

Scalable Feature Matching via State Space Modeling and Sparse Correlation

Choo Sin Wai¹ Bo Li^{1,2*}

¹School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China

²Shenzhen Research Institute, Northwestern Polytechnical University, Shenzhen, China

caosw25@mails.tsinghua.edu.cn, libo@nwpu.edu.cn

Abstract

*Efficient and robust feature matching is crucial for latency-sensitive and resource-constrained applications. However, many current semi-dense feature matching methods scale quadratically with spatial resolution, either due to transformer-based long-range context modeling or from redundant full correlation computations. To overcome these limitations, we present **SLiM** (Salient **L**ightweight **M**atching), a novel scalable feature matching method that delivers reliable matches with low memory footprint and latency, especially at high resolutions. Our approach introduces three key innovations: (1) a hybrid Conv-Mamba backbone for efficient cross-scale and cross-view feature extraction with linear complexity, (2) a training-free norm-based feature filtering mechanism, enabling sparse correlation that significantly reduces computation overhead during inference, and (3) a lightweight recurrent coordinate refinement that surpasses expectation-based regression in subpixel accuracy. Experimental results show that SLiM consistently achieves a strong accuracy-efficiency trade-off across indoor and outdoor benchmarks, with clear advantages in memory usage, inference speed, and high-resolution scalability. These results demonstrate the practical efficiency and strong scalability of SLiM. Project page: <https://github.com/Band-127/SLiM>*

1. Introduction

Image matching serves as a fundamental cornerstone in 3D computer vision, underpinning critical tasks including Structure-from-Motion (SfM) [1, 26, 32], Novel View Synthesis (NVS) [7, 21, 33, 34], Simultaneous Localization and Mapping (SLAM) [15, 35], and Visual Localization [49, 51].

Historically, image matching has evolved through two dominant paradigms. In the early stage, handcrafted feature-based approaches relied on descriptors like SIFT [29]

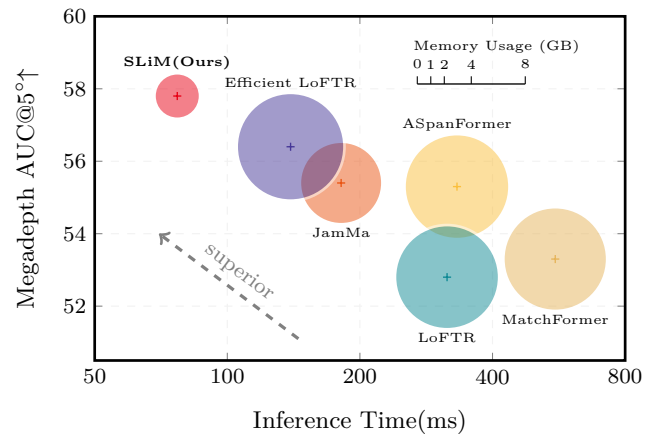


Figure 1. Accuracy-efficiency tradeoff comparison of semi-dense feature matchers on the MegaDepth dataset at 1200×1200 resolution. Inference time (ms) is shown on a logarithmic scale for clarity. Circle diameters represent peak memory consumption per method. Our approach achieves state-of-the-art performance across all metrics, demonstrating superior balance between matching accuracy and computational efficiency.

and ORB [40] for keypoint detection and description, yet struggled with challenging scenarios involving extreme viewpoint changes, textureless regions, and repetitive patterns. The deep learning revolution subsequently shifted the field towards data-driven learning methods for feature detection [4], description [11, 38, 56], and matching [27, 42]. Despite improved performance, decoupling of processing stages in these approaches results in increased model complexity and computational latency.

To overcome these limitations, detector-free methods emerged, bypassing explicit keypoint detection by direct modeling of dense feature correlations [24, 39]. However, their generalization remains constrained by the feature extraction backbone due to the limited cross-view feature interaction, which faces fundamental trade-offs between localization precision and feature discriminability. Recent transformer-based architectures [20] address this by modeling cross-view and cross-scale context interaction, achiev-

*Corresponding author.

ing SOTA accuracy. Subsequent optimizations have further refined these mechanisms across accuracy, efficiency, and robustness dimensions [6, 44, 50, 52].

Despite these advances, efficiency optimization receives limited attention compared to accuracy improvements. Reviewing contemporary frameworks, e.g. LoFTR [44], MatchFormer [50] and ASpanFormer [6], reveals that feature matching relies on full correlation computations, incurring memory consumption that scales quadratically with spatial resolution $\mathcal{O}(W^2H^2)$. Besides, the quadratic complexity of vanilla transformers is still a bottleneck for high-resolution matching, motivating efficient alternatives such as state-space models [19, 37], which achieve linear complexity while preserving global context. Nevertheless, empirical studies show that matching accuracy strongly depends on high input resolution, computational latency and memory demands exhibit prohibitively expensive growth patterns. These efficiency bottlenecks, particularly in feature extraction and global correlation, become critically limiting in resource-constrained applications.

We designed an efficient matching pipeline to improve computational efficiency while maintaining high matching accuracy. Our contributions can be summarized in the following three aspects:

- **Lightweight Network Design:** We develop a lightweight yet efficient feature matching network, integrating a three-layer Conv-Mamba backbone coupled with convolutional FPN for cross-view and cross-scale feature interaction with linear complexity, and a novel lightweight Recurrent Refinement Unit (RRU) with only 816K parameters to iteratively optimize spatial offsets.
- **Sparse Correlation via Norm-based Feature Filtering:** We observe that optimizing a log-likelihood loss tends to increase the L_2 norm of in-distribution features, and propose a training-free feature filtering strategy based on the L_2 norm as a discriminative saliency prior. This mechanism reduces computation by selectively identifying geometrically salient regions and significantly accelerates inference while preserving accuracy, outperforming the full-correlation strategy used in existing methods.
- **SOTA Results:** We achieve an AUC@5° of 57.9 on MegaDepth, which is 1.5 points higher than Efficient LoFTR, with only 5.9M parameters. In resolution scaling tests, our method uses 45% less memory and runs 2.4× faster than JamMa, and it also achieves 57% lower memory usage with 1.8× faster inference than Efficient LoFTR at high resolutions. These results consistently demonstrate a stronger accuracy-efficiency trade-off than existing semi-dense matchers, especially under high-resolution settings.

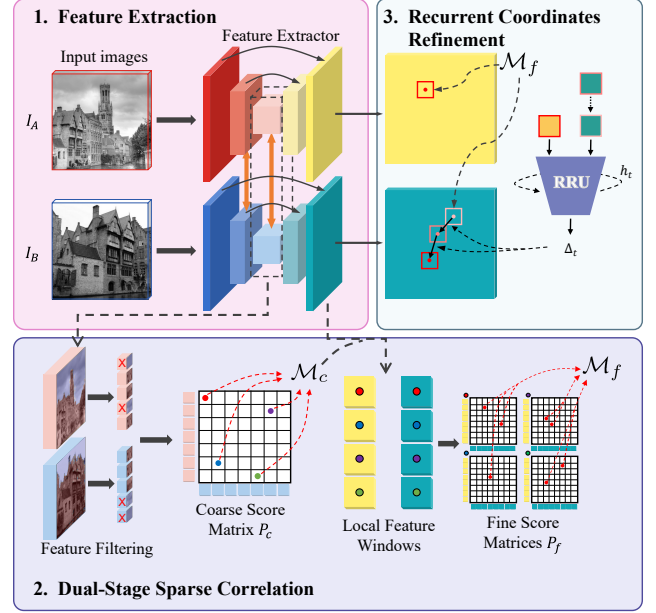


Figure 2. Our method consists of three main components: (1) Feature extraction employing a hybrid Conv-Mamba backbone with Feature Pyramid Network (FPN) for fusing cross-view and cross-scale context modeling; (2) Dual-stage coarse-to-fine sparse correlation establishing initial correspondences at 1/8 scale followed by high-resolution correlation at 1/2 scale; (3) Recurrent coordinate refinement operating on original-scale feature windows through a lightweight recurrent network that predicts subpixel offsets between reference and target feature windows.

2. Related Work

2.1. Feature Matching

Early visual-based matching methods relied on handcrafted descriptors like SIFT [29], SURF [5] and ORB [40], which detect keypoints and compute descriptors through predefined gradient or intensity patterns. These methods excel in efficiency and interpretability but struggle with textureless regions or extreme viewpoint changes due to their reliance on handcrafted heuristics.

The rise of deep learning has shifted the paradigm towards data-driven detector-based methods. Approaches like D2Net[11], R2D2[38] and SuperPoint[10] employ CNNs to jointly learn keypoint detection and descriptor extraction, while SuperGlue [42] introduces graph neural networks to model matching as a partial assignment problem. Despite improved robustness, these methods inherit a fragmented pipeline where detection, description, and matching are optimized separately. These fragmented stages lead to information loss and suboptimal performance in challenging scenarios, such as low-texture scenes or ambiguous matches.

To overcome these limitations, detector-free methods emerged, bypassing explicit keypoint detection by directly

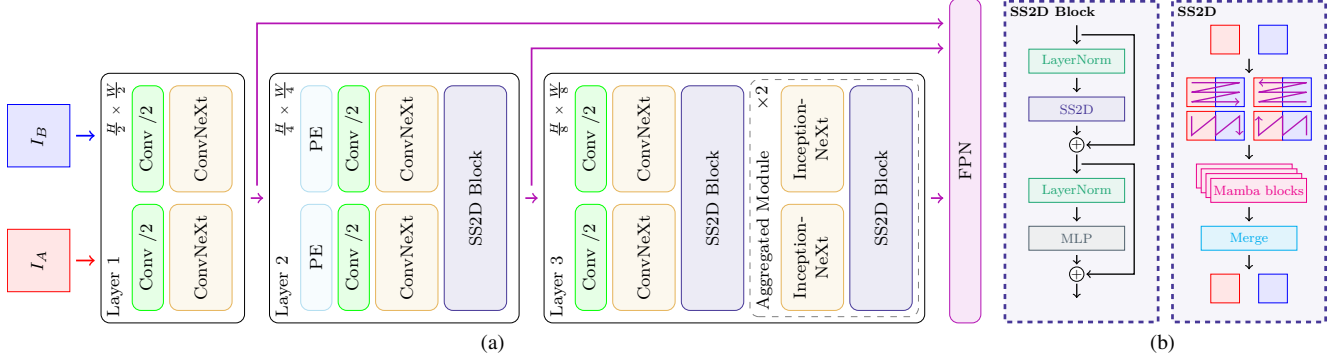


Figure 3. Backbone architecture overview: (a) Our feature extraction backbone consists of ConvNeXt, InceptionNeXt, and SS2D blocks in three hierarchical layers and a convolutional FPN at the end to produce multi-scale features; (b) shows the detailed architecture of SS2D Block and SS2D Module. In a single SS2D Module, features concatenate along width. The concatenated feature is then processed through four-directional scans and Mamba blocks, then splits and averages outputs to produce features with contextual information.

modeling dense feature correlations. NC-Net[39] pioneers this direction by constructing a 4D correlation tensor and applying 4D convolutions to enforce neighborhood consensus, while DRC-Net[24] further refines this through multi-resolution feature fusion. Recent works integrate transformers for long-context modeling: COTR[20] leverages cross-attention for cross-view correspondence search, while LoFTR[44] introduces a coarse-to-fine strategy with linear attention to balance efficiency and accuracy. Subsequent innovations, such as MatchFormer[50] and ASpanFormer[6], optimize attention mechanisms for cross-scale fusion, while Efficient LoFTR[52] reduces computational overhead through token pruning.

Recent efforts integrate state-space architectures into matching tasks. MambaGlue[41] replaces the transformer in SuperGlue with Mamba blocks to reduce computation. JamMa[30] introduces Joint Mamba with JEGO scanning to improve cross-view feature interaction at low cost. In contrast, our method adopts an interleaved Conv-Mamba backbone rather than simply replacing transformer modules with Mamba blocks.

Dense matching methods like DKM[12] and RoMa[13] aim to estimate pixel-wise correspondences by directly predicting a dense warp field between two images. While these dense matchers often outperform sparse and semi-dense pipelines in accuracy and robustness, they generally require substantially more parameters and longer runtimes, making them less suitable for latency-sensitive settings. In contrast, our method targets semi-dense matching with competitive performance and significantly better efficiency.

2.2. Modern State-space Models

Modern state-space models (SSMs) have emerged as efficient alternatives to transformers for sequence modeling, offering linear complexity with global receptive fields. Innovations include RWKV[37] with time-dependent recur-

rence and Mamba[19] featuring dynamic context-aware selection. Mamba2[9] further accelerates selective scans with parallelized state transitions for up to $8\times$ faster inference.

Vision adaptations exploit SSMs’ representational capacity: Vim[59] flattens images into bidirectional sequences, while VMamba[28] introduces structured spatial scanning (SS2D). Recent extensions demonstrate success in segmentation[31], detection[53], depth prediction[18], and video understanding[23], frequently outperforming transformers.

Hybrid Conv-Mamba architectures balance global context and local priors: U-Mamba[31] integrates SSMs into U-Net encoders, ReMamba[17] fuses CNN features for re-ID, and HC-Mamba[54] uses dilated convolutions before SSM layers. These approaches model global context while enabling high-resolution processing with computational efficiency.

3. Method

Fig. 2 illustrates the overall architecture of SLiM. Given a grayscale image pair $(I_A, I_B) \in \mathbb{R}^{H \times W \times 1}$, our method establishes semi-dense correspondences through three hierarchical stages. First, a Conv-Mamba hybrid encoder with a Feature Pyramid Network (FPN) extracts multi-scale feature pyramids $\{\mathbf{F}_l^A\}_{l=1}^L$ and $\{\mathbf{F}_l^B\}_{l=1}^L$ while fusing cross-scale and cross-view contextual information. Subsequently, we perform a dual-stage correlation process: coarse matching at $1/8$ resolution ($l = 3$) generates initial correspondences $\mathcal{M}_c$ through sparse correlation on filtered features, followed by fine matching at $1/2$ resolution ($l = 1$) that produces refined correspondences $\mathcal{M}_f$ via local window correlation. Finally, a lightweight Recurrent Refinement Unit (RRU) iteratively predicts precise subpixel offsets between feature windows.

3.1. Conv-Mamba Feature Extractor

As shown in Fig. 3a, our hybrid encoder employs a three-layer architecture for multi-scale feature extraction. The first layer employs a single ConvNeXt block with 7×7 kernels to process the input grayscale image $I \in \mathbb{R}^{H \times W \times 1}$, generating feature map $\mathbf{F}_1 \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 48}$ while preserving high-frequency details. The second layer stacks a ConvNeXt block with an SS2D block to produce $\mathbf{F}_2 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 96}$, where the SS2D block enables cross-view context fusion through four-directional scanning. The third layer combines a ConvNeXt block, an SS2D block, and two aggregated modules, each containing an InceptionNeXt block and an SS2D block, to generate high-level features $\mathbf{F}_3 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 192}$.

Cross-view fusion in SS2D blocks operates through a four-stage process: First, concatenate the paired features $\mathbf{F}^A$ and $\mathbf{F}^B$ along the width dimension to form $\mathbf{F}^{AB} \in \mathbb{R}^{H \times 2W \times C}$. Next, apply independent four-directional scans followed by four parallel Mamba blocks. Finally, split and average the processed features to obtain enhanced representations $\tilde{\mathbf{F}}^A$ and $\tilde{\mathbf{F}}^B$ with contextual information.

For the aggregated modules in the third layer, following the design philosophy of Efficient LoFTR, we downsample the deepest feature map by $4 \times$ via a stride-4 convolution to enhance the effective network depth without significantly increasing complexity. The downsampled features are then processed through an InceptionNeXt block with parallel branches of 3×3 , 1×7 , and 7×1 depthwise convolutions, followed by an SS2D block. The processed features are finally upsampled by a factor of 4 using bilinear interpolation. This design effectively balances receptive field expansion and computational efficiency.

Cross-scale fusion is achieved through a Feature Pyramid Network (FPN) that combines features from different levels using lateral connections. The fusion process at each level $l \in \{1, 2, 3\}$ can be formulated as:

$$\mathbf{F}_l^{\text{fused}} = \text{Conv}(\text{Conv}_{1 \times 1}(\mathbf{F}_l) + \text{Up}(\mathbf{F}_{l+1}^{\text{fused}})) \quad (1)$$

merging semantic information from deep layers with spatial details from shallow layers, enabling robust multi-scale representation learning.

3.2. Dual-stage Sparse Correlation

This work introduces a feature filtering strategy, using the L_2 norm of feature vectors as a saliency indicator. Our approach is inspired by out-of-distribution detection approaches [36, 57] which observe that in-distribution features progressively develop larger norms during training, creating a discernible separation from out-of-distribution features. This phenomenon occurs because classification outputs become more polarized and form harder probability distributions under softmax normalization, indirectly amplifying the L_2 norm of in-distribution features [55]. Some

detector-based matching approaches [2, 40, 47] correlate feature saliency with entropy and variance, demonstrating that these metrics can serve as keypoint saliency indicators.

Extending this mechanism to detector-free matching, we observe that salient and effective features naturally exhibit increased norms and variances during training. Empirical comparisons on these metrics (detailed in Sec. 4.4.2) demonstrate the superior discriminative power of L_2 norms for correspondence filtering. This also aligns with the practical need to reduce the coarse-stage global correlation bottleneck observed in recent efficient matchers such as Efficient LoFTR and JamMa.

Our method implements sparse correlation and dual softmax exclusively on high-norm features. Candidate positions $\mathcal{G}^A$ and $\mathcal{G}^B$ are determined by threshold η :

$$\begin{aligned} \mathcal{G}^A &= \{i \mid \|\mathbf{F}_{3,i}^A\|_2 > \eta\} \\ \mathcal{G}^B &= \{j \mid \|\mathbf{F}_{3,j}^B\|_2 > \eta\} \end{aligned} \quad (2)$$

In the coarse stage, local correlation C_{ij}^c is computed on $\frac{1}{8}$ scale features $\mathbf{F}_3^A$ and $\mathbf{F}_3^B$ within $\mathcal{G}^A$ and $\mathcal{G}^B$:

$$C_{ij}^c = \frac{\langle \mathbf{F}_{3,i}^A, \mathbf{F}_{3,j}^B \rangle}{\sqrt{d}}, \quad \forall i \in \mathcal{G}^A, j \in \mathcal{G}^B \quad (3)$$

with feature dimension $d = 192$. Dual softmax normalization yields the coarse score matrix P_{ij}^c :

$$P_{ij}^c = \text{softmax}_i(C_{ij}^c) \cdot \text{softmax}_j(C_{ij}^c) \quad (4)$$

Mutual nearest neighbors that satisfy $P_{ij}^c > \tau_c$ constitute initial coarse correspondences $\mathcal{M}_c$, with τ_c being the coarse confidence threshold.

In the fine stage, we perform local window correlation of $\frac{1}{2}$ scale features centered on $\mathcal{M}_c$ positions. For each coarse correspondence $(i_c, j_c) \in \mathcal{M}_c$, we extract $k \times k$ windows ($k=4$ in practice) from $\mathbf{F}_1^A$ and $\mathbf{F}_1^B$:

$$\mathbf{W}_{i_c}^A = \text{Crop}(\mathbf{F}_1^A, i_c), \quad \mathbf{W}_{j_c}^B = \text{Crop}(\mathbf{F}_1^B, j_c) \quad (5)$$

Fine correlation is computed after flattening each $k \times k$ window into k^2 feature tokens, and dual softmax then yields the correspondence score matrix P_{pq}^f :

$$C_{pq}^f = \frac{\langle \mathbf{W}_{i_c,p}^A, \mathbf{W}_{j_c,q}^B \rangle}{\sqrt{d}}, \quad p, q \in \{1, \dots, k^2\} \quad (6)$$

$$P_{pq}^f = \text{softmax}_p(C_{pq}^f) \cdot \text{softmax}_q(C_{pq}^f) \quad (7)$$

where feature dimension $d = 48$.

Final fine matches $\mathcal{M}_f$ are determined by:

$$\mathcal{M}_f = \left\{ (i_c + \Delta(p), j_c + \Delta(q)) \mid P_{pq}^f > \tau_f \right\} \quad (8)$$

where $\Delta(\cdot)$ maps a flattened index to its 2D offset in the $k \times k$ window, and τ_f is the fine confidence threshold.

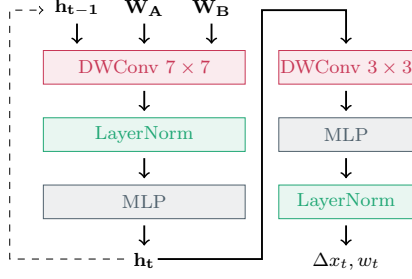


Figure 4. Architecture of the Recurrent Refinement Unit (RRU). The module iteratively adjusts coordinates using local feature-window context, with DWConv: Depthwise Convolution, LayerNorm: Layer Normalization.

3.3. Recurrent Coordinates Refinement

Existing feature matching frameworks [44, 50] typically employ expectation-based coordinate regression at the fine matching stage. Although computationally efficient, this approach suffers from two inherent limitations: (1) The expectation operation assumes a unimodal probability distribution within local windows, struggling with matching ambiguities caused by repetitive textures or noise; (2) Direct regression lacks explicit modeling of spatial consistency between adjacent pixels, potentially amplifying localization errors.

To address these limitations, drawing inspiration from recurrent architectures in optical flow estimation [43, 46], we reformulate subpixel refinement as an iterative coordinate optimization task. As shown in Fig. 4, this lightweight design (816K parameters) achieves high-accuracy localization in just 3–5 iterations while maintaining real-time performance. In practice, we set $T = 4$ for all evaluations and use a hidden state dimension of 32.

For each fine correspondence $(x^A, x_0^B) \in \mathcal{M}_f$ obtained from dual-stage matching, RRU begins by extracting $k \times k$ feature windows around current coordinates from original-scale features $\mathbf{W}^A, \mathbf{W}_t^B$. The core refinement process computes offset and context updates through efficient operations, starting with concatenated features passing through depthwise convolutions and MLPs to update hidden states. These states are then decoded into subpixel displacement map Δx_t and confidence weights w_t . Coordinates are adjusted via weighted averaging of local displacements: $x_{t+1}^B \leftarrow x_t^B + \sum_{i=1}^{k \times k} w_t^{(i)} \Delta x_t^{(i)}$, maintaining fixed x^A positions throughout iterations. After T iterations, final matches $\mathcal{M}_r$ achieve subpixel precision. This design provides adaptive accuracy-speed control by dynamically adjusting iteration counts during inference.

3.4. Supervision

We supervise our network on three loss functions:

Matching losses: Given the ground truth coarse corre-

spondences $\mathcal{M}_c^{gt}$ and fine correspondences $\mathcal{M}_f^{gt}$, we calculate the negative log-likelihood loss on both score matrices:

$$\begin{aligned} \mathcal{L}_c &= -\frac{1}{|\mathcal{M}_c^{gt}|} \sum_{i,j \in \mathcal{M}_c^{gt}} \log(P^c(i, j)) \\ \mathcal{L}_f &= -\frac{1}{|\mathcal{M}_f^{gt}|} \sum_{i,j \in \mathcal{M}_f^{gt}} \log(P^f(i, j)) \end{aligned} \quad (9)$$

Coordinate refinement loss: For recurrent coordinate refinement, we supervise all intermediate predictions using the ground-truth correspondence x_{gt}^B paired with the reference coordinate x^A , with an exponentially decayed L_2 loss:

$$\mathcal{L}_r = \sum_{t=1}^T \gamma^{T-t} \|x_{gt}^B - x_t^B\|_2 \quad (10)$$

where γ is the decay factor, empirically set to 0.8, and $\|\cdot\|_2$ computes the L_2 norm of a vector.

The final training objective combines these losses with balancing weights α, β , and λ :

$$\mathcal{L} = \alpha \mathcal{L}_c + \beta \mathcal{L}_f + \lambda \mathcal{L}_r \quad (11)$$

where the weights are determined through cross-validation to balance scale differences between matching probabilities and coordinate offsets.

4. Experiments

4.1. Implementation Details

All training uses the MegaDepth dataset [25], which comprises 130,000 internet photos from 196 scenes reconstructed using COLMAP. Following the SuperGlue experiment protocol [42], we have used 38,300 image pairs from 368 subscenes, using precomputed depth maps and camera intrinsics. All input images are resized to 1024×1024 resolution with edge padding while preserving the original aspect ratios.

Our training configuration uses AdamW with a weight decay of 0.1 and an initial learning rate of 1×10^{-3} . We set the loss weights to $\alpha = 0.25$, $\beta = 0.2$, and $\lambda = 1.0$ as defined in Eq. 11, and use a batch size of 6 on six RTX 4090 GPUs with 24GB VRAM each. The model converges within 21 hours over 19 epochs. We evaluate the trained model on other datasets without fine-tuning, following the Efficient LoFTR protocol [52] to demonstrate generalization capability.

4.2. Relative Pose Estimation

Datasets. We evaluate indoor relative pose estimation on ScanNet[8] using 1,500 challenging monocular image pairs. All images are resized to 640×480 while preserving aspect ratios to maintain indoor scene geometry. For outdoor evaluation, we use 1,500 MegaDepth pairs sampled from the

Table 1. Relative pose estimation results on MegaDepth (1200×1200) and ScanNet (640×480). AUC scores at 5°/10°/20° thresholds with matching time in ms. † denotes the variant using 1 iteration for refinement while the main variant uses 4 iterations. ‡ denotes MegaDepth results obtained with tuned RANSAC inlier threshold, following LightGlue. Parameter counts indicate trainable weights only. Feature filtering is used in all variants of our method.

Method	Params (M)	MegaDepth AUC↑			ScanNet AUC↑			Inference Time (ms)↓	
		5°	10°	20°	5°	10°	20°	MegaDepth	ScanNet
SuperPoint[10]+SuperGlue[42]	13.3	49.7	67.1	80.6	16.2	32.8	49.7	114.1	48.3
SuperPoint[10]+LightGlue[27]‡	13.2	49.9	67.0	80.1	14.8	30.8	47.5	44.2	30.7
DRC-Net[24]	10.2	27.0	42.9	58.3	7.7	17.9	30.5	1479.0	328.0
LoFTR[44]	11.6	52.8	69.2	81.2	16.9	33.6	50.6	315.4	66.2
QuadTree[45]	12.7	54.6	70.5	82.2	19.0	<u>37.3</u>	53.5	391.9	100.7
MatchFormer[50]	20.3	53.3	69.7	81.8	15.8	32.0	48.0	555.2	128.9
ASpanFormer[6]	15.8	55.3	71.5	83.1	19.6	37.7	54.4	332.2	81.6
Efficient LoFTR[52]	16.0	56.4	72.2	83.5	<u>19.2</u>	37.0	<u>53.6</u>	139.2	34.4
MambaGlue[41]	17.8	50.1	67.5	80.3	/	/	/	46.3	40.9
JamMa[30]	5.7	55.4	70.8	82.1	14.5	29.8	46.2	181.3	<u>26.6</u>
SLiM	5.9	<u>57.9</u>	<u>72.8</u>	83.5	18.0	34.7	50.4	77.0	32.9
SLiM †	5.9	<u>57.6</u>	<u>72.8</u>	83.3	17.7	33.8	49.3	65.3	23.9
SLiM ‡	5.9	60.1	73.8	<u>83.4</u>	18.2	34.7	51.3	77.0	32.9

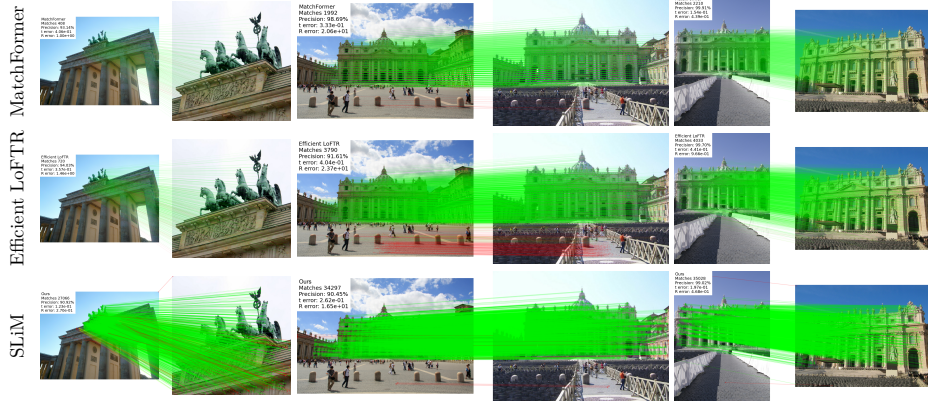


Figure 5. Qualitative comparison on MegaDepth. Our method produces denser and more accurate correspondences compared to Efficient LoFTR and MatchFormer. The improved matching quality directly contributes to better pose estimation accuracy as shown in Table 1.

Table 2. Comparison with dense matchers on MegaDepth. AUC scores are taken from RoMa v2. Runtime is measured on a single RTX 4090 GPU under AMP. “Res.” indicates the input resolution used for runtime measurement; “/” denotes that runtime is not reported/measured.

Method	MegaDepth AUC↑			Res.	Time (ms)↓
	5°	10°	20°		
DKM[12]	60.4	74.9	85.1	1200	799.6
RoMa[13]	62.6	76.7	86.3	1200	1176.8
RoMa v2[14]	62.8	77.0	86.6	640	202.1
MASt3R[22]	42.4	61.5	76.9	/	/
UFM[58]	41.5	57.9	72.4	/	/
SLiM	57.9	72.8	83.5	1200	77.0

“Sacre Coeur” and “St. Peter’s Square” sub-scenes, resizing images to 1200×1200 with edge padding while preserving the original aspect ratios.

Evaluation Protocols. Pose accuracy is measured by the Area Under the Cumulative Curve (AUC) of the maximum angular pose error in rotation and translation at 5°/10°/20°

thresholds. Essential matrices are estimated via vanilla RANSAC from predicted correspondences. Inference timings are reported on a single RTX 4090 GPU under PyTorch automatic mixed precision (AMP). For SuperGlue, LightGlue, and MambaGlue, we follow their official evaluation protocols where at most 2048 SuperPoint keypoints are extracted. For fairness with LightGlue, we additionally report a tuned RANSAC inlier threshold setting (‡) on MegaDepth.

Comparative Methods. We compare against three categories of approaches: (1) Sparse methods built on SuperPoint features, including SuperPoint+SuperGlue, SuperPoint+LightGlue, and SuperPoint+MambaGlue; (2) Semi-dense methods, including DRC-Net, LoFTR, QuadTree, ASpanFormer, MatchFormer, Efficient LoFTR, and JamMa; and (3) Dense matchers, including DKM, RoMa, RoMa v2, MASt3R, and UFM.

Results. As shown in Table 1, our method achieves state-of-the-art performance among sparse and semi-dense matchers

on MegaDepth. SLiM reaches an AUC@5° of 57.9, which is 1.5 points higher than Efficient LoFTR, while using only 5.9M parameters, which is 50.9% of LoFTR’s model size.

Under a tuned RANSAC inlier threshold on MegaDepth, SLiM further reaches 60.1/73.8/83.4 AUC at 5°/10°/20°. In terms of efficiency, the single-iteration variant marked by † processes ScanNet pairs in 23.9ms under AMP and remains faster than JamMa at 26.6ms while delivering better accuracy. Although our method has higher latency than sparse methods such as LightGlue and MambaGlue at high resolutions, it provides substantially stronger accuracy across diverse scenarios.

Compared with dense matchers in Table 2, our method remains substantially more efficient while preserving competitive accuracy, aligning with our primary goal. Methods such as DKM, RoMa, and MAST3R are generally stronger in absolute accuracy partly because they are trained with larger and more diverse multi-dataset supervision rather than MegaDepth alone, which gives them a clear training-data advantage.

4.3. Homography Estimation

Datasets. We evaluate on HPatches[3], which contains 108 image sequences, including 56 viewpoint-varying sequences and 52 illumination-varying sequences. Each sequence contains one reference image and five query images. Following standard protocols, we generate 540 test pairs and resize images to 480 pixels along the shorter edge while preserving aspect ratios.

Evaluation Protocol. Homography accuracy is measured by the AUC of mean corner reprojection errors at 3, 5, and 10 pixel thresholds. Following [44], we estimate homographies via RANSAC [16] using at most 1000 correspondences to ensure fair comparison.

Comparative Methods. We compare against two categories of approaches: (1) Sparse methods, including D2Net+NN, R2D2+NN, DISK+NN, and SuperPoint+SuperGlue; and (2) Semi-dense methods, including NC-Net, DRC-Net, LoFTR, Efficient LoFTR, and JamMa.

Results. As shown in Table 3, our method achieves state-of-the-art 69.4/78.2/86.3 AUC at 3/5/10 pixel thresholds. It outperforms JamMa by 1.3/1.2/0.9 points at the corresponding thresholds. We attribute these gains to the proposed recurrent refinement mechanism, which provides stronger multi-peak hypothesis aggregation than expectation-based coordinate regression and improves geometric consistency.

4.4. Ablation study

4.4.1. Resolution Scaling Evaluation

To assess our method’s scalability across resolutions, we conduct some resolution scaling tests on the MegaDepth dataset. As shown in Fig. 6, all evaluations operate under mixed precision with disabled gradient computation.

Table 3. Homography estimation results on HPatches. AUC scores of mean corner reprojection errors at 3/5/10px thresholds demonstrate our method’s superior matching precision.

Method	HPatches AUC↑		
	3px	5px	10px
D2Net+NN[11]	23.2	35.9	53.6
R2D2+NN[38]	50.6	63.9	76.8
DISK+NN[48]	52.3	64.9	78.9
SP+SG[42]	53.9	68.3	81.7
NC-Net[39]	48.9	54.2	67.1
DRC-Net[24]	50.6	56.2	68.2
LoFTR[44]	65.9	75.6	84.6
EfficientLoFTR[52]	66.5	76.4	85.5
JamMa[30]	68.1	77.0	85.4
SLiM	69.4	78.2	86.3

The horizontal axis represents input resolution, with vertical axes measuring: (a) peak GPU memory consumption, (b) per-pairs inference latency, and (c) matching accuracy (AUC@5°).

Our approach achieves 45% memory reduction compared to JamMa at 1200×1200 resolution while maintaining near-linear scaling characteristics (Fig. 6a). Inference acceleration reaches 1.8× compared to Efficient LoFTR (Fig. 6b), with consistent accuracy outperformance on high resolutions (Fig. 6c).

4.4.2. Filtering Metrics Comparison

We have conducted qualitative analysis comparing mean, variance, and L_2 norm as correspondence filtering metrics. Fig. 7 reveals that at 1/8 resolution scale, variance and L_2 norm distribution (b)(c) precisely identify salient regions like building structures. After threshold filtering (normalized value $> \eta$, where η empirically set to 0.65, shown in red), the highlighted regions correspond well to salient structures (e)(f). Quantitative analysis in Table 4 and 5 demonstrate that L_2 norm filtering significantly improves efficiency and accuracy on both MegaDepth and ScanNet datasets, outperforming variance-based and mean-based filtering alternatives.

Table 4. Accuracy and efficiency comparison of feature filtering metrics on MegaDepth.

Feature Metric	MegaDepth AUC↑			Inference Time (ms)
	5°	10°	20°	
No Filtering	57.9	72.6	83.2	128.7
Mean	55.1	69.8	80.3	65.0
Variance	57.2	72.3	83.0	78.4
L_2 Norm	57.6	72.8	83.3	77.0

4.4.3. Plug-in Generalization on External Methods

We evaluate the generality of our training-free filter strategy by inserting it into off-the-shelf matchers at the coarse correlation stage without retraining. All methods are evaluated under the same resolution and AMP setup as in Sec. 4.2.

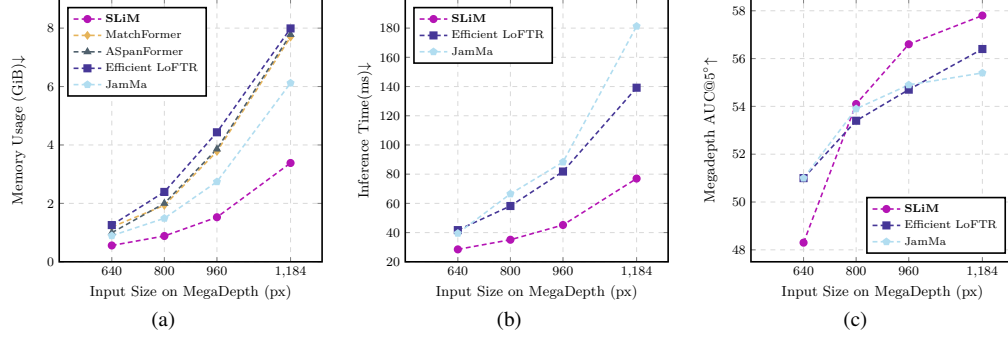


Figure 6. Resolution Scaling Evaluation: (a) Comparison of peak GPU memory consumption across input resolutions; (b) Comparison of inference time measurements across input resolutions; (c) Comparison of matching accuracy (AUC@5° on MegaDepth dataset) across input resolutions. All measurements performed under mixed-precision mode with gradient computation disabled.

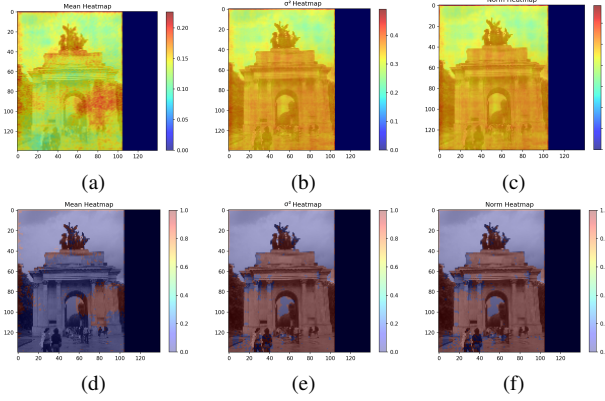


Figure 7. Feature filtering metric analysis at 1/8 resolution: (a-c) Raw distributions of feature (a) means, (b) variances, (c) L_2 norms; (d-f) Thresholded versions showing regions of feature (d) mean, (e) variances, and (f) L_2 norms exceeding the 0.65 normalized threshold (red regions). The L_2 norm demonstrates superior discriminative power for salient features while suppressing background responses.

Table 5. Accuracy and efficiency comparison of feature filtering metrics on ScanNet.

Feature Metric	ScanNet AUC $\uparrow$			Inference Time (ms)
	5°	10°	20°	
No Filtering	17.4	33.6	49.2	41.1
Mean	0.6	1.8	4.4	31.7
Variance	15.1	30.5	45.5	35.0
L_2 Norm	17.7	33.8	49.3	23.9

Protocol. We threshold at $\eta=0.65$ on the 1/8 scale for both views, keeping the rest of each method unchanged. We report AUC at 5°/10°/20° and inference time on a single RTX 4090 under AMP with batch size 1.

Results. On LoFTR and MatchFormer, the plug-in variant even slightly improves accuracy while reducing latency, e.g., LoFTR gains 0.4 at AUC@5° while latency drops from

315.4ms to 268.9ms, and MatchFormer gains 0.1/0.2/0.3 at AUC@5°/10°/20° while latency drops from 560.6ms to 487.2ms. For Efficient LoFTR, accuracy remains essentially unchanged but latency is nearly halved from 139.2ms to 76.0ms. These observations indicate that the training-free filter serves as a general accelerator and can yield small but consistent gains by suppressing low-saliency responses without retraining. We hypothesize that these heterogeneous improvements arise because the optimal threshold η is method-dependent, leading to different accuracy-efficiency trade-offs across architectures. The threshold η should be tuned per method when adopting this plug-in.

Table 6. Plug-in generalization: applying our L_2 -norm filtering to existing matchers without retraining. $\star$ denotes the plugin variant.

Method	MegaDepth AUC $\uparrow$			Time $\downarrow$ (ms)
	5°	10°	20°	
LoFTR	52.8	69.2	81.2	315.4
LoFTR $\star$	53.2	69.2	81.2	268.9
MatchFormer	53.3	69.7	81.8	555.2
MatchFormer $\star$	53.4	69.9	82.1	487.2
Efficient LoFTR	56.4	72.2	83.5	139.2
Efficient LoFTR $\star$	56.1	71.9	83.3	76.0

5. Conclusion

We present an efficient and scalable feature matching method by combining a Conv-Mamba hybrid backbone with L_2 norm guided sparse correlation, eliminating redundant computation while preserving salient features. The lightweight recurrent refinement module (RRU) further enhances accuracy without compromising throughput. Moreover, our proposed L_2 -norm filtering strategy is training-free and plug-and-play, and can be integrated into existing semi-dense matchers at inference without retraining to reduce correlation cost while largely preserving accuracy.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China under Grant 62001394, and the Science, Technology and Innovation Commission of Shenzhen Municipality under Grant ZD-CYKCX20250901094205007.

References

- [1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, pages 72–79, 2009. 1
- [2] Vassileios Balntas, Lilian Tang, and Krystian Mikolajczyk. Bold - binary online learned descriptor for efficient image matching. In *CVPR*, pages 2367–2375, 2015. 4
- [3] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 7
- [4] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. KeyNet: Keypoint Detection by Handcrafted and Learned CNN Filters. In *ICCV*, 2019. 1
- [5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006. 2
- [6] 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. 2, 3, 6
- [7] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *CVPR*, 2023. 1
- [8] 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. 5
- [9] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *ICML*, 2024. 3
- [10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, pages 337–33712, 2018. 2, 6
- [11] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR*, 2019. 1, 2, 7
- [12] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *CVPR*, 2023. 3, 6
- [13] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. In *CVPR*, 2024. 3, 6
- [14] 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. *arXiv preprint arXiv:2511.15706*, 2025. 6
- [15] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, pages 834–849, 2014. 1
- [16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 7
- [17] Haokun Geng, Jiaren Peng, Wenzhong Yang, Danny Chen, Hongzhen Lv, Guanghan Li, and Yi Shao. Remamba: a hybrid cnn-mamba aggregation network for visible-infrared person re-identification. *Scientific Reports*, 14(1):29362, 2024. 3
- [18] Ionuț Grigore and Călin-Adrian Popa. Mambadepth: Enhancing long-range dependency for self-supervised fine-structured monocular depth estimation. *arXiv preprint arXiv:2406.04532*, 2024. 3
- [19] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024. 2, 3
- [20] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. COTR: Correspondence Transformer for Matching Across Images. In *ICCV*, 2021. 1, 3
- [21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1
- [22] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024. 6
- [23] Kunchang Li, Xinhao Li, Yi Wang, Yanan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *ECCV*, 2024. 3
- [24] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *NeurIPS*, 2020. 1, 3, 6, 7
- [25] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 5
- [26] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, 2021. 1
- [27] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 1, 6
- [28] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024. 3
- [29] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 1, 2
- [30] Xiaoyong Lu and Songlin Du. Jamma: Ultra-lightweight local feature matching with joint mamba. In *CVPR*, 2025. 3, 6, 7
- [31] Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024. 3
- [32] Wei-Chiu Ma, Anqi Joyce Yang, Shenlong Wang, Raquel Urtasun, and Antonio Torralba. Virtual correspondence: Hu-

- mans as a cue for extreme-view geometry. In *CVPR*, 2022. 1
- [33] 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*, pages 405–421, 2020. 1
- [34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 1
- [35] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1
- [36] Jaewoo Park, Jacky Chen Long Chai, Jaeho Yoon, and Andrew Beng Jin Teoh. Understanding the feature norm for out-of-distribution detection. In *ICCV*, 2023. 4
- [37] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In *EMNLP*, pages 14048–14077, 2023. 2, 3
- [38] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 1, 2, 7
- [39] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiro Torii, Tomás Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2018. 1, 3, 7
- [40] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571, 2011. 1, 2, 4
- [41] Kihwan Ryoo, Hyungtae Lim, and Hyun Myung. MambaGlue: Fast and Robust Local Feature Matching With Mamba. *arXiv preprint arXiv:2502.00462*, 2025. 3, 6
- [42] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4937–4946, 2020. 1, 2, 5, 6, 7
- [43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 5
- [44] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2, 3, 5, 6, 7
- [45] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *ICLR*, 2022. 6
- [46] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 5
- [47] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2d: Keypoint extraction with describe to detect approach. In *ACCV*, pages 223–240, 2021. 4
- [48] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. In *NeurIPS*, 2020. 7
- [49] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *ICCV*, pages 627–637, 2016. 1
- [50] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, 2022. 2, 3, 5, 6
- [51] Sijie Wang, Qiyu Kang, Rui She, Wee Peng Tay, Andreas Hartmannsgruber, and Diego Navarro Navarro. Robustloc: robust camera pose regression in challenging driving environments. In *AAAI*, 2023. 1
- [52] 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. 2, 3, 5, 6, 7
- [53] Zeyu Wang, Chen Li, Huiying Xu, and Xinzhong Zhu. Mamba yolo: Ssms-based yolo for object detection. *arXiv preprint arXiv:2406.05835*, 2024. 3
- [54] Jiahu Xu. Hc-mamba: Vision mamba with hybrid convolutional techniques for medical image segmentation. *arXiv preprint arXiv:2405.05007*, 2024. 3
- [55] Kunran Xu, Lai Rui, Yishi Li, and Lin Gu. Feature normalized knowledge distillation for image classification. In *ECCV*, pages 664–680, 2020. 4
- [56] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal V. Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 1
- [57] Yeonguk Yu, Sungho Shin, Seongju Lee, Changhyun Jun, and Kyoobin Lee. Block selection method for using feature norm in out-of-distribution detection. In *CVPR*, pages 15701–15711, 2023. 4
- [58] Yuchen Zhang, Nikhil Varma 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. 6
- [59] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *ICML*, 2024. 3