders candidates by quality, then by effective observation count, using candidate order for exact ties. Algorithm 2 uses Euclidean distance between Gaussian centers and returns distinct indices. In Algorithm 1, i^* is the retrieved candidate, $\boldsymbol{x}$ its predicted point, and e its reprojection error; in Algorithm 2, i^* is the highest-ranked remaining neighbor.

A.3 Mean-preserving compression

We store each row of $\boldsymbol{A}_i$ as signed b_{q}-bit integers with a separate scale. For row index $r \in \{1, 2, 3\}$ and column index $h \in \{1, \ldots, d_{\mathrm{c}}\}$, define

$$q_{\max} = 2^{b_{\mathrm{q}}-1} - 1, \qquad \gamma_{ir} = \frac{\max_h |A_{irh}|}{q_{\max}}, \qquad \widehat{A}_{irh} = \gamma_{ir} \operatorname{round}(A_{irh}/\gamma_{ir}). \tag{13}$$

Here A_{irh} and $\widehat{A}_{irh}$ are entries of $\boldsymbol{A}_i$ and $\widehat{\boldsymbol{A}}_i$, respectively, $q_{\max}$ is the largest positive quantized value, γ_{ir} is the row scale, and round rounds to the nearest integer. An all-zero row is stored as zeros. To preserve the decoder's prediction at the weighted descriptor mean $\bar{\boldsymbol{d}}_i$, we correct the intercept after projection and quantization. With dequantized coefficients $\widehat{\boldsymbol{A}}_i$, the compact decoder is

$$\widetilde{\boldsymbol{b}}_i = \boldsymbol{b}_i + \boldsymbol{L}_i\bar{\boldsymbol{d}}_i - \widehat{\boldsymbol{A}}_i\boldsymbol{P}^{\mathsf{T}}\bar{\boldsymbol{d}}_i, \qquad \widetilde{\mathcal{F}}_i(\boldsymbol{q}) = \boldsymbol{\mu}_i + \boldsymbol{H}_i\Pi_\rho(\widehat{\boldsymbol{A}}_i\boldsymbol{P}^{\mathsf{T}}\boldsymbol{q} + \widetilde{\boldsymbol{b}}_i). \tag{14}$$

Substituting $\boldsymbol{q} = \bar{\boldsymbol{d}}_i$ gives $\widehat{\boldsymbol{A}}_i\boldsymbol{P}^{\mathsf{T}}\bar{\boldsymbol{d}}_i + \widetilde{\boldsymbol{b}}_i = \boldsymbol{L}_i\bar{\boldsymbol{d}}_i + \boldsymbol{b}_i$. This identity aligns the compact and full predictors at the weighted descriptor mean before the final storage conversion. Compression therefore changes the query-dependent displacement around that mean.

A.4 Localization settings

Direct matching. For each query descriptor, we retrieve the most similar Gaussian and retain matches with positive similarity scores to establish 2D-3D correspondences. We use $\eta_{\mathrm{sim}} = 0$ in both mapping-view scoring and query localization. Tiny-ReCoLoc uses the unnormalized projected inner products with the same threshold.

Pose estimation. We estimate the camera pose using PoseLib RANSAC-PnP. The reprojection threshold is 12 pixels and the confidence is 0.99999, with 1,000 to 100,000 RANSAC iterations. A refinement iteration updates the pose only when sufficient matches and PnP inliers are available; otherwise, it retains the previous estimate.

A.5 Closed-form coordinate fitting

Let $\delta\boldsymbol{d}_{ivk} = \boldsymbol{d}_{vk} - \bar{\boldsymbol{d}}_i$ and $\delta\boldsymbol{\xi}_{ivk} = \boldsymbol{\xi}_{ivk} - \bar{\boldsymbol{\xi}}_i$. The normal equations for Eq. (5) yield $\boldsymbol{b}_i = \bar{\boldsymbol{\xi}}_i - \boldsymbol{L}_i\bar{\boldsymbol{d}}_i$ and

$$\boldsymbol{L}_i = \left(\sum_{(v,k)\in\mathcal{O}_i} \bar{w}_{ivk}\,\delta\boldsymbol{\xi}_{ivk}\delta\boldsymbol{d}_{ivk}^{\mathsf{T}}\right)\left(\sum_{(v,k)\in\mathcal{O}_i} \bar{w}_{ivk}\,\delta\boldsymbol{d}_{ivk}\delta\boldsymbol{d}_{ivk}^{\mathsf{T}} + \lambda\boldsymbol{I}_D\right)^{-1}.$$

Here, $\boldsymbol{I}_D$ is the $D \times D$ identity matrix, and $\lambda > 0$ ensures invertibility.

B Parameter and Compression Analysis

B.1 Parameter sensitivity

We study five choices in landmark fitting and selection on Cambridge Landmarks. Each comparison varies one factor from the default configuration and constructs a new map of 20,000 landmarks. Table 9 reports recall at 15 cm/5°, averaged over the five scenes.

Table 9: **Parameter analysis on Cambridge Landmarks.** Recall (%) at 15 cm/5° is averaged over five scenes. Each sub-table varies one factor from the default configuration (gray); the best result in each sub-table is bold.

λ	[15/5] ↑
0.001	**81.78**
0.01	80.01
0.1	77.53

(a) Ridge regularization.

σ (px)	[15/5] ↑
2.5	81.65
5	**81.78**
10	81.42

(b) Reprojection scale.

Strategy	[15/5] ↑
Obs. count	80.37
Retrieval	80.70
Coordinate	80.59
Joint	**81.78**

(c) Map selection.

τ	[15/5] ↑
0.01	80.59
0.05	**81.78**
0.1	81.57

(d) Contribution threshold.

κ	[15/5] ↑
0	81.43
4	**81.78**
8	81.47

(e) Prior strength.

Strength of the coordinate correction. The ridge coefficient λ controls the magnitude of the query-dependent term in Eq. (6). Increasing it suppresses variation around the regional mean coordinate. Recall falls from 81.78% to 77.53% as λ increases from 10^{-3} to 10^{-1}, favoring a decoder that retains more of this variation. This agrees with the coordinate comparison in Table 5, where query-conditioned coordinates outperform regional means.

Information used for landmark selection. We compare landmark selection by effective observation count, retrieval quality, coordinate accuracy, and the joint criterion. In Table 9(c), retrieval and coordinate scores each improve on selection by effective observation count. Combining the two raises recall to 81.78%, exceeding the three alternatives by 1.41, 1.08, and 1.19 percentage points, respectively. Retrieval quality and coordinate accuracy therefore guide selection toward complementary properties: reliable region matching and accurate point prediction. The resulting maps improve localization with the same number of landmarks.

Strength of pixel-Gaussian associations. The contribution threshold τ controls which observations enter the descriptor and decoder fits. A lower threshold admits more weak associations, while a higher threshold restricts the observations to stronger contributors. The intermediate value $\tau = 0.05$ reaches 81.78% recall, compared with 80.59% at 0.01 and 81.57% at 0.1. These results favor balancing observation coverage with association strength when constructing a regional landmark.

Tolerance and smoothing of the selection score. The scale σ sets how strongly reprojection errors reduce a region's score; κ regularizes scores estimated from few views toward the shared prior. Recall varies by 0.36 percentage points across the three reprojection scales and by 0.35 points across prior strengths, so selection is robust to both parameters around the defaults of $\sigma = 5$ pixels and $\kappa = 4$. Smoothing with $\kappa = 4$, which stabilizes the scores of sparsely observed candidates, improves recall by 0.35 points over the unsmoothed estimate ($\kappa = 0$).

B.2 REGIONAL COORDINATE REPRESENTATIONS

We compare the affine decoder with regional nearest-neighbor retrieval and prototype interpolation on Cambridge Landmarks to examine how coordinate representation affects localization. For each observation-selected or joint-selected map, the alternatives share observations, identity descriptors, Top-1 matches, and PnP settings; they differ in the coordinate output within the retrieved Gaussian.

Regional retrieval and interpolation. The observation nearest-neighbor baseline retains all associated feature-coordinate pairs and returns the coordinate whose descriptor is most similar to the query. The prototype baselines summarize these observations with K feature-coordinate pairs per Gaussian, where K is the prototype count. We obtain feature prototypes by contribution-weighted spherical clustering into K groups and compute each coordinate as the weighted mean of its assigned targets. Nearest-prototype retrieval returns one coordinate; prototype interpolation combines them using softmax weights on query-prototype cosine similarities. We select its temperature from $\{0.02, 0.05, 0.1, 0.2, 0.5\}$ using mapping views reserved for validation, then refit on all mapping views. All predictions use the same local coordinate bound. Storage includes the identity descriptors, Gaussian geometry, and coordinate model; feature-coordinate pairs and affine coefficients use FP16.

Table 10: **Coordinate representations and map storage on Cambridge Landmarks.** Each map contains 20,000 landmarks. Recall is measured at 15 cm/5°; errors are scene-median translation/rotation errors (cm/°). All entries are averaged over five scenes. K denotes the number of prototypes per Gaussian.

Coordinate output	Observation-selected			Joint-selected		
	Map [MiB]	Recall ↑	Error ↓	Map [MiB]	Recall ↑	Error ↓
Observation nearest neighbor	1,013.08	**80.91**	**6.3/0.11**	771.35	81.48	**6.0/0.11**
Nearest prototype, $K = 2$	40.42	74.32	8.0/0.13	40.42	75.87	7.6/0.13
Prototype interpolation, $K = 2$	40.42	74.51	8.1/0.13	40.42	76.14	7.6/0.13
Nearest prototype, $K = 3$	50.32	74.83	8.1/0.13	50.32	75.59	7.7/0.13
Prototype interpolation, $K = 3$	50.32	74.82	8.2/0.13	50.32	76.01	7.7/0.13
Nearest prototype, $K = 4$	60.22	75.06	8.0/0.13	60.22	76.19	7.5/0.13
Prototype interpolation, $K = 4$	60.22	75.34	7.9/0.13	60.22	75.89	7.5/0.12
Nearest prototype, $K = 8$	99.81	76.67	7.3/0.12	99.81	77.39	7.1/0.12
Prototype interpolation, $K = 8$	99.81	76.47	7.6/0.12	99.81	77.35	7.4/0.12
ReCoLoc	50.03	80.37	6.6/0.12	50.03	**81.78**	6.3/**0.11**

Accuracy and storage. As shown in Table 10, at nearly the same storage as ReCoLoc (50.32 versus 50.03 MiB), three-prototype retrieval and interpolation reach 75.59% and 76.01% strict recall on joint-selected maps, compared with 81.78% for the affine decoder. Increasing to eight prototypes nearly doubles storage to 99.81 MiB, while the decoder still achieves higher recall and lower translation error. These advantages also hold on observation-selected maps.

Retaining every observation lowers the translation error on joint-selected maps from 6.3 cm to 6.0 cm, but requires 771.35 MiB, 15 times the storage of ReCoLoc, for similar strict recall (81.48% versus 81.78%). On observation-selected maps, it achieves 80.91% recall versus 80.37% for ReCoLoc at 1,013.08 MiB. The affine decoder uses substantially less storage than observation retrieval and achieves higher recall than the prototype methods.

B.3 Map storage and coordinate accuracy

Descriptor dimension. To examine the storage-accuracy trade-off, we project the same fitted model to 32, 64, and 128 dimensions, retaining the same 20,000 landmarks. All variants use 8-bit coordinate coefficients and 16-bit descriptors and auxiliary parameters; map size includes the shared basis. As shown in Fig. 3, 64 dimensions preserve most of the full model's accuracy at 7.45 MiB per scene. Increasing to 128 dimensions nearly doubles the map to 13.59 MiB but raises strict recall by at most 0.67 percentage points (on Cambridge). Reducing to 32 dimensions gives a 4.39 MiB map, but strict recall on 7-Scenes falls from 68.25% to 64.00%. We use 64 dimensions to balance map size and accuracy across datasets.

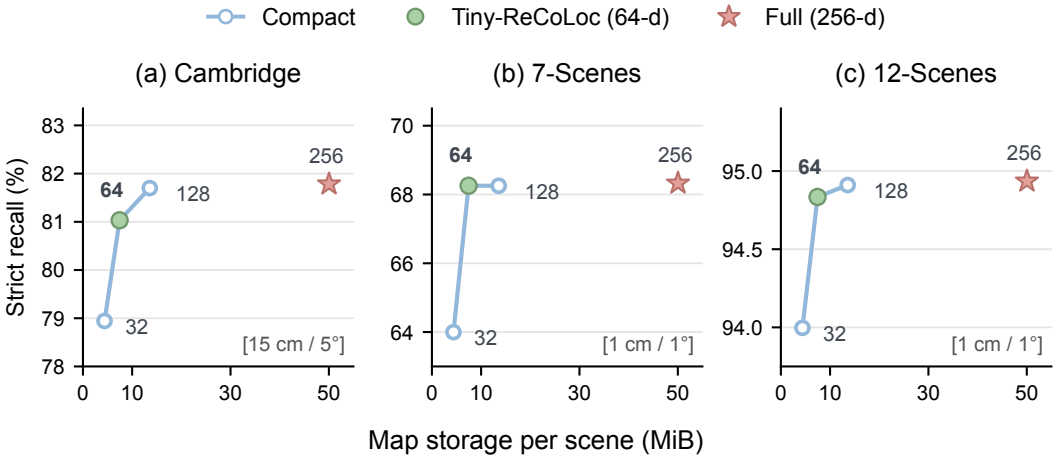

Figure 3: **Map storage and localization accuracy.** Labels indicate descriptor dimension; the filled circle marks Tiny-ReCoLoc and the star marks the full model. Strict recall is averaged over scenes.

Information used to construct the shared basis. We compare the joint basis in Eq. (9) with an identity-only basis on 7-Scenes to examine the role of coordinate prediction in compression. Both 64-dimensional variants use the same fitted models and compression procedure, occupying 7.45 MiB per scene. As shown in Table 11, including the decoder coefficients raises strict recall from 66.86% to 68.25% and correspondence accuracy from 43.54% to 45.58%, yielding gains of 1.39 and 2.04 percentage points, respectively. Accounting for coordinate prediction when constructing the basis therefore improves both correspondence quality and localization accuracy at the same storage cost.

Table 11: **Shared-basis ablation on 7-Scenes.** R is scene-averaged recall (%) at 1 cm/1°; C is correspondence accuracy (%) at 5 pixels.

Basis	$R \uparrow$	$C \uparrow$
Identity only	66.86	43.54
Joint (Eq. (9))	**68.25**	**45.58**

C Descriptor and Coordinate Analysis

C.1 Alternative local features

To evaluate compatibility with different feature extractors, we replace SuperPoint with XFeat (Potje et al., 2024) and R2D2 (Revaud et al., 2019) on all three datasets. Each extracts up to 2,048 keypoints per image, with 64- and 128-dimensional descriptors, respectively. We rebuild both identities and decoders from these features.

Table 12 shows that R2D2 improves Cambridge recall and reduces the average median translation error to 6.0 cm, while SuperPoint gives the highest strict recall indoors. Compared with the results in Table 4, ReCoLoc without refinement exceeds the strict recall of STDLoc and ULF-Loc without refinement on all three datasets with each of the three features.

Table 12: **Local feature replacement.** Scene-averaged median errors (cm/°) and recall (%) at the listed thresholds.

Local features	Error↓	Recall (%)↑		
Cambridge		[15/5]	[25/5]	[50/5]
SuperPoint	6.3/**0.11**	81.78	89.83	95.97
XFeat	7.2/0.13	78.11	88.33	95.80
R2D2	**6.0**/0.12	**82.87**	**91.01**	**96.41**
7-Scenes		[1/1]	[2/2]	[5/5]
SuperPoint	**0.78/0.22**	**68.33**	**92.16**	**99.11**
XFeat	0.90/0.28	63.10	86.89	97.00
R2D2	0.81/0.24	67.37	89.12	98.80
12-Scenes		[1/1]	[2/2]	[5/5]
SuperPoint	**0.33/0.14**	**94.94**	99.25	99.74
XFeat	0.38/0.16	93.15	99.21	**99.98**
R2D2	**0.33/0.14**	94.85	**99.36**	99.74

C.2 Coordinate decoding with ULF-Loc descriptors

To examine coordinate prediction with another retrieval rule, we construct ULF-Loc descriptors on an observation-selected map. We compare Gaussian centers, contribution-weighted means, and decoded coordinates using the same observations and Top-1 matches.

As shown in Table 13, the regional mean improves strict recall over the center on Cambridge but reduces it indoors. The decoder improves both recall and correspondence accuracy over either query-independent coordinate on all three datasets. Relative to the regional mean, strict recall rises from 59.87% to 70.10% on Cambridge, from 47.03% to 63.03% on 7-Scenes, and from 83.71% to 90.67% on 12-Scenes. Thus, query-conditioned coordinate prediction improves localization with ULF-Loc retrieval descriptors as well.

Table 13: **Coordinate prediction with ULF-Loc identities.** Metrics follow Table 5.

Output	Cambridge			7-Scenes			12-Scenes		
	Error↓	$R\uparrow$	$C\uparrow$	Error↓	$R\uparrow$	$C\uparrow$	Error↓	$R\uparrow$	$C\uparrow$
Center	12.9/0.22	58.53	7.30	1.08/0.30	48.65	28.97	0.49/0.22	87.18	13.35
Weighted mean	12.4/0.23	59.87	7.96	1.15/0.37	47.03	24.51	0.54/0.22	83.71	11.98
Decoder	**9.2/0.16**	**70.10**	**11.18**	**0.86/0.27**	**63.03**	**36.23**	**0.41/0.17**	**90.67**	**20.90**

D Additional Localization Results

D.1 Per-scene results on 12-Scenes

Table 14 gives the results for individual 12-Scenes environments. ReCoLoc reaches an average scene-median error of 0.33 cm/0.14° before refinement, with translation medians below 0.5 cm in all twelve scenes. One refinement iteration reduces the average translation error to 0.31 cm; the average rotation error is 0.14°. Tiny-ReCoLoc also reaches an average scene-median error of 0.33 cm/0.14°.

Table 14: **Per-scene localization results on 12-Scenes.**

	Method	Apartment 1		Apartment 2				Office 1				Office 2		Avg.↓ [cm/°]
		Kitchen	Living	Bed	Kitchen	Living	Luke	Gates362	Gates381	Lounge	Manolis	5a	5b	
SCR	DSAC*	-	-	-	-	-	-	-	-	-	-	-	-	0.5/0.25
SCR	ACE	0.53/0.27	0.60/0.19	0.49/0.20	0.63/0.26	0.59/0.23	0.76/0.27	0.69/0.22	0.81/0.32	0.84/0.21	0.76/0.28	0.86/0.33	0.82/0.28	0.7/0.26
NeRF	PNeRFLoc	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	-	2.8/0.9	2.8/1.08
3DGS	GSplatLoc	1.31/0.24	0.68/0.21	1.42/0.24	0.67/0.26	0.54/0.19	1.84/0.31	1.82/0.26	1.99/0.26	0.64/0.22	0.78/0.27	3.66/0.37	0.94/0.27	1.4/0.26
3DGS	ACE+GS-CPR	0.46/0.22	0.44/0.17	0.44/0.18	0.40/0.19	0.50/0.20	0.64/0.27	0.57/0.20	0.56/0.24	0.71/0.21	0.51/0.20	0.68/0.27	0.57/0.19	0.5/0.21
3DGS	STDLoc	0.36/0.21	0.49/0.22	0.53/0.25	0.38/0.21	0.46/0.20	0.65/0.29	0.56/0.21	0.51/0.25	0.66/0.24	0.51/0.26	0.91/0.38	0.72/0.27	0.56/0.25
3DGS	STDLoc (PR)	0.24/0.15	0.26/0.13	0.25/0.11	0.27/0.16	0.25/0.12	0.43/0.16	0.35/0.14	0.29/0.13	0.43/0.13	0.32/0.14	0.44/0.20	0.37/0.15	0.33/0.14
3DGS	ULF-Loc	0.33/0.19	0.45/0.18	0.49/0.22	0.40/0.19	0.40/0.17	0.58/0.23	0.54/0.18	0.51/0.24	0.63/0.17	0.50/0.23	0.62/0.27	0.60/0.24	0.50/0.21
3DGS	ULF-Loc (PR)	0.31/0.19	0.28/0.14	0.28/0.12	0.26/0.17	0.29/0.13	0.45/0.17	0.38/0.17	0.31/0.14	0.45/0.15	0.34/0.17	0.51/0.24	0.44/0.19	0.36/0.16
3DGS	**Tiny-ReCoLoc**	0.25/0.15	0.27/0.12	0.30/0.15	0.27/0.14	0.29/0.12	0.43/0.15	0.31/0.12	0.33/0.15	0.40/0.13	0.34/0.15	0.44/0.20	0.34/0.12	0.33/0.14
3DGS	**ReCoLoc**	0.25/0.15	0.27/0.12	0.30/0.15	0.27/0.14	0.30/0.12	0.43/0.15	0.31/0.11	0.33/0.15	0.40/0.12	0.34/0.15	0.44/0.20	0.34/0.12	0.33/0.14
3DGS	**ReCoLoc (PR)**	0.25/0.15	0.25/0.12	0.28/0.14	0.25/0.17	0.26/0.12	0.42/0.16	0.32/0.12	0.29/0.13	0.38/0.12	0.29/0.13	0.40/0.18	0.30/0.13	0.31/0.14

D.2 Spatial distribution of localization errors

We visualize localization results across all seven 7-Scenes environments after two refinement iterations in Fig. 4. Most query poses have low translation and rotation errors. Larger residual errors are concentrated along limited portions of the trajectories, particularly in Stairs.

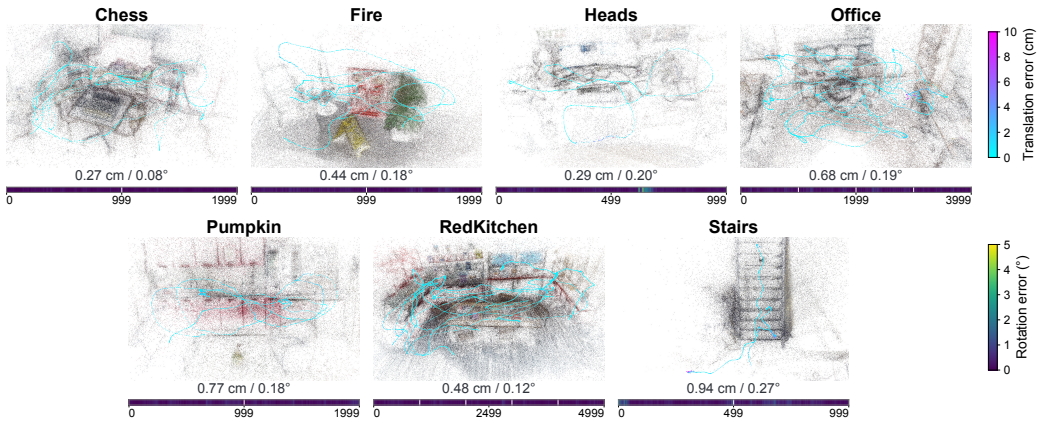

Figure 4: **Qualitative localization results on 7-Scenes.** Camera positions are colored by translation error; the strips show rotation error in query order. Values above the strips are median translation/rotation errors.

D.3 CORRESPONDENCE QUALITY AND IMAGE ALIGNMENT

The examples from the three datasets in Fig. 5 illustrate how correspondence quality affects localization. ReCoLoc yields more geometrically consistent matches, reflected in higher inlier ratios and lower pose errors. The enlarged regions make the effect visible: structural edges remain displaced across the query-render boundary for ULF-Loc, while ReCoLoc aligns them more closely.

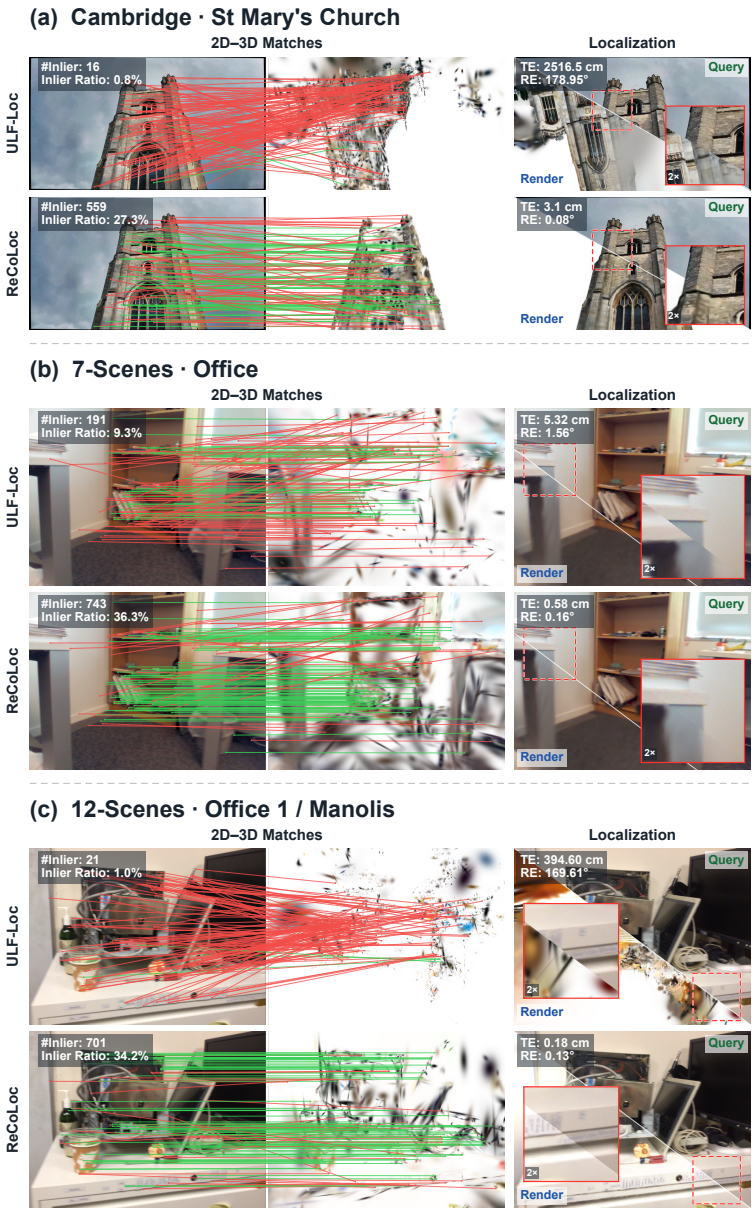

Figure 5: **2D-3D matching and localization without pose refinement.** Across the three datasets, these examples show more correct matches and closer query-render alignment with ReCoLoc (bottom) than with ULF-Loc (top). Green/red lines indicate correct/incorrect matches at a 5-pixel reprojection threshold. Statistics use all 2,048 matches; 128 are displayed. The localization panels combine the query (upper right) and estimated-pose rendering (lower left); the enlarged regions show alignment along visible edges. TE/RE denote translation/rotation errors.

E COMPUTATIONAL COST

We measure inference on an NVIDIA RTX 4090 with an Intel Xeon Platinum 8368Q CPU. For each scene, we select 100 uniformly spaced queries and average their runtimes over three runs after 20 warm-up queries. Table 15 separates match construction from pose estimation and refinement.

Query-time cost. Retrieval and coordinate decoding together require 1.31-1.39 ms across the datasets, while PnP accounts for most of the initial localization time. Tiny-ReCoLoc combines retrieval and decoding in a single kernel, taking 0.33 ms on Cambridge and 0.30 ms on each indoor dataset. ReCoLoc takes 110.67 ms on 7-Scenes and 88.99 ms on 12-Scenes including refinement.

Table 15: **Localization runtime.** Mean time per query (ms), averaged over scenes.

Stage	Cambridge	7-Scenes	12-Scenes
Feature extraction	25.75	2.98	9.75
Identity retrieval	0.77	0.75	0.76
Coordinate decoding	0.62	0.56	0.57
PnP	101.85	56.51	53.12
Total (w/o PR)	**128.99**	**60.79**	**64.21**
Iteration 1	-	25.42	24.78
Iteration 2	-	24.46	-
Total (with PR)	-	**110.67**	**88.99**

We compare ReCoLoc with ULF-Loc on 7-Scenes using SuperPoint and identical PoseLib settings and report the stage-wise runtime in Fig. 6. Total localization time decreases from 129.78 ms to 60.79 ms, primarily through the PnP stage, which falls from 126.06 ms to 56.51 ms. This is because ReCoLoc raises the mean PnP inlier ratio from 39.21% to 53.39%, which makes consistent pose hypotheses more likely during RANSAC-based robust estimation. Notably, coordinate decoding takes 0.56 ms per query.

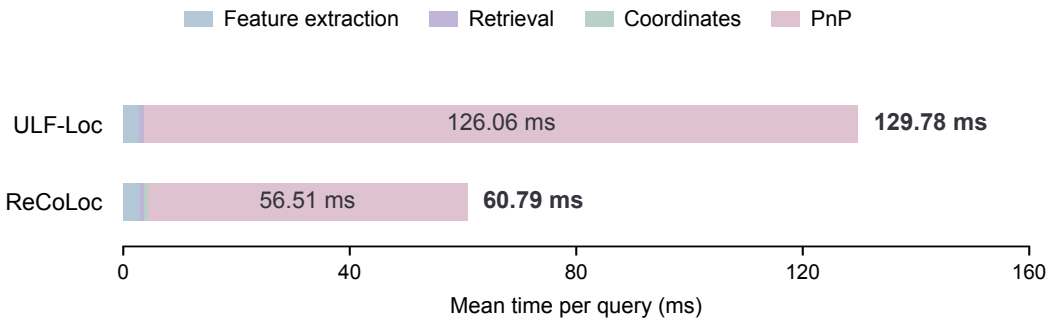

Figure 6: **Runtime comparison on 7-Scenes.**

Map construction. Table 16 lists the cost on one scene from each dataset. Starting from a pretrained RGB 3DGS, map construction takes 1.76-4.23 minutes. It comprises multiview observation collection, descriptor and decoder fitting, and landmark selection.

Table 16: **Map construction cost.** Time to build localization maps from pretrained RGB 3DGS. GPU memory is peak allocated/reserved memory.

Scene	Mapping views	Time (min)	GPU memory (MiB)
King's College	1220	4.23	2368/3894
Fire	2000	3.06	3736/4628
Apartment 1 Kitchen	744	1.76	1521/1978

F LIMITATIONS AND FUTURE WORK

Without ground-truth scene coordinates, ReCoLoc learns coordinate functions from the maximum-density point of each Gaussian along a mapping ray. These pseudo-coordinate targets lie on the image ray, but depend on the fitted Gaussian geometry and can deviate from the physical surface. Observations of the same physical point from different views can therefore receive different pseudo-coordinates, limiting the cross-view consistency of the training targets.

Future work could incorporate multiview SfM tracks or aligned depth observations as cross-view geometric supervision. These observations could constrain targets for the same physical point to a common 3D location and improve the cross-view consistency of coordinate prediction.