FLEXGLUE: SPARSE-TO-DENSE FEATURE MATCHING AND VISUAL DISAMBIGUATION

Anonymous authors
Paper under double-blind review

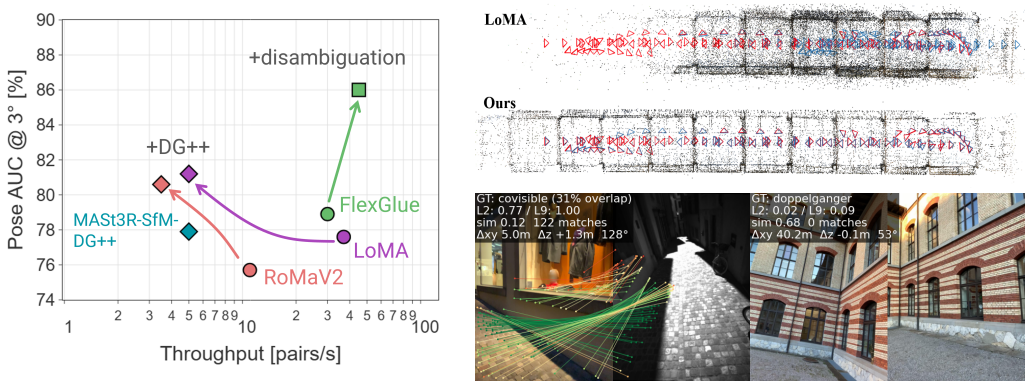


Figure 1: **Efficient and robust matching with FlexGlue.** Deep feature matchers are only trained on covisible pairs, yielding abundant and incorrect self-similar correspondences otherwise. This causes catastrophic 3D reconstruction failures (top right). Dedicated disambiguation works are expensive and do not generalize well because they are trained separate. In contrast, we propose to learn sparse-to-dense feature matching and visual disambiguation together, which yields increased robustness and accuracy at higher throughput (left). To measure this, we propose the first indoor and cross-device doppelgangers benchmark, with challenging positives and hard mined negatives (bottom-right).

ABSTRACT

Estimating robust correspondences across images is a fundamental building block of geometric computer vision. Existing approaches force a trade-off as they match either sparse points, which is highly efficient, or dense pixel grids, which is more robust but expensive. They also often struggle to reject unrelated image pairs that exhibit visual symmetries. We bridge these gaps by introducing a neural network for asymmetric sparse-to-dense image matching and view-graph filtering. Given two images and query points in the first one, we predict whether the image pair is covisible as well as the subpixel coordinates of the corresponding points in the second image. By searching for correspondences over the entire target image, the sparse-to-dense formulation is not limited by the repeatability of keypoint detectors. Furthermore, aggregated context from images provides global understanding necessary for both robust geometric matching and effective visual disambiguation, while jointly learning these tasks leverages their inherent synergy. Experiments on relative pose estimation, Structure-from-Motion, and visual localization show that our method retains subpixel precision and adds robustness to doppelgangers, while achieving computational efficiency of sparse matchers. Together, our method achieves state-of-the art accuracy on IMC2025, InLoc, Doppelgangers, ViSym, ETH3D SfM with COLMAP, and our proposed indoor doppelgangers benchmark. Weights and code will be released upon acceptance.

1 INTRODUCTION

Image matching estimates correspondences across images and is a fundamental building block of 3D vision, from Structure-from-Motion (SfM) to visual localization. Most geometric applications

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.

only require correspondences for a sparse set of distinctive interest points (Lowe, 2004; Yi et al., 2016; DeTone et al., 2018), which both simplifies the problem and reduces its cost. These points are detected independently in each image and matched by a nearest-neighbor search of their descriptors or by deep networks that learn priors on motion and scene structure (Sarlin et al., 2020; Lindenberger et al., 2023; Nordström et al., 2026). Matching is thus fundamentally bounded by the detector: a 3D point can only be matched if it is detected in both views, and any localization noise of the detections propagates directly into the correspondences.

Dense matchers (Edstedt et al., 2023b;c; Leroy et al., 2024; Edstedt et al., 2026) lift this bound by matching all pixels directly, and are remarkably robust to large viewpoint and appearance changes while achieving subpixel accuracy. This comes at a high cost: they estimate and refine a correspondence for every pixel, which restricts them to low resolutions and makes training expensive, even though downstream applications typically use only a sparse subset. Multi-view applications such as SfM further require discrete tracks, forcing pipelines to snap dense correspondences to detected keypoints with additional heuristics (Sun et al., 2021; He et al., 2024). Semi-dense matchers (Sun et al., 2021; Li et al., 2025) reduce the cost by matching on a coarse grid, but quantize correspondences to grid cells and still scale with the image resolution. Existing matchers thus face a clear trade-off between the robustness of dense and the efficiency of sparse matching.

Beyond this trade-off, learned sparse and dense matchers share a global blind spot: they are trained exclusively on covisible image pairs. Real-world applications such as SfM and SLAM, however, must reject pairs that depict different places, and matchers that confidently match visually similar structures, *e.g.*, symmetric facades, cause catastrophic scene collapses, where classical methods like SIFT (Lowe, 2004) often still perform best. Recent works therefore add bespoke models that predict covisibility after matching (Cai et al., 2023; Xiangli et al., 2025), which introduces architectural redundancy and cost, although local matching and global disambiguation are intrinsically the same task: finding overlapping context.

To bridge these gaps, we introduce FlexGlue (Fig. 1), which matches sparse queries against dense context: given an arbitrary query point in one image, it predicts its subpixel correspondence in the other. A lightweight Transformer (Vaswani et al., 2017) aggregates context over a dense token grid of both images, and a cheap readout and refinement head extracts correspondences only at the queries (Fig. 2). Decoupling dense context aggregation from sparse querying removes the dependence on repeatable detections, since correspondences can land on any pixel, while retaining the efficiency of sparse matching. As a result, FlexGlue is substantially cheaper to train and run than dense matchers, while matching or even surpassing them on relative pose estimation, SfM, and visual localization.

Crucially, FlexGlue is trained jointly to match covisible pairs and to reject non-covisible ones, exploiting the synergy between matching and visual disambiguation. Since its dense context aggregation already builds a global understanding of the image pair, a lightweight covisibility head suffices to outperform bespoke disambiguation models at a fraction of their cost. As training data with symmetries is scarce, we further show that disambiguation can be learned label-free from flip augmentations. Detecting negative pairs in early layers moreover accelerates matching, in contrast to prior efficient matchers (Lindenberger et al., 2023), which stop early on easy covisible pairs but run the full network on unmatchable ones. In summary, our main contributions are as follows:

(i) **Efficient sparse-to-dense matching:** an architecture that combines the efficiency of sparse matchers with the robustness and accuracy of dense matchers, without a dense refinement network, achieving state-of-the-art accuracy on HardMatch and InLoc.

(ii) **Joint matching and disambiguation:** a joint formulation that outperforms dedicated disambiguation models at $5\times$ higher throughput, avoids wasting compute on unmatchable pairs, and can be learned label-free. Combined with our matcher, it significantly improves the robustness of SfM on ETH3D and the CVPR 2025 Image Matching Challenge, where we surpass the 1st-place solution on the ongoing leaderboard.

(iii) **Indoor doppelganger benchmark:** the first indoor benchmark for visual disambiguation, which we found to be a crucial blindspot of all current methods, built from LaMAR with a technique to mine hard doppelgangers that also yields indoor training data; our method improves over prior work by 10% AUROC.

The code and trained models will be made publicly available upon acceptance.

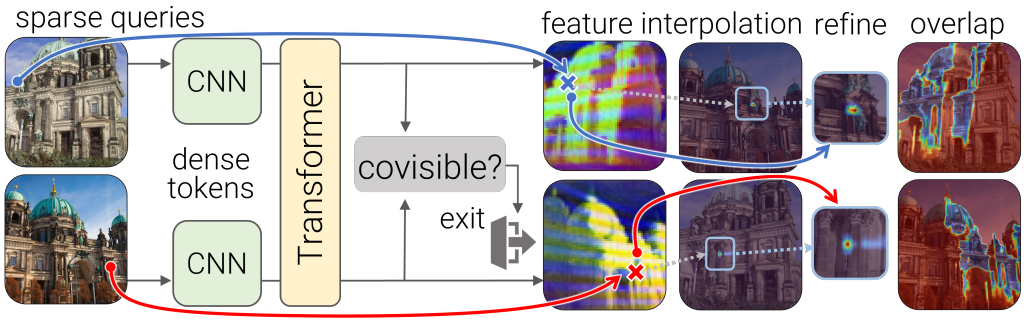


Figure 2: **Sparse-to-dense matching and disambiguation.** A pair of images is encoded with a DINOv3 ConvNeXt (Siméoni et al., 2025). Dense tokens are forwarded through a series of transformers gathering context within and between images. The aggregated tokens are interpolated at sparse keypoints, and we estimate their coarse correspondence with a windowed soft-argmax. A simple refinement head yields subpixel accuracy. We also detect unmatchable image pairs in every layer.

2 Related Work

Sparse feature matching detects keypoints independently in each image (Lowe, 2004; Yi et al., 2016; DeTone et al., 2018; Tyszkiewicz et al., 2020; Revaud et al., 2019; Shenoi et al., 2026; Zhao et al., 2023) and matches them by a nearest-neighbor search of their descriptors (DeTone et al., 2018; Mishchuk et al., 2017; Tyszkiewicz et al., 2020; Zhao et al., 2023) or with deep networks that predict a partial assignment (Sarlin et al., 2020; Lindenberger et al., 2023; Nordström et al., 2026). These approaches are efficient and naturally yield the discrete tracks required for multi-view geometry, but since detection is independent per image, their accuracy is upper-bounded by the repeatability and localization noise of the detector. Moreover, their attention aggregates context only at the keypoint locations, which limits the global understanding of the scene, in particular in textureless or repetitive regions. In contrast, FlexGlue aggregates context over a dense grid independently of the keypoints and uses them only as queries, retaining the efficiency and the discrete tracks of sparse matchers.

Dense and semi-dense matching circumvents the repeatability bottleneck by estimating correspondences directly on image grids (Edstedt et al., 2023a;c; 2026; Zhang et al., 2025; Leroy et al., 2024). Dense matchers achieve impressive robustness and accuracy by refining a correspondence for every pixel in a coarse-to-fine scheme, whose cost grows with the image resolution. Semi-dense matchers (Sun et al., 2021; Wang et al., 2024; Li et al., 2025) reduce this cost with a coarse assignment on a low-resolution grid, but quantize correspondences to its cells, which complicates the extraction of discrete multi-view tracks (He et al., 2024). In contrast, our sparse-to-dense formulation aggregates context on a coarse grid but refines correspondences only at a requested set of sparse queries, such that its high-resolution stage scales linearly with their number and naturally yields discrete tracks.

Sparse-to-dense matching bypasses keypoint detection in the target image by matching sparse queries against its dense features. S2DNet (Germain et al., 2020) casts this as a dense classification over CNN hypercolumns, and CAPS (Wang et al., 2020a) uses it to learn descriptors from camera poses only. Both rely on local receptive fields and lack the global context needed to resolve ambiguities. RCM (Lu & Du, 2024) follows the design of semi-dense matchers but replaces the coarse grid of one view, chosen per pair, by sparse keypoints, which limits its context in that view to the keypoints. Differently, FlexGlue aggregates context over the full dense grids of both images.

Visual disambiguation is critical to reject non-covisible pairs and visual symmetries, which otherwise cause catastrophic failures in 3D reconstruction. Pipelines shortlist pairs with global image retrieval (Arandjelovic et al., 2016; Izquierdo & Civera, 2024; Berton & Masone, 2025) and verify them by matching and view-graph reasoning (Heinly et al., 2014), which fails on subtle symmetries, so-called doppelgangers, since learned matchers are trained on covisible pairs only. Dedicated models (Cai et al., 2023; Xiangli et al., 2025) predict the covisibility of a pair after matching, but add computational overhead, generalize poorly beyond their training data, and are evaluated on outdoor landmarks only. In contrast, FlexGlue predicts point-level matches and pair-level covisi-

bility within a single model, benefitting from joint estimation and, following the adaptive depth of LightGlue (Lindenberger et al., 2023), exits early on unmatchable pairs.

3 METHOD

3.1 SPARSE-TO-DENSE MATCHING

Problem formulation: Given two images $\mathbf{I}_a, \mathbf{I}_b \in \mathbb{R}^{3\times H\times W}$ and their corresponding N and M keypoints, with coordinates $\mathcal{K}_a \in \mathbb{R}^{N\times 2}$ and $\mathcal{K}_b \in \mathbb{R}^{M\times 2}$, we aim to estimate two sets of sparse-to-dense correspondences. Concretely, for each keypoint in $\mathbf{I}_a$, FlexGlue predicts its corresponding 2D coordinates $\mathbf{p}_{a\to b} \in \mathbb{R}^2$ in $\mathbf{I}_b$, and vice versa, $\mathbf{p}_{b\to a}$. FlexGlue also predicts whether each point is matchable – a semantic signal indicating whether the point is visible in the other image. From the estimated correspondences, we derive point-to-point assignments between keypoints via geometric matching based on their distances in the images. Visual disambiguation is done with a binary global classification of whether the two images observe the same scene, in spite of *e.g.*, visual symmetries.

Overview: As illustrated in Fig. 2, FlexGlue aggregates context over dense image features rather than sparse keypoints, avoiding both information gaps between points and the quadratic cost of attention over large keypoint sets. A frozen pretrained CNN extracts features from each image, which we fuse across layers into a coarse grid of tokens. A stack of L identical layers updates these tokens with self- and cross-attention, while a covisibility head decides at each layer whether the two images overlap and halts inference early on non-overlapping pairs. A lightweight head then reads out, for each query keypoint, a correspondence $\mathbf{p}_{a\to b}$ with confidence c and matchability m, which a second head refines to subpixel accuracy. For multi-view tracks, we associate queries with keypoints of the other image via mutual nearest neighbors.

3.2 COARSE MATCHING

Context aggregation: We first extract dense features from each image with a ConvNeXt (Liu et al., 2022) pretrained with DINOv3 (Siméoni et al., 2025), whose weights we keep frozen. As observed by RoMa (Edstedt et al., 2023c; 2026), this reduces overfitting to the training data and improves robustness, while being cheaper to train. To combine fine-grained local structure with higher-level semantics, we resample all four backbone stages to stride 8 and linearly project them into initial token grids $\mathbf{T}_a^0, \mathbf{T}_b^0 \in \mathbb{R}^{P\times D}$ at normalized coordinates $\mathbf{g} \in \mathbb{R}^{P\times 2}$, with $P{=}\frac{HW}{8^2}$. Differently from sparse matchers (Lindenberger et al., 2023; Nordström et al., 2026), which sample features at the keypoints and thereby discard information between them, we retain the full token grid and interpolate keypoint descriptors only after context aggregation. We then update all tokens using L Transformer layers $\mathbf{T}_a^l, \mathbf{T}_b^l = \mathcal{T}^l(\mathbf{T}_a^{l-1}, \mathbf{T}_b^{l-1})$ that aggregate context within each image through self-attention and across images through cross-attention. In self-attention, we encode the relative positions of the token coordinates $\mathbf{g}$ with learned rotary embeddings (Su et al., 2021), following LightGlue (Lindenberger et al., 2023). Cross-attention uses no positional encoding. At stride 8, these tokens aggregate distinctive context from the entire image pair, but are too coarse for subpixel-accurate matching, which we recover through the correspondence readout and local refinement described next.

Correspondence estimation: For a query keypoint $\mathbf{k}_a^i \in \mathcal{K}_a$, we interpolate ($[\cdot]$) the token grid and down-project it into a descriptor $\mathbf{q}_a^i{=}W\mathbf{T}_a^L[\mathbf{k}_a^i]$ with a linear layer W, which we share with the other image to reduce the cost of the dense correlation and keep matching symmetric. Given its cosine similarities $s_{ij}{=}\cos(\mathbf{q}_a^i, \mathbf{t}_b^j)$ to all tokens of the other image, we estimate the correspondence $\mathbf{p}_{a\to b}^i$ from a global softmax, renormalized within the $3{\times}3$ neighborhood $\mathcal{N}(j^*)$ of the nearest-neighbor cell $j^*{=}\operatorname{argmax}_j s_{ij}$:

$$P_{ij} = \operatorname{softmax}_j\Big(\frac{s_{ij}}{\tau}\Big), \qquad \mathbf{p}_{a\to b}^i = \frac{\sum_{j\in\mathcal{N}(j^*)} P_{ij}\,\mathbf{g}_b^j}{\sum_{j\in\mathcal{N}(j^*)} P_{ij}}\ , \tag{1}$$

where $\mathbf{t}_b^j$ is the j-th token of $W\mathbf{T}_b^L$ with cell center $\mathbf{g}_b^j$, τ is a learned temperature, and $\cos(\mathbf{x},\mathbf{y}){=}\mathbf{x}^\top\mathbf{y}/\|\mathbf{x}\|_2\|\mathbf{y}\|_2$ is the cosine similarity. Correspondences $\mathbf{p}_{b\to a}^j$ for keypoints in $\mathbf{I}_b$ are obtained symmetrically. Empirically, this windowed readout improves accuracy at depth discontinuities, where a global expectation averages between competing candidates, while avoiding the cell

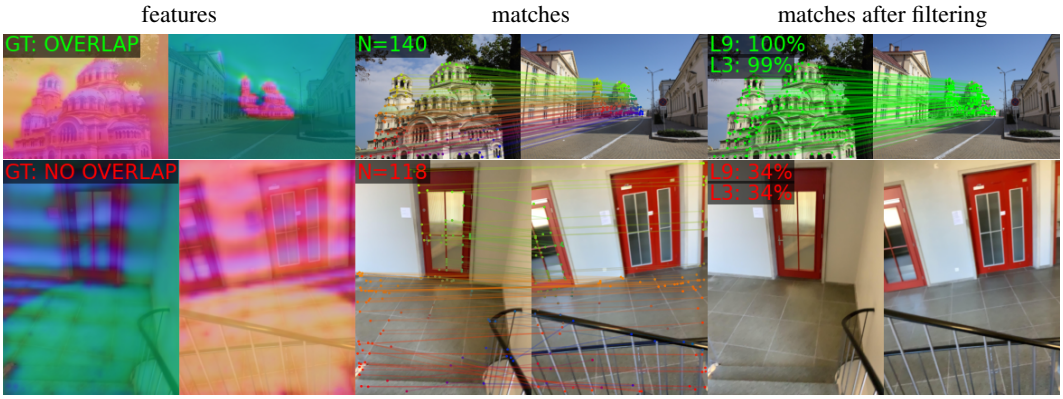


Figure 3: Visual disambiguation examples. **Left:** Images overlayed by PCA of learned features. **Center:** Estimated correspondences pre-filtering . **Right:** We report the covisibility score after 3 and 9 layers in the top-left. **Top:** Our method correctly identifies a covisible pair. **Bottom:** We correctly identify challenging doppelgangers from various domains early and stop matching.

discretization of nearest-neighbor matching at negligible cost. We supervise P_{ij} with a cross-entropy against the bilinear interpolation weights of the ground-truth correspondence over its four surrounding cells (Section 3.5). Its optimum is reached when P_{ij} equals these weights, for which Eq. (1) recovers the ground-truth position exactly, such that the network learns to interpolate between cells from context and reaches sub-cell accuracy . Unlike semi-dense matchers (Sun et al., 2021; Li et al., 2025), we thus avoid quantizing correspondences to cells, and unlike sparse matchers (Sarlin et al., 2020; Lindenberger et al., 2023), our targets are derived from the exact corresponding pixel rather than a hard assignment between discrete keypoints.

Matchability: We also predict a matchability score per query from the interpolated token $\mathbf{q}_a^i$ via convex upsampling (Teed & Deng, 2020) and a lightweight classification head. This score indicates whether the point is visible in the other view and accounts for occlusions and limits of the field-of-view. Compared to sparse matchers, the uniform token grid is not biased by the local feature density and removes the need to learn repeatability, an artifact of local feature detectors.

3.3 Local Refinement

Since the coarse readout already places well above 80% of queries within 2 px of their ground truth, the remaining error is local. Instead of a separate fine-scale CNN as in dense matchers (Edstedt et al., 2023b;c; 2026), we follow EDM (Li et al., 2025) and predict a sub-cell offset $\mathbf{\Delta}^i$ with a small MLP from the tokens of the current layer, interpolated at $\mathbf{k}_a^i$ and $\mathbf{p}_{a\to b}^i$ and concatenated. As in EDM, the MLP classifies the offset per axis over bins spanning ± 1 cell, and we take the expectation, $\mathbf{p}_{a\to b}^i \leftarrow \mathbf{p}_{a\to b}^i + \mathbf{\Delta}^i$. This requires only two interpolations and an MLP per query and thus scales linearly with the number of queries. We stop the gradient through the coarse position $\mathbf{p}_{a\to b}^i$, which is already supervised by $\mathcal{L}_{\text{coarse}}$, and supervise the refined position with a robust loss (Section 3.5). In practice, it shifts correspondences by only 0.2 px on average, confirming that it corrects residual sub-pixel errors while leaving the coarse matches intact.

3.4 Visual Disambiguation

Rationale: Maximizing correspondence accuracy is not sufficient in practice: non-overlapping but visually similar pairs (Fig. 3), like opposite facades of a symmetric building, yield spurious correspondences that corrupt the view graph and can collapse downstream 3D reconstruction. Prior work addresses this as a separate post-processing step, conditioned on pre-extracted correspondences (Cai et al., 2023) or on a pretrained two-view geometry model (Xiangli et al., 2025). We instead integrate disambiguation directly into the matcher, motivated by the observation of LightGlue (Lindenberger et al., 2023) that deep matchers identify covisibility already in early layers, which we confirm for our model (Fig. 5).

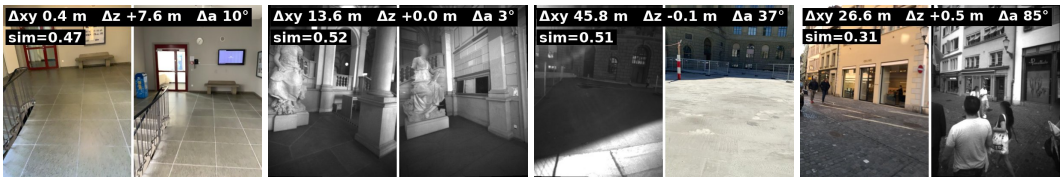


Figure 4: **Hard negatives of our LaMAR doppelganger benchmark.** From left to right: different floors (CAB), indoor (HGE), outdoor (HGE), and cross-device (LIN) pairs. None of the pairs overlap, despite a high retrieval similarity (sim) from MegaLoc; Δxy, Δz, and Δa denote the horizontal and vertical distance and the viewing angle difference between the two cameras.

Mechanism: A linear layer W_o followed by a sigmoid σ maps each token of $\mathbf{T}_a^L$ and $\mathbf{T}_b^L$ to a covisibility vote $o_j^a, o_j^b \in [0, 1]$, which we average into the pair score $\bar{o} = \frac{1}{2P}\sum_j (o_j^a + o_j^b)$. Voting per token rather than predicting from a single pooled feature provides dense supervision and requires all tokens to agree on the pair label. We supervise the score at every layer, such that at inference, pairs with $\bar{o}$ below a threshold exit early, avoiding computation on unmatchable pairs.

3.5 Supervision

During training, we sample one query per 16×16 cell of a uniform grid, displaced by a random sub-cell offset, rather than using detected keypoints. Since the network is thus never tied to a specific detector, the query keypoints can be freely exchanged at test time. The coarse loss is the cross-entropy of P_{ij} against the bilinear weights w_k of the ground-truth correspondence over its four surrounding cells, $\mathcal{L}_{\text{coarse}} = -\sum_k w_k \log P_{ik}$. Its optimum is $P_{ik}{=}w_k$, for which Eq. (1) recovers the ground-truth position exactly. Interpolating in log space is crucial, as minimizing $-\log \sum_k w_k P_{ik}$ instead concentrates all mass on the closest cell and loses sub-cell supervision. The refinement is supervised with the generalized Charbonnier loss (Barron, 2019; Edstedt et al., 2023c). We only apply it to targets within the ±1-cell range of the offset. The matchability is supervised per query point with a balanced binary cross-entropy loss $\mathcal{L}_{\text{match}}$ for which the label indicates whether the point is visible in the other image. Similarly, we supervise the per-token covisibility votes with a balanced binary cross-entropy loss $\mathcal{L}_{\text{covis}}$ given the ground-truth pair label. The total loss at each layer is $\mathcal{L} = 10\,\mathcal{L}_{\text{coarse}} + \mathcal{L}_{\text{fine}} + \mathcal{L}_{\text{match}} + \mathcal{L}_{\text{covis}}$, and we apply deep supervision across all L layers. For sparse-to-sparse applications like SfM, we perform 2D nearest neighbor search with predicted correspondences. More details are in the supplementary material.

3.6 Indoor Doppelgangers Benchmark

Motivation: Existing doppelganger benchmarks (Cai et al., 2023; Xiangli et al., 2025) consist exclusively of outdoor scenes. Indoor environments, however, exhibit their own, particularly hard failure cases: different floors or hallways of the same building often look nearly identical, while repetitive textures, low visual diversity, challenging lighting, and small scene depth further limit the cues available for disambiguation. To evaluate this so far unexplored setting, we introduce an indoor doppelganger benchmark.

Labels: We build the benchmark on three large buildings of LaMAR (Sarlin et al., 2022) (CAB, HGE, LIN), which provides accurate ground-truth poses and meshes. We compute the overlap of each image pair by reprojecting depth rendered from the meshes with the ground-truth poses, and label a pair as positive if its overlap is non-zero and as negative otherwise. Randomly sampled non-overlapping pairs are trivially rejected. We therefore mine hard negatives, Fig. 4, with the off-the-shelf image retrieval model MegaLoc (Berton & Masone, 2025), selecting visually similar yet non-overlapping pairs. This mirrors practical SfM pipelines, which shortlist candidate pairs by retrieval before matching . As positives, we include both high-similarity pairs, such that the retrieval score alone does not separate the classes, and pairs with very low overlap, which are critical to retain for loop closure. Each scene contains 1000 pairs, 500 positive and 500 negative.

Availability of training data: Collecting indoor hard negatives at scale is difficult: unlike internet photo collections, indoor images carry no GPS tags that would reveal visually similar but distinct places. Consequently, all existing training sets for visual disambiguation are outdoor, and we evaluate all methods zero-shot to measure how well they generalize across this domain gap. As shown in Ta-

ble 2, all methods drop substantially compared to the pure outdoor benchmarks. In the supplementary (Table 7) we build a small training dataset from Naverlabs Lee et al. (2021) and show that this helps in-domain generalization.

4 EXPERIMENTS

4.1 IMPLEMENTATION DETAILS

We train FlexGlue on eight datasets: MegaDepth (Li & Snavely, 2018), BlendedMVS (Yao et al., 2020), ScanNet++ (Yeshwanth et al., 2023), the TartanAir (Wang et al., 2020b) split of MapAnything (Keetha et al., 2026), HyperSim (Roberts et al., 2021), MapFree (Arnold et al., 2022), Aerial MegaDepth (Vuong et al., 2025) and DL3DV (Ling et al., 2024) with balanced sampling over scenes. For all datasets we sample pairs with dense visual overlap, normalized by area with depth coverage, in $[0.05, 0.8]$. Images are center-padded and resized to 512×512 px. We train for 50 epochs with 150k samples per epoch, a batch size of 16 per device, and mixed precision on 4 H200 GPUs ($\sim$4 GPU-days). With gradient checkpointing the same model can be trained on 4 RTX 4090 in about 1.5 days. We use a learning rate of $2 \cdot 10^{-4}$, held constant for 5 epochs, then decayed to $2 \cdot 10^{-5}$ over the remaining 45 epochs. The matching temperature is initialized to 10 and learned independently per layer. Our transformer has an internal dimension of 512 dimensions and 9 layers. Light photometric augmentations are applied throughout training. In the last 10 epochs, we reset the learning rate to 5^{-5}, increase the resolution to and mix in doppelganger Cai et al. (2023) pairs (50% positive, 50% negative) and flip augmentations to generate additional hard negatives.

4.2 RELATIVE POSE ESTIMATION

Setup: We evaluate image matching via calibrated relative pose estimation on two established benchmarks: MegaDepth-1500 (Sun et al., 2021), consisting of outdoor phototourism image pairs, and ScanNet-1500 (Dai et al., 2017; Sarlin et al., 2020), an indoor benchmark with challenging self-similarities and textureless regions. We further evaluate all methods on the challenging HardMatch benchmark (Nordström et al., 2026) which contains image pairs under extreme appearance variations. We provide additional details in the supplementary materials.

Results: As reported in Table 1, in the sparse setting, FlexGlue performs similar to LoMA (Nordström et al., 2026). This large improvement on indoor scenes – where sparse matchers traditionally struggle due to repetitive textures and limited visual diversity – demonstrates that grid-based context aggregation is more effective there than between sparse keypoints. In the sparse-to-dense setting, our method performs competitively with RoMAv2 while being substantially faster. Notably, dense correspondence estimation consistently improves over the sparse-to-sparse variant on relative pose estimation, but lacks robustness on HardMatch. Sparse-to-Dense methods take the middle-ground, and our method achieves highest accuracy on HardMatch.

4.3 VISUAL DISAMBIGUATION

Setup: We consider three benchmarks: Doppelgangers (Cai et al., 2023), which contains phototourism image pairs that often require analyzing fine-grained structural details, ViSym (Xiangli et al., 2025), a collection of phone-captured labelled street scene pairs, and our LaMAR-DG benchmark. All methods are trained on annotated pairs from doppelgangers and flip-augmented images. Evaluation on ViSym (Xiangli et al., 2025) and LaMAR (Sarlin et al., 2022) is thus zero-shot. We report AUROC, average precision (AP) and run time per method, and compare against Doppelgangers Cai et al. (2023) and DG++ (Xiangli et al., 2025). Further details in appendix.

Results: As reported in Table 2, FlexGlue achieves the highest accuracy on ViSym, outperforming DG++ while being over $5\times$ faster. On the in-distribution Doppelgangers benchmark, FlexGlue performs on par with DG++. On our challenging LaMAR (Sarlin et al., 2022) dataset, FlexGlue significantly outperforms SotA methods, especially on the extremely challenging and out-of-distribution LaMAR benchmark. Notably, our method outperforms literature even without any manually annotated doppelganger training pairs, suggesting that visual disambiguation is emergent in our model (more details in supplementary, Table 7).

Table 1: **Relative pose estimation on MegaDepth-1500, ScanNet-1500 and HardMatch.** We report the AUC of the pose error at 5°, 10°, and 20°, the HardMatch mAA@10px, and the extraction and matching time; the first, second, and third best results within each group are highlighted (lower is better for time). Our method achieves comparable accuracy with dense feature matchers at a fraction of the cost. Notably, our model also improves sparse-to-sparse feature matching on two datasets.

	Matching	MegaDepth-1500			ScanNet-1500			HardMatch	Extr.	Match
		5°	10°	20°	5°	10°	20°	mAA	[ms]	[ms]
Sparse 4096 kp.	ALIKED+LightGlue	52.4	68.8	81.0	17.0	34.2	51.3	36.3	36.0	**12.7**
	DaD+LoMA-G	**56.1**	**72.2**	**84.0**	29.3	51.7	70.0	**54.3**	128.5	21.4
	SP+**Ours**	54.2	71.1	83.3	**30.0**	**52.4**	**71.0**	54.0	**33.1**	27.4
Dense –	EDM	57.5	73.2	84.2	19.8	37.5	54.4	33.1	**5.0**	**16.0**
	RoMA	62.6	76.7	86.3	31.8	53.4	70.9	**48.1**	48.2	73.1
	UFM	41.5	57.9	72.4	31.3	54.1	72.0	33.9	35.7	37.4
	RoMAv2	**62.8**	**77.0**	**86.6**	**33.6**	**56.2**	**73.8**	46.5	42.4	73.1
Sparse-to-Dense 2048 keypoints	SP+CAPS	34.8	52.2	67.5	9.4	21.9	37.1	-	-	-
	SP+S2DNet	35.0	51.9	64.2	10.3	21.6	34.5	-	-	-
	SP+RCM	53.2	69.4	81.5	17.3	34.6	52.1	-	-	-
	SP+RoMAv2	57.1	73.1	84.1	33.5	56.0	73.3	49.0	53.4	63.2
	SP+**Ours**	56.4	72.9	84.0	**33.8**	56.1	73.7	54.9	**31.0**	**26.7**
	+RoMAv2 refine	**60.4**	**75.2**	**85.5**	33.7	**56.2**	**73.8**	**55.6**	**31.0**	46.8

Table 2: **Detection of Doppelgangers.** We report AUROC and AP on DG (in-distribution), ViSym and LaMAR, as well as the matching time. FlexGlue achieves highest disambiguation accuracy across all datasets trained on same labelled data (DG), especially $+10\%$ AUROC on the LaMAR benchmark, at much higher throughput.

Method	DG		ViSym (zero-shot)		LaMAR (zero-shot)		Time
	AUROC	AP	AUROC	AP	AUROC	AP	[ms]
DG Cai et al. (2023)	94.4	95.4	72.6	81.6	69.5	69.8	139.5
DG++ Xiangli et al. (2025)	98.1	98.0	90.9	93.6	78.6	79.4	124.7
Ours	**98.4**	**98.2**	**94.5**	**95.1**	**88.8**	**89.2**	**26.7**
Ours (label free)	97.0	97.1	92.0	92.5	88.5	89.0	**26.7**

4.4 Multi-view estimation

Setup: We evaluate end-to-end SfM on 25 scenes of the ETH3D dataset (Schops et al., 2017), spanning both indoor and outdoor environments. The main challenges of this dataset are low-texture environments, self-similarities, and symmetries, which are classical failure cases of SfM. This benchmark is joint test of both matching accuracy and visual disambiguation. We report the AUC of relative pose errors within each scene, averaged over all scenes, as well as the matching throughput as measured by hloc (Sarlin et al., 2019). We also evaluate noisy 3D reconstructions on the CVPR 2025 image matching challenge (Bellavia et al., 2025b). This benchmark provides random collections of images, and requires grouping and accurately reconstructing multiple clusters in one scene. We run all methods on the hidden test set on kaggle (Bellavia et al., 2025a), and report the private and public score there.

Results: As shown in Table 3, even without visual disambiguation, our method (SP+Ours) outperforms all other methods. All deep feature matchers break under symmetries, including dense RoMA. Pairing image matchers with visual disambiguation significantly increases the robustness of all methods, but at typically lower throughput. Notably, early stopping does not impair accuracy while increasing throughput further, avoiding wasteful computation on unmatchable pairs, and still outperforms the combined SP+RoMA+DG++ baseline by a comfortable margin at a fraction of its runtime. These results support the central claim that robust multi-view reconstruction requires both high matching accuracy and effective visual disambiguation, which our model provides jointly.

On IMC2025 (Table 4), our method outperforms the heavily tuned 2025 winning MASt3R (Duisterhof et al., 2025) entry with 4096 keypoints only. With visual disambiguation we improve 5 points over matching, and 7 points over LoMA (Nordström et al., 2026). On the ongoing benchmark (Bellavia et al., 2025a) we achieve a score of 66.41815, which beats the top leaderboard entry (65.55643) Bellavia et al. (2025a) without extensive tuning.

Table 3: **Uncalibrated end-to-end SfM.** On 25 scenes from ETH3D Schops et al. (2017), our method achieves higher matching accuracy than baselines, resulting in higher pose accuracy. Disambiguation further improves pose estimation by reducing symmetry artifacts.

Matching	Filter	Pose AUC			Throughput
		1°	3°	5°	[1/s]
SP+LG	-	29.5	51.8	58.5	**40**
SP+RoMAv2	-	56.4	75.7	80.7	11
LoMA-G	-	58.3	77.6	82.3	37
Ours	-	**60.1**	**78.9**	**83.5**	30
MASt3R-SfM	DG++	54.1	77.9	81.5	5
SP+RoMAv2	DG++	62.0	80.6	85.1	4
LoMA-G	DG++	62.4	81.2	85.8	5
Ours	Ours	**68.9**	**86.8**	**92.3**	30
Ours	Ours (early)	67.5	86.0	91.6	**45**

Table 4: **In-the-wild SfM with outliers.** On IMC2025 (Bellavia et al., 2025b), our method yields more accurate reconstructions and the disambiguation significantly increases robustness.

Method	Filter	Private Score	Public Score
MASt3R*	-	56.00	52.64
LoMA-B	-	47.54	42.98
LoMA-G	-	43.82	41.31
Ours	-	49.40	45.46
Ours	Ours	**55.20**	**49.66**
LoMA-B (4k)	-	55.89	56.34
Ours (4k)	Ours	**63.83**	**61.86**

Table 5: **Visual localization on InLoc.** We report the percentage of correctly localized queries at $(0.25\text{m}, 10°)$, $(0.5\text{m}, 10°)$, and $(1.0\text{m}, 10°)$, and the matching throughput. Our method achievest highest accuracy with high efficiency.

Method	DUC1			DUC2			matching
	0.25m	0.5m	1m	0.25m	0.5m	1m	[1/s]
SP+SG	47.0	69.2	79.8	53.4	77.1	80.9	25
SP+LG	49.0	68.2	79.3	55.0	74.8	79.4	35
EDM	51.5	72.7	85.9	59.5	82.4	88.5	**40**
MASt3R	56.1	79.3	90.9	71.0	87.0	91.6	5
RoMA	60.6	79.3	89.9	66.4	83.2	87.8	4
RoMAv2	**67.7**	**84.3**	**94.4**	73.3	86.3	90.8	8
LoMA-G	55.6	80.3	91.4	73.3	87.8	89.3	35
Ours	62.6	**84.3**	**94.4**	**77.9**	**88.5**	**94.7**	32

Table 6: **Matching ablation.** HardMatch measures robustness, PCK@1 px precision. The refinement head improves sub-pixel precision.

Component	HardMatch mAA@10px	EPE PCK@1px	Time [ms]
Ours	54.9	60.5	26.7
- 3 layers	46.5	50.6	**14.6**
- no refinement head	53.0	58.6	25.0
- windowed soft-argmax	51.2	58.3	24.8
- disambiguation finetune	52.5	60.3	26.7
+ unfrozen backbone	47.5	61.0	26.7
stride 16 + convex up	42.0	44.2	20.4
w/ RoMav2 refinement	**55.5**	**64.5**	46.8

4.5 Visual Localization

We evaluate indoor visual localization on InLoc (Taira et al., 2019), where the goal is to estimate the 6-DoF absolute pose of a query image given a set of database images with known poses and dense depth maps. We follow the hloc (Sarlin et al., 2019) protocol. Further setup details can be found in supplementary. Table 5 shows that semi-dense and dense matchers consistently outperform sparse methods, as they are not limited by keypoint repeatability. Our method is more robust and precise than dense matchers, achieving SotA on 5 / 6 thresholds, at much higher throughput.

4.6 Ablations

Matching: Table 6 removes or replaces one component at a time, with SuperPoint (DeTone et al., 2018) queries. Stride 8 is the most important choice: running at stride 16 with convex upsampling (Teed & Deng, 2020) costs over 12 points on both metrics at little speed-up. The windowed soft-argmax improves robustness, while the refinement head improves both metrics at negligible cost. Hard negatives improve match filtering. Fine-tuning the backbone slightly improves precision but hurts generalization. Stopping after 3 layers incurs a moderate loss in accuracy (more detailed in supplementary, Fig. 5). Dense RoMa v2 (Edstedt et al., 2026) refinement further improves precision.

5 Conclusion

We introduced a deep matching architecture to obtain sparse-to-dense correspondences which bridges the efficiency of sparse matchers and robustness of dense matchers. By combining uniform attention over a grid with well localized query points and visual disambiguation, our method achieves the highest accuracy on two multi-view reconstruction benchmarks (ETH3D, IMC2025). We further improve over literature on all disambiguation benchmarks, especially the proposed indoor LaMAR dataset, on InLoc visual localization, and the challenging two-view HardMatch benchmark, all while maintaining competitive computational efficiency.

REFERENCES

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.

Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free Visual Relocalization: Metric Pose Relative to a Single Image. In *ECCV*, 2022.

Jonathan T Barron. A general and adaptive robust loss function. In *CVPR*, 2019.

Fabio Bellavia, Jiri Matas, Dmytro Mishkin, Luca Morelli, Fabio Remondino, Amy Tabb, Eduard Trulls, Kwang Moo Yi, and Sohier Dane. Image matching challenge 2025 ongoing. `https://www.kaggle.com/competitions/image-matching-challenge-2025-ongoing`, 2025a. Kaggle.

Fabio Bellavia, Jiri Matas, Dmytro Mishkin, Luca Morelli, Fabio Remondino, Amy Tabb, Eduard Trulls, Kwang Moo Yi, Sohier Dane, Addison Howard, and María Cruz. Image matching challenge 2025. `https://www.kaggle.com/competitions/image-matching-challenge-2025`, 2025b. Kaggle.

Gabriele Berton and Carlo Masone. Megaloc: One retrieval to place them all. In *CVPRW*, 2025.

Ruojin Cai, Joseph Tung, Qianqian Wang, Hadar Averbuch-Elor, Bharath Hariharan, and Noah Snavely. Doppelgangers: Learning to disambiguate images of similar structures. In *CVPR*, 2023.

Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. ASpanFormer: Detector-Free Image Matching with Adaptive Span Transformer. In *ECCV*, 2022.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *CVPR Workshop on Deep Learning for Visual SLAM*, 2018.

Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *2025 International Conference on 3D Vision (3DV)*, 2025.

Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023a.

Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense Kernelized Feature Matching for Geometry Estimation. In *CVPR*, 2023b.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Revisiting Robust Lossses for Dense Feature Matching. *arXiv preprint*, 2023c.

Johan Edstedt, David Nordström, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. RoMa v2: Harder Better Faster Denser Feature Matching. In *ECCV*, 2026.

Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2DNet: Learning accurate correspondences for sparse-to-dense feature matching. In *ECCV*, 2020.

Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. *CVPR*, 2024.

Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Correcting for Duplicate Scene Structure in Sparse 3D Reconstruction. In *ECCV*, 2014.

Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *CVPR*, pp. 17658–17668, 2024.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*, 2026.

Donghwan Lee, Soohyun Ryu, Suyong Yeon, Yonghan Lee, Deokhwa Kim, Cheolho Han, Yohann Cabon, Philippe Weinzaepfel, Nicolas Guérin, Gabriela Csurka, and Martin Humenberger. Large-scale Localization Datasets in Crowded Indoor Spaces. In *CVPR*, 2021.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024.

Xi Li, Tong Rao, and Cihui Pan. Edm: Efficient deep feature matching. In *CVPR*, 2025.

Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020.

Zhengqi Li and Noah Snavely. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In *CVPR*, 2018.

Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A Large-Scale Scene Dataset for Deep Learning-based 3D Vision. In *CVPR*, 2024.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *CVPR*, 2022.

David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

Xiaoyong Lu and Songlin Du. Raising the ceiling: Conflict-free local feature matching with dynamic view switching. In *ECCV*, 2024.

Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NeurIPS*, 2017.

David Nordström, Johan Edstedt, Georg Bökman, Jonathan Astermark, Anders Heyden, Viktor Larsson, Mårten Wadenbäck, Michael Felsberg, and Fredrik Kahl. LoMa: Local Feature Matching Revisited. In *ECCV*, 2026.

Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*, 2019.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *ICCV*, 2021.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*, 2020.

Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In *ECCV*, 2022.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016.

Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017.

Abhiram Shenoi, Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. RaCo: Ranking and Covariance for Practical Learned Keypoints. In *International Conference on 3D Vision*, 2026.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv:2508.10104*, 2025.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv:2104.09864*, 2021.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. *CVPR*, 2021.

Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. *IEEE TPAMI*, 2019.

Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV*, 2020.

Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. In *NeurIPS*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Khiem Vuong, Anurag Ghosh, Deva Ramanan, Srinivasa Narasimhan, and Shubham Tulsiani. AerialMegaDepth: Learning Aerial-Ground Reconstruction and View Synthesis. In *CVPR*, 2025.

Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *Proc. European Conference on Computer Vision (ECCV)*, 2020a.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A Dataset to Push the Limits of Visual SLAM. In *IROS*, 2020b.

Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient LoFTR: Semi-dense local feature matching with sparse-like speed. In *CVPR*, 2024.

Yuanbo Xiangli, Ruojin Cai, Hanyu Chen, Jeffrey Byrne, and Noah Snavely. Doppelgangers++: Improved visual disambiguation with geometric 3d features. In *CVPR*, 2025.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.

Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *ECCV*, 2016.

Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, Sebastian Scherer, and Wenshan Wang. Ufm: A simple path towards unified dense correspondence with flow. In *NeurIPS*, 2025.

Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter C. Y. Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023. doi: 10.1109/TIM.2023.3271000. URL https://arxiv.org/pdf/2304.03608.pdf.

AI USE STATEMENT

In this work, we used generative AI tools for polishing, retrieval and discovery, research ideation and execution, and to draft sections of the paper. We have not used generative AI tools for generating synthetic datasets or proving mathematical claims. and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

No ethics concerns with image matching.

REPRODUCIBILITY STATEMENT

The full code and training pipeline will be made publicly available.

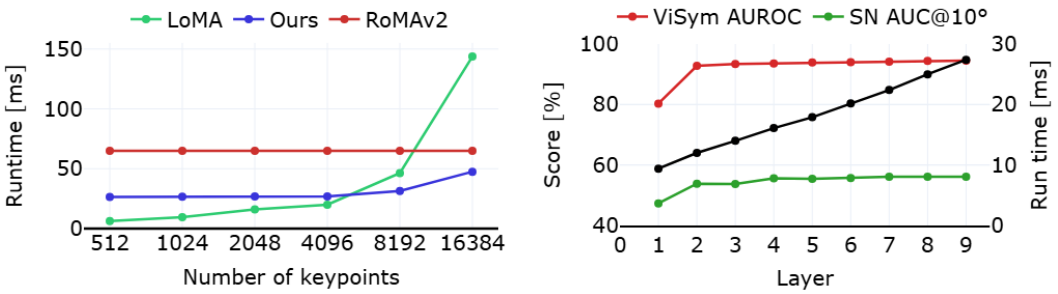


Figure 5: **Run-time on different keypoint budgets and layer ablations**. RoMAv2's Edstedt et al. (2026) performance is keypoint-independent, but has a large overhead. LoMA Nordström et al. (2026) is very efficient but scales quadratically, while our method is fairly efficient and scales linearly (left). Our method learns strong matching and disambiguation early on, while layer scale linear in run time (black line) (right).

A Appendix

We first report additional experimental results on inference time, per-layer performance, and a detailed ablation on visual disambiguation. We then show qualitative examples, and finally provide additional experimental and implementation details.

B Additional results

B.1 Inference time & layer ablation

Fig. 5 compares the matching time as a function of the number of keypoints. All methods use mixed precision, and we measure run time on a single RTX 4090. Dense matchers such as RoMa v2 (Edstedt et al., 2026) have a constant but large cost Sparse matchers such as LightGlue (Lindenberger et al., 2023) and LoMA (Nordström et al., 2026) are fast for few keypoints, but they scale quadratically with keypoints. FlexGlue aggregates context over a fixed token grid, and only its lightweight readout scales, linearly, with the number of queries, making it faster than sparse matchers for large keypoint sets and faster than RoMAv2.

In Fig. 5, on the right, we report matching performance and visual disambiguation performance against the layer count. We observe that the model yields strong performance on both benchmarks very early in the network, supporting the early-exit on negative pairs.

B.2 Visual disambiguation ablation

Table 7 ablates the covisibility head on ViSym and LaMAR. Training for covisibility alone, without the matching losses, drops AUROC by up to 9.3 points, confirming that matching and disambiguation reinforce each other. Without the DG training data, relying only on flip augmentations and negatives from our matching datasets, our model retains most of its performance, and stopping after 3 layers costs little. Adding additional indoor negatives mined on NaverLabs (Lee et al., 2021) further improves LaMAR, and training on ViSym (Xiangli et al., 2025) improves performance in-domain.

B.3 Detailed results on LaMAR DG

In Fig. 6 we provide an addition comparison to DG++(Xiangli et al., 2025) on LaMAR, and group image pairs by difficulty (visual overlap and similarity), time (same vs. cross day), and same or cross device. Our method improves especially on challenging pairs.

C Qualitative results

In Fig. 7 we illustrate the difference between our matching modes. We show powerful visualizations that our method learned in Fig. 8. We also illustrate strong clustering of our method with COLMAP

Table 7: **Doppelgangers ablation.** AUROC on ViSym and LaMAR Disambiguation. Our model learns disambiguation early, and guidance from match supervision improves generalization. Even without labelled data our model achieves high accuracy. Training on in-domain data increases accuracy.

Method	ViSym AUROC	LaMAR AUROC
Ours	94.5	88.8
- 3 layers	94.1	87.4
- no matching loss	88.8	79.5
- DG data	92.5	88.5
+ NaverLabs	94.3	**91.5**
+ & ViSym	**99.3**	90.8
DG++ w/ ViSym	99.0	75.3

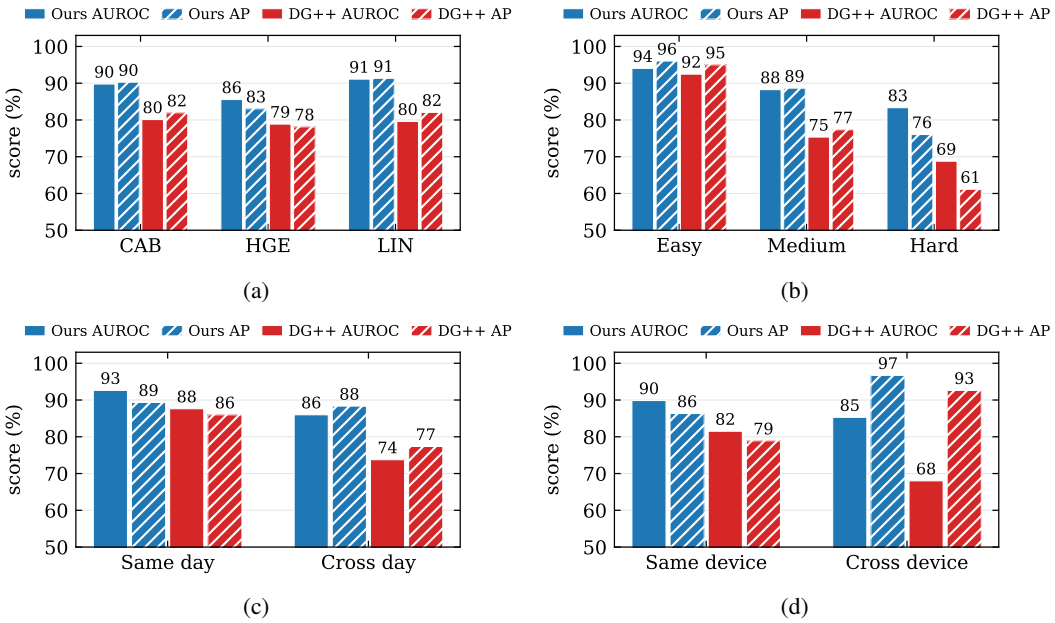


Figure 6: **LaMAR detailed results**. Our method achieves higher accuracy on challenging doppelgangers, under temporal or cross-device changes.

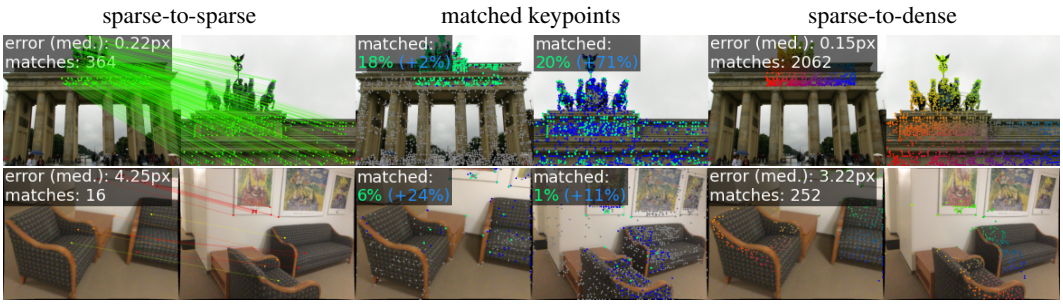


Figure 7: Comparison of matching modes. **Left:** Point-point correspondences estimated by our model, colored by epipolar error (green < 1 px). **Center:** Keypoints colored by being matchable sparse-to-sparse (green), and keypoints only matchable sparse-to-dense (blue), highlighting the repeatability issue. **Right:** Sparse-to-dense correspondences. Especially under large scale changes and on low-texture areas, sparse-to-dense matching yields many more, accurate correspondences.

on IMC2025 (Bellavia et al., 2025b) in Fig. 9. We finally show visual comparison to matches from LoMA Nordström et al. (2026) and RoMAv2 Edstedt et al. (2026).

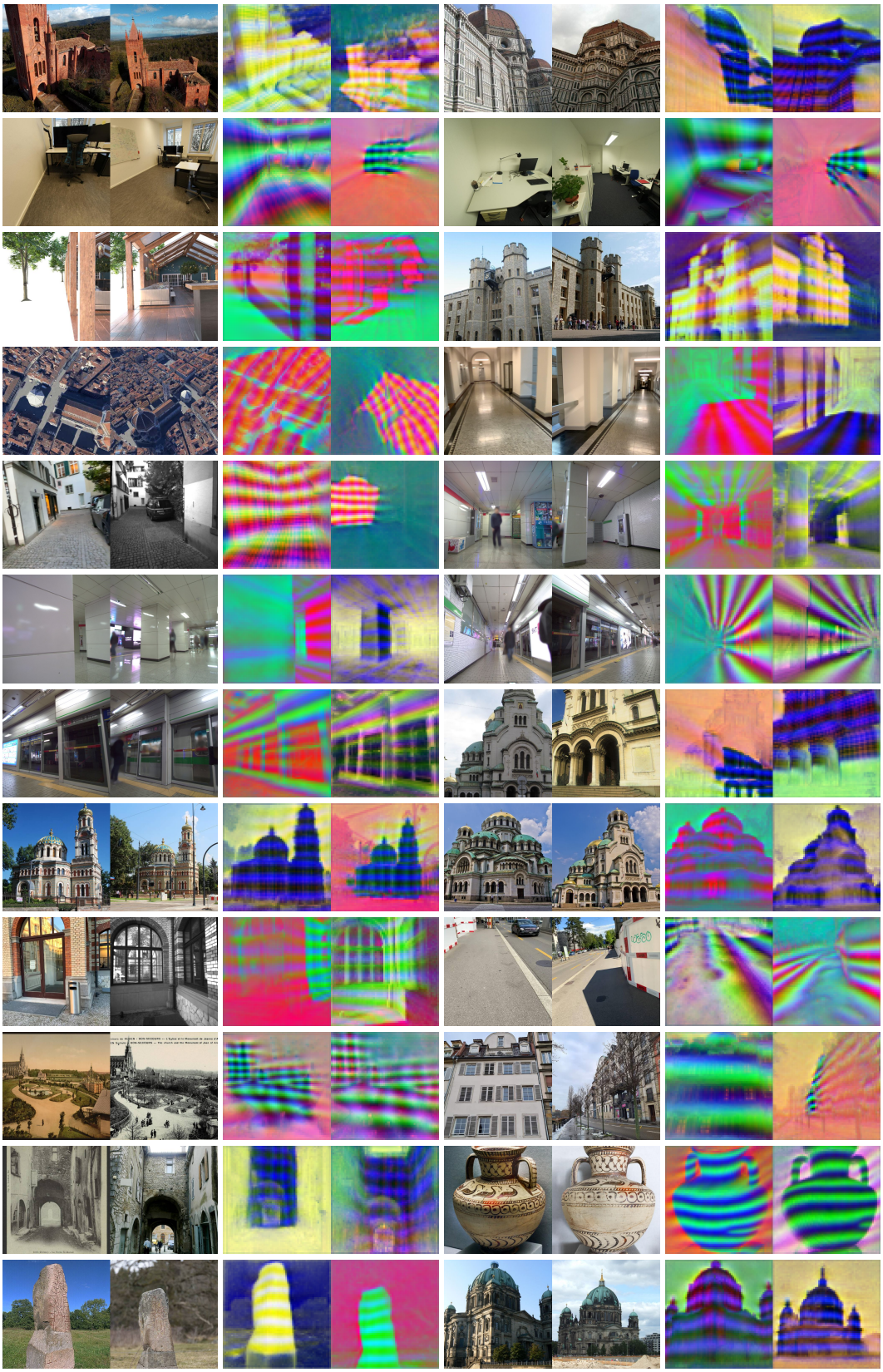


Figure 8: **PCA of the aggregated tokens.** For each image pair, we show the two views and the first three principal components of their tokens at 9 layers as RGB. Our method learns very strong structural priors like horizontal and vertical lines, camera perspective, and planarity. Covisibile areas are clearly learned.

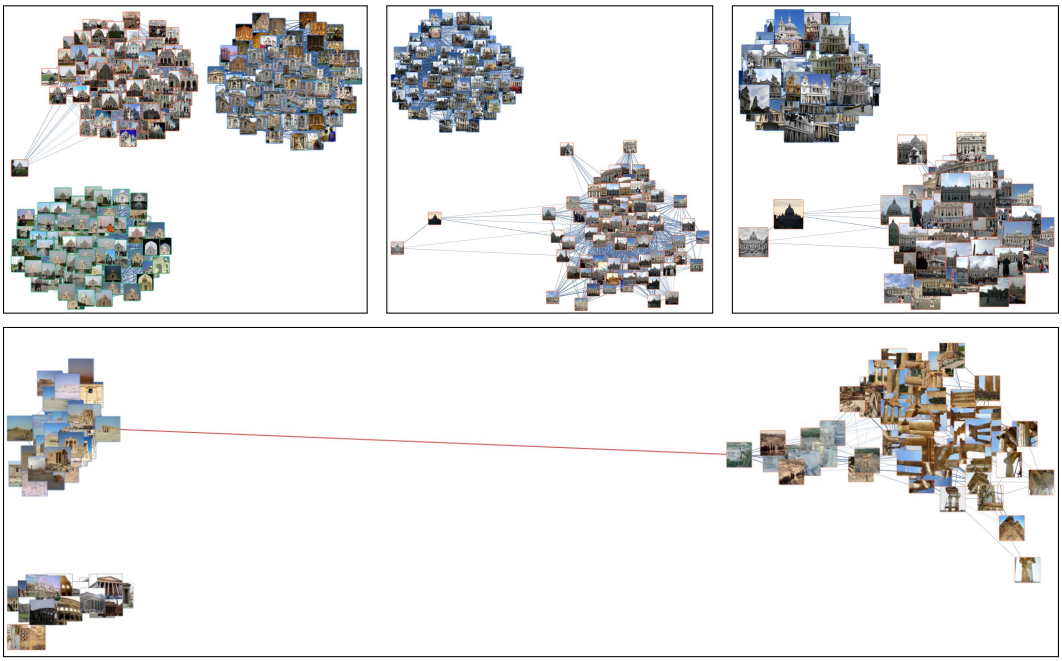


Figure 9: **Scene clustering on IMC2025.** Image graphs recovered by our method with visual disambiguation; colors denote the predicted clusters and edges the retained image pairs.

Table 8: **Training datasets.** Number of training scenes and sampling weight of each dataset. Doppelgangers is only mixed in during fine-tuning with negatives.

Dataset	Type	#Scenes	Weight [%]
MegaDepth (Li & Snavely, 2018)	outdoor, real	152	15.0
AerialMegaDepth (Vuong et al., 2025)	aerial–ground, rendered	113	15
BlendedMVS (Yao et al., 2020)	objects & scenes, rendered	493	15
ScanNet++ (Yeshwanth et al., 2023)	indoor, real	827	15
TartanAir (Wang et al., 2020b)	indoor & outdoor, synthetic	56	10
Hypersim (Roberts et al., 2021)	indoor, synthetic	399	10
Map-free (Arnold et al., 2022)	outdoor, real	460	10
DL3DV (Ling et al., 2024)	indoor & outdoor, real	9862	10
Doppelgangers (Cai et al., 2023)	outdoor, real	206	50 (stage 2)

D Implementation details

D.1 Training data

We list our full training data in Table 8, and perform light photometric augmentations for robustness. We sample pairs with dense visual overlap, normalized by area with depth coverage, in $[0.05, 0.8]$.

D.2 Supervision

The refinement is supervised with the generalized Charbonnier loss Barron (2019); Edstedt et al. (2023c): (Barron, 2019; Edstedt et al., 2023c)

$$\mathcal{L}_{\text{fine}} = \mathcal{L}_{\text{robust}}\left(\left\|\boldsymbol{\Delta}^i - \left(\mathbf{p}_i^{\text{gt}} - \text{sg}(\mathbf{p}_{a\to b}^i)\right)\right\|_2\right), \quad \mathcal{L}_{\text{robust}}(x) = \frac{|\alpha-2|}{\alpha}\left(\left(\frac{(x/c)^2}{|\alpha-2|}+1\right)^{\frac{\alpha}{2}} - 1\right), \tag{2}$$

with α=0.5 and c=0.24 px, where $\mathbf{p}_i^{\text{gt}}$ is the ground-truth correspondence and sg the stop-gradient.

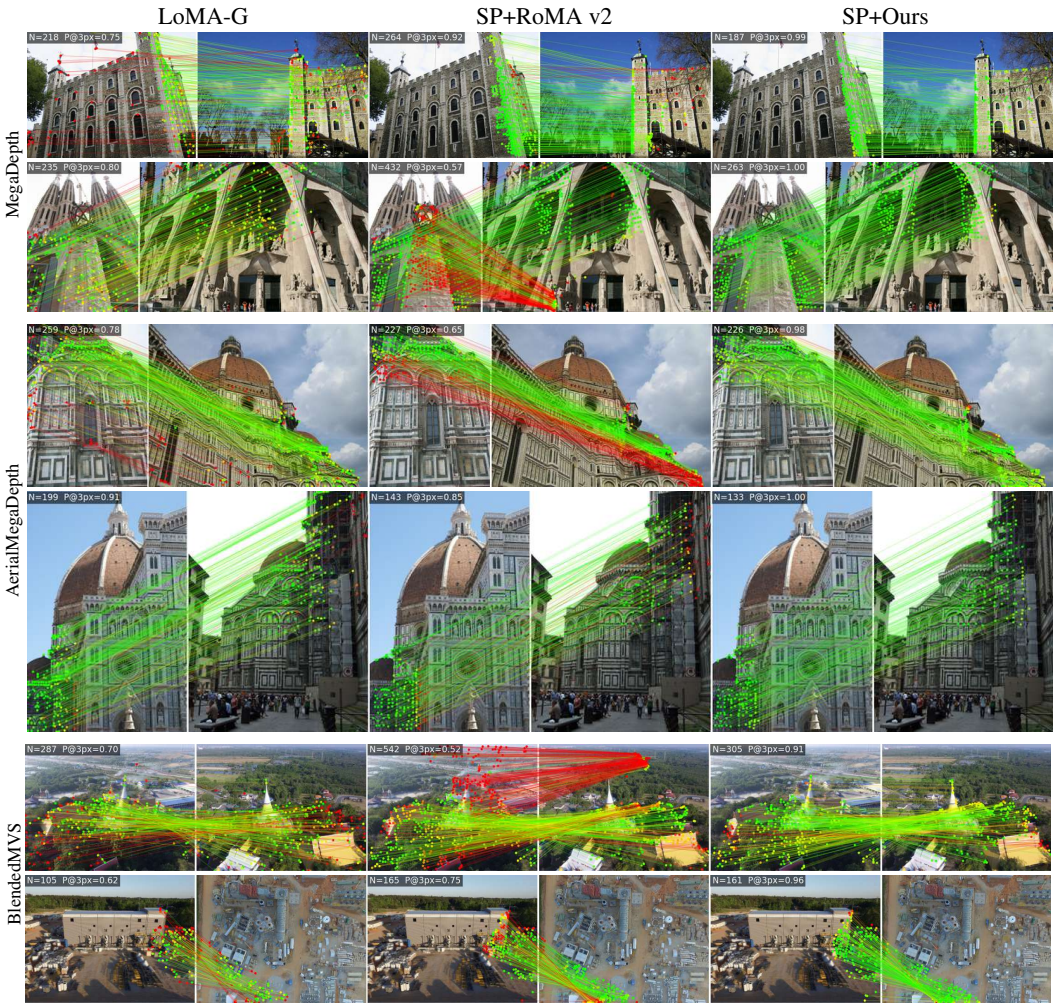


Figure 10: **Qualitative matches.** Correspondences of LoMA-G, SP+RoMA v2, and SP+Ours on two pairs per dataset, colored by their reprojection error; we report the number of matches N and the precision at 3 px. LoMA gets 2048 keypoints, and the RoMAv2 and our method get 1024 keypoints for fair comparison.

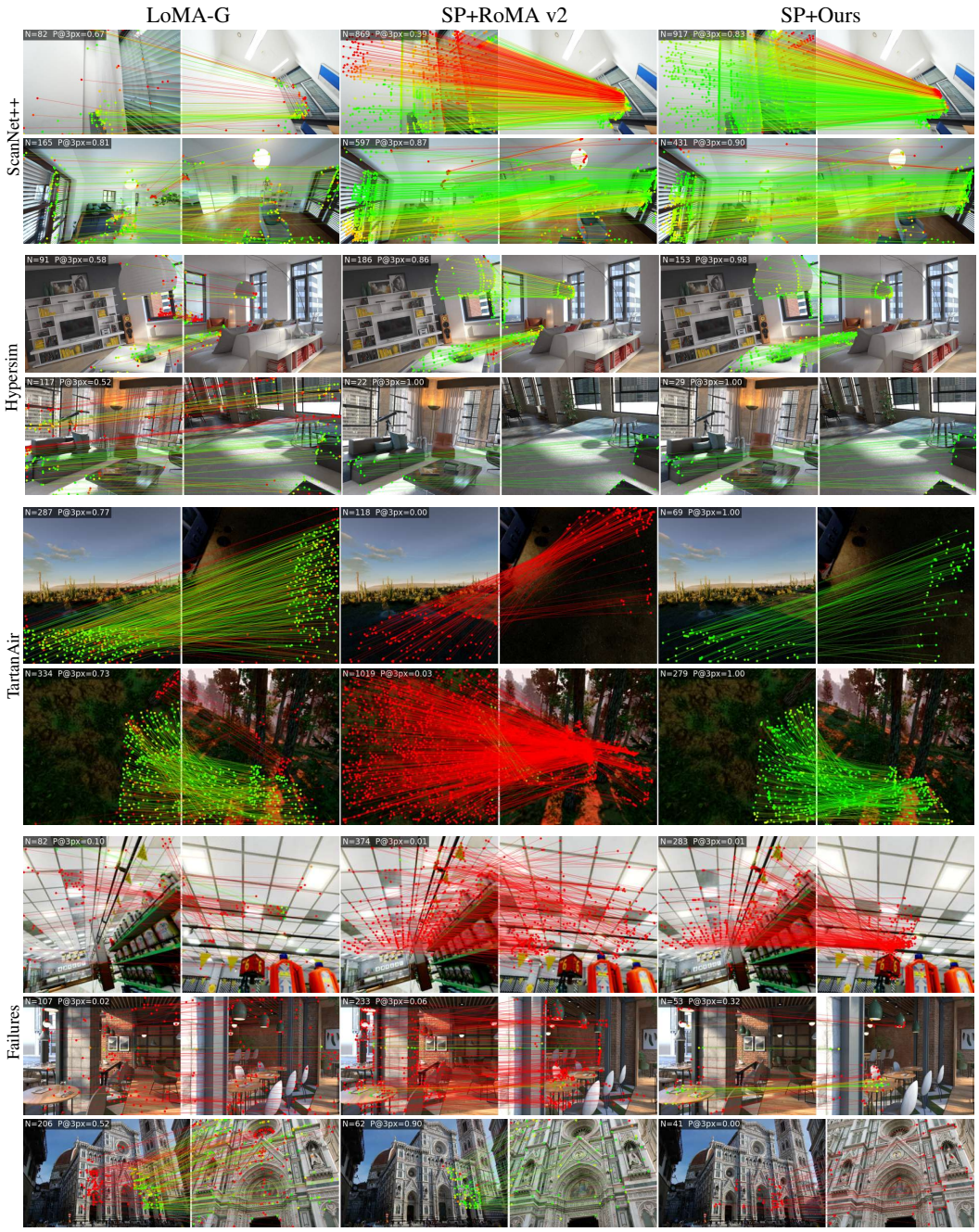


Figure 10: **Qualitative matches (continued).** Bottom: failure cases of our method on TartanAir, Hypersim, and MegaDepth.

D.3 Recovering Point-Point Correspondences

From the established sparse-to-dense correspondences, we can derive point-to-point assignments between keypoints, as required by downstream applications such as SfM. Concretely, for each keypoint $\mathbf{k}_a^i \in \mathcal{K}_a$, we compute the Euclidean distance between its predicted correspondence $\mathbf{p}_{a\to b}^i$ and all keypoints in $\mathcal{K}_b$, and analogously in the reverse direction:

$$\mathbf{D}_{a\to b}^{ij} = \|\mathbf{p}_{a\to b}^i - \mathbf{k}_b^j\|_2, \quad \mathbf{D}_{b\to a}^{ji} = \|\mathbf{p}_{b\to a}^j - \mathbf{k}_a^i\|_2. \tag{3}$$

We select pairs of keypoints that are mutual nearest neighbors.

E Implementation details

E.1 Relative Pose estimation

We follow the evaluation protocols of the respective baselines. Images in ScanNet-1500 are resized to 640×480 px, following standard practice. We compute the pose error as the maximum of the angular rotation and translation errors and report the area under the cumulative recall curve (AUC) at $5°/10°/20°$. On HardMatch, we estimate the Fundamental Matrix F from correspondences, and compute the epipolar error of the GT correspondences, and report the mean average accuracy (mAA) at 10px.

Baselines: For sparse matching, all methods detect up to 4096 keypoints. Semi-dense and dense matchers differ in their evaluation protocol, and typically estimate many more correspondences. In the sparse-to-dense category, we benchmark CAPS (Wang et al., 2020a) and S2DNet (Germain et al., 2020), two monocular baselines which perform an exhaustive search over dense features, and RCM (Lu & Du, 2024), a deep sparse-to-dense matcher. We further pair the SotA dense matcher RoMAv2 (Edstedt et al., 2026) with SuperPoint keypoints, where we interpolate the dense correspondence maps at keypoint locations. In this setting, we detect only 2048 keypoints to remain roughly comparable to sparse methods, which are limited by keypoint repeatability.

E.2 Multi-view estimation

Baselines: We report results for SuperPoint keypoints matched with LightGlue (Lindenberger et al., 2023) and RoMAv2 (Edstedt et al., 2026), LoMA-G (Nordström et al., 2026), and the symmetry-aware baseline MASt3R-SfM (Duisterhof et al., 2025) with DG++ (Xiangli et al., 2025) visual disambiguation, following the original implementation. We further strengthen baselines with Doppelgangers++ disambiguation. We run COLMAP (Schönberger & Frahm, 2016) SfM via hloc (Sarlin et al., 2019) with default settings and exhaustive pair selection, unless otherwise stated. We report results for SuperPoint keypoints matched with LightGlue (Lindenberger et al., 2023) and RoMAv2 (Edstedt et al., 2026), LoMA-G (Nordström et al., 2026), and the symmetry-aware baseline MASt3R-SfM (Duisterhof et al., 2025) with DG++ (Xiangli et al., 2025) visual disambiguation, following the original implementation. We further strengthen baselines with Doppelgangers++ disambiguation. We run COLMAP (Schönberger & Frahm, 2016) SfM via hloc (Sarlin et al., 2019) with default settings and exhaustive pair selection, unless otherwise stated.

E.3 Visual disambiguation

Setup: Visual disambiguation is closely related to view graph filtering in SfM, where ambiguous correspondences on non-covisible image pairs can lead to catastrophic failures. such as scene collapses. We consider three benchmarks: Doppelgangers (Cai et al., 2023) contains 4620 challenging phototourism image pairs that often require analyzing fine-grained structural details. ViSym (Xiangli et al., 2025) is a collection of 3180 phone-captured street scene pairs from around the globe.

Baselines: We compare against the foundational works in visual disambiguation: Doppelgangers (DG) (Cai et al., 2023), which uses a CNN encoder conditioned on local feature matches, and DG++ (Xiangli et al., 2025), which adds a lightweight decoder on top of the frozen dense matcher MASt3R (Leroy et al., 2024). We further re-train our method with only flip augmentations on the remaining training dataset, and thus label-free.

E.4 VISUAL LOCALIZATION

Baselines: On Inloc, Table 5, we compare against the sparse matchers SP+SG (Sarlin et al., 2020), SP+LG (Lindenberger et al., 2023) and LoMA (Nordström et al., 2026), the default methods in hloc, as well as the semi-dense and dense baselines ASpanFormer (Chen et al., 2022), EDM (Li et al., 2025), MASt3R (Duisterhof et al., 2025), RoMA (Edstedt et al., 2023c) and RoMAv2 (Edstedt et al., 2026). We report the percentage of correctly localized queries at $(0.25\text{m}, 10°)$ / $(0.5\text{m}, 10°)$ / $(1.0\text{m}, 10°)$ and list the official numbers from the respective papers. We follow the hloc (Li et al., 2020) evaluation protocol. For each query, we retrieve the top-40 database images, establish 2D-2D correspondences per pair, lift them to 2D-3D using the database depth, and estimate the absolute pose with PnP.