– Supplementary Material –
Matching 2D Images in 3D: Metric Relative Pose from Metric Correspondences

1. Architecture Details

Complementary to the details and definitions of MicKey from the main paper, we include the details of its implementation in Tables 1 and 2. As mentioned, from a single input image I, MicKey computes the 2D keypoint offset (U), confidence (C), depth (Z), and descriptor (D) maps. The network is split into two main blocks, the feature encoder and the keypoint heads. We use as our feature encoder DINOv2, without further training or fine-tuning. We refer to its original paper [19] for additional details on DINOv2. In Tables 1 and 2, we detail the layers within the different keypoint heads. Each keypoint head is composed of ResNet blocks [1], a small self-attention layer, and specific activation functions. As explained in the tables, a ResNet block is composed of 3×3 convolutions, batch normalization layers, ReLU activations, and a residual connection. The residual connection is done between the input of the block and its output. Given that DINOv2 is not trained, in every head, we add a small self-attention layer to allow trainable message-passing systems within the network. We use linear attention to reduce the computational complexity [13]. The transformer has three attention layers, and each layer has eight attention heads. Note that we do not apply the transformer right after DINOv2 encoder, but instead, we use it after processing the feature maps to a smaller descriptor dimension to reduce the overall complexity and memory of our network. Finally, at the end of the heads, we use different activation functions depending on the keypoint head. For instance, we apply a Sigmoid activation to the keypoint offsets to map them to the range $[0, 1]$, *i.e.*, the offset is allowed to move within its corresponding grid.

Offset Head (U)

Layer	Description	Output Shape
	Feature map F	[b, 1024, w, h]
1	ResNet block 1	[b, 512, w, h]
2	ResNet block 2	[b, 256, w, h]
3	ResNet block 3	[b, 128, w, h]
4	Self-Attention	[b, 128, w, h]
5	ResNet block 4	[b, 64, w, h]
6	Conv. - Sigmoid	[b, 2, w, h]

Depth Head (Z)

Layer	Description	Output Shape
	Feature map F	[b, 1024, w, h]
1	ResNet block 1	[b, 512, w, h]
2	ResNet block 2	[b, 256, w, h]
3	ResNet block 3	[b, 128, w, h]
4	Self-Attention	[b, 128, w, h]
5	ResNet block 4	[b, 64, w, h]
6	3×3 Convolution	[b, 1, w, h]

Table 1. **3D Keypoint Coordinate Heads.** The feature extractor computes features F from an input image I. The dimension of the feature map is $(1024, w, h)$, where $w = W/14$ and $h = H/14$, and W and H refer to the width and height of I. The feature map F is then processed by different keypoint heads in parallel. A ResNet block is composed of 3×3 convolutions, batch normalization layers [12], ReLU activations [1], and a residual connection. The Self-Attention layer refers to a self-attention transformer with linear attention [13].

2. Training Details

Training parameters. We train MicKey with a batch size of 48 image pairs. At the start of the training, we use only the 30% of pairs to optimize the network ($b_{\text{min}} = 14$), and linearly increase the number of used pairs by 10% every 4k training iterations. We stop increasing the number of considered pairs when we reach the 80% of the batch ($b_{\text{max}} = 38$). The warm-up period finishes after 20k iterations. For the *null hypothesis*, we define $\text{VCRE}^{\text{max}} = 120$ pixels and s^0 is defined as the 30% of correspondences being inliers, *i.e.*, if having a correspondence set of size 100, $s^0 = 30$. During correspondence selection, we define the temperature of the descriptor Softmax (Equation 3 in the main paper) as $\theta_m = 0.1$ and initialize the learnable dustbin parameter to 1. In Equation 5 of the main paper, β controls the smoothness on the soft-inlier counting, and it is defined in dependence of the inlier threshold, τ. We define the threshold as $\tau = 0.15$m.

Optimization. MicKey is trained in an end-to-end manner with randomly initialized weights and ADAM optimizer [15] with a learning rate of 10^{-4}. We train on four V100 GPUs and the network converges after seven days. To pick the best checkpoint, we evaluate the performance in a subset of the validation dataset in terms of the Area Under the

Confidence Head (C)		
Layer	Description	Output Shape
	Feature map F	[b, 1024, w, h]
1	ResNet block 1	[b, 512, w, h]
2	ResNet block 2	[b, 256, w, h]
3	ResNet block 3	[b, 128, w, h]
4	Self-Attention	[b, 128, w, h]
5	ResNet block 4	[b, 64, w, h]
6	Conv. - Spatial Softmax	[b, 1, w, h]

Descriptor Head (D)		
Layer	Description	Output Shape
	Feature map F	[b, 1024, w, h]
1	ResNet block 1	[b, 512, w, h]
2	ResNet block 2	[b, 256, w, h]
3	ResNet block 3	[b, 128, w, h]
4	Self-Attention	[b, 128, w, h]
5	ResNet block 4	[b, 128, w, h]
6	L2 Normalization	[b, 128, w, h]

Table 2. **Confidence and Descriptor Heads.** Similar to the 3D coordinate regressors, MicKey also computes the descriptors and confidence scores of each keypoint. Each head has a different activation function, *e.g.*, in the descriptor head, we L2 normalize the descriptors and map them to a sphere of radius 1.

Curve (AUC) of the VCRE metric. We check validation results twice at every epoch, which corresponds to $\sim$1k training iterations.

Virtual Correspondences. Equation 7 from the main paper uses virtual correspondences to compute the Virtual Correspondence Reprojection Error (VCRE). We define such virtual correspondences as in Map-free benchmark [2]. The virtual correspondences represent a uniform grid of 3D points that will be projected into the 2D image plane to compute the VCRE. We define a total of 196 virtual correspondences ($|\mathcal{V}| = 196$). They correspond to a cube of $2.1 \times 1.2 \times 2.1$ meters in XYZ coordinates, where the minimum separation between 3D points is 0.3m. Note that this formulation already embeds the quality of an estimated pose in a single value, the VCRE. Hence, contrary to the standard pose loss formulation, using VCRE as a loss function does not require a parameter that balances the translational and rotational components of the pose error loss [3, 21].

3. Additional Experiments

In addition to the experiments and visualization reported in the main paper, we provide more insights, visualizations, and experiments in this section.

3.1. ScanNet

Depth ablation. Similar to the ablation study on depth estimation methods done in the main paper, we also report the results of sparse and dense matchers combined with different depth estimators in Table 3. Specifically, we use

ScanNet Dataset			
	VCRE		Median Errors
	AUC	Prec. (%)	Trans (m) / Rot (°)
Depth Estimation			
SuperGlue [6, 22]			
DPT [20]	0.98	90.0	0.17 / **2.06**
PlaneRCNN [17]	0.98	90.6	0.15 / **2.06**
Our Depth	**0.99**	**91.7**	**0.11 / 2.06**
GT Depth	0.99	92.9	0.07 / 2.06
LoFTR [23]			
DPT [20]	**0.99**	89.4	0.16 / **1.81**
PlaneRCNN [17]	**0.99**	**91.3**	0.13 / **1.81**
Our Depth	**0.99**	90.3	**0.10 / 1.81**
GT Depth	0.99	91.3	0.07 / 1.81

Table 3. **Relative pose evaluation in ScanNet for different depth estimators**. We show here the results of different matching algorithms paired with different depth estimators and MicKey's depths. SuperGlue, a sparse feature matcher, obtains top results when combined with our depth estimations. This is in line with our system, since we optimize our depth head to work with sparse rather than dense features.

ScanNet	δ_1 / δ_2 / δ_3 ↑	REL ↓	RMSE ↓	$\log_{10}$ ↓
DPT [20]	0.72 / 0.91 / 0.97	0.21	0.38	0.08
PlaneRCNN [17]	0.75 / 0.93 / 0.98	0.18	0.37	0.07
ZoeDepth [4]	0.79 / 0.94 / 0.98	0.17	**0.33**	0.07
MicKey	0.79 / **0.95** / 0.98	0.16	0.37	**0.06**
MicKey-Sc	**0.80 / 0.95 / 0.99**	**0.15**	0.35	**0.06**

Table 4. **Indoor monocular depth evaluation.** Even though MicKey depth maps are optimized to produce precise relative poses, and hence, might not be accurate beyond keypoint positions, we see that MicKey still produces accurate depth maps that are on par with or surpass current state-of-the-art methods.

DPT [20] and PlaneRCNN [17], where the latest used the ground truth depth maps provided in ScanNet for its training. As a reference, we provide the results when combining the matchers with the ground truth depths. Even though MicKey did not use any ground truth depth data during training, both matchers, SuperGlue [6, 22] and LoFTR [23], benefit from using our depth maps. SuperGlue, a sparse feature method like MicKey, is the one that yields better results with our depths. MicKey's depth maps were trained specifically for sparse matching, and hence, a sparse matching method could benefit more from them.

Monocular depth evaluation. Table 4 presents monocular depth evaluation metrics for the ScanNet test set, where we have access to GT depths. For completeness, besides DPT [20] and PlaneRCNN [17], we also evaluate the depth estimations of recent ZoeDepth [4]. MicKey-Sc corresponds to evaluating the depth estimates only on the positions where MicKey is most confident. Specifically, we take the depth estimations that correspond to the top 50% scoring positions. And therefore, in this setup, we focus on the posi-

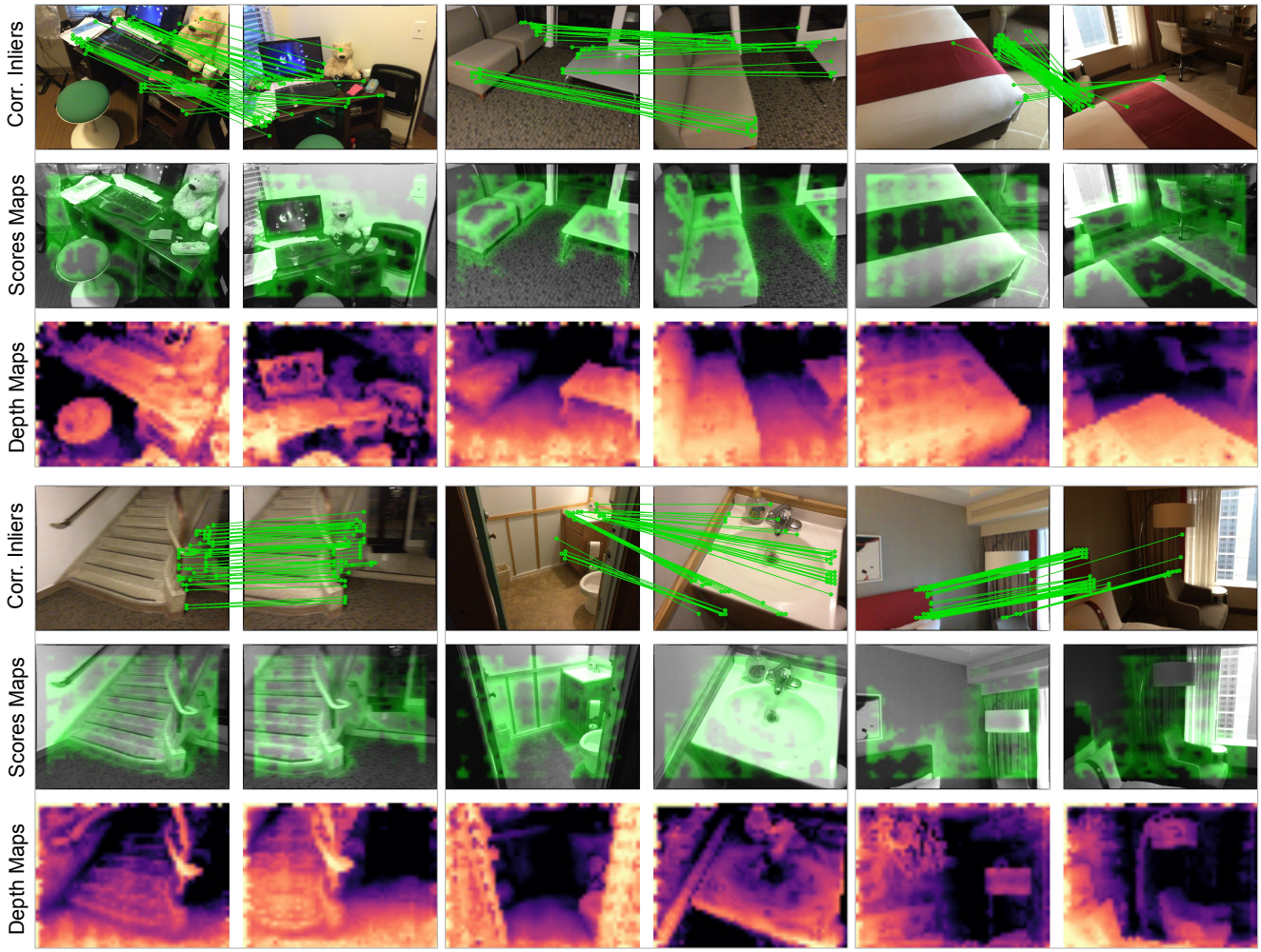

Figure 1. **Visual examples on ScanNet dataset.** We visualize the inlier correspondences, score, and depth maps predicted by MicKey in ScanNet images. We see that although the images show repetitive patterns and flat areas, MicKey is able to find correct correspondences across views. Besides making our correspondences metric, our depth estimation head provides additional 3D information and geometric constraints to our probabilistic pose solver, making the final decision robust against the mentioned indoor challenges.

tions that will be used to compute the metric relative pose between images. We see that our relative pose supervision, even though not using any depth maps during training, allows MicKey to compute accurate depth maps.

Visual examples. We also show some visual examples of the depth maps estimated by MicKey in Figure 1. Moreover, in the figure, we display the inlier correspondences and the score maps that MicKey computes for each input image. We see that MicKey finds high scoring keypoints in structures beyond corners or blobs, establishing correct matches in images where there are few discriminative structures.

3.2. Map-free

Method Confidence tells when we can or not trust a pose estimate. Map-free benchmark evaluates the confidence of the methods via the area under the curve (AUC) metric. The benchmark ranks the poses by confidence, and hence, the AUC is only maximized when the most accurate poses are ranked first. In Figure 2, we visualize the precision versus the ratio of estimates in the validation set scenes, where ground truth poses are available. All matching methods use their inlier counting as their confidence value, meanwhile, MicKey relies on the soft-inlier counting. MicKey's confidence, hence, is entangled within its training pipeline and directly optimized to be correlated with the quality of its pose predictions. From Figure 2, we see that MicKey obtains the highest number of correctly ranked images before assigning a high score to an invalid pose, where an invalid pose refers to a relative pose with a VCRE higher than 90 pixels. Contrary, as seen in the plot, the second best method,

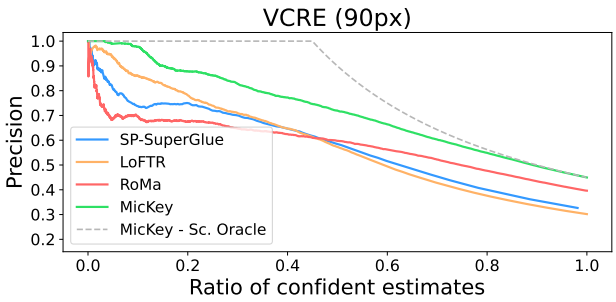

Figure 2. **Precision vs ratio of estimates.** In this plot, the pose estimates are ranked by the confidence values of the methods, *e.g.*, inlier counting. The Map-free benchmark computes the AUC of the curves to also evaluate the ability of the methods to decide whether their poses should be trusted or not. We plot MicKey-Score (Sc.) Oracle, which corresponds to the curve that MicKey would obtain if its pose confidences were perfect.

Map-free Dataset	VCRE (90px)		Pose (25cm, 5°)	
	AUC	Prec. (%)	AUC	Prec. (%)
D+O+P Signal				
SuperGlue [6, 22]	0.60	36.1	0.35	16.8
LightGlue [16, 24]	0.53	33.2	0.31	15.8
DeDoDe [8]	0.53	31.2	0.26	12.5
LoFTR [23]	0.61	34.7	0.35	15.4
ASpanFormer [5]	0.64	36.9	0.36	16.3
RoMa [7]	0.67	45.6	**0.41**	**22.8**
O+P Signal				
RPR [R(6D) + t] [2]	0.40	40.2	0.06	6.0
MicKey-O (ours)	**0.75**	**49.2**	0.33	13.3
Pose Signal				
RPR [R(6D) + t]	0.18	18.1	0.01	0.6
MicKey (ours)	0.74	**49.2**	0.28	12.0

Table 5. **Additional metrics on Map-free dataset.** Besides the VCRE results, we also show the AUC and precision values for a very fine threshold (pose errors at 25cm and 5°).

RoMa [7], struggles to rank its pose estimates. *I.e.*, RoMa accepts incorrect poses (VCRE $>$ 90px) as its most confidence estimates. This indicates that RoMa's poses, although very accurate, do not have a valid mechanism to decide whether they should be trusted or rejected, making them unreliable in an AR application [2].

Pose Metrics. Besides the experiments from the main paper in Map-free, we provide additional metrics in Table 5. Map-free benchmark, although it focuses on the VCRE metric for evaluating algorithms for an AR experience, it also computes the AUC and precision error pose under a very fine threshold (pose error $<$ 25cm, 5°). Under such conditions, all methods have a small AUC and precision value, and hence, applications built on that restrictive threshold would need to discard most of the relative pose estimates. Even though, it gives some possible directions for future work,

Map-free Dataset	VCRE (90px)		Pose (25cm, 5°)	
	AUC	Prec. (%)	AUC	Prec. (%)
Pose Solvers				
SuperGlue [6, 22]				
Ess. Scale	**0.60**	**36.1**	**0.35**	**16.8**
PnP	**0.60**	36.0	0.25	10.7
LoFTR [23]				
Ess. Scale	0.61	**34.7**	**0.35**	**15.4**
PnP	**0.62**	33.4	0.27	9.8
MicKey w/ Overlap				
Ess. Scale	0.66	39.3	0.22	8.4
PnP	0.70	42.1	**0.36**	**14.6**
Our Solver	**0.75**	**49.2**	0.33	13.3
MicKey				
Ess. Scale	0.65	37.1	0.20	6.9
PnP	0.70	42.5	**0.33**	**12.8**
Our Solver	**0.74**	**49.2**	0.28	12.0

Table 6. **Pose solver ablation on Map-free**. Results show that state-of-the-art matchers work better when estimating the essential matrix from 2D-2D correspondences, and then recovering the metric scale from the depth predictor. MicKey, meanwhile, obtains the top VCRE results when recovering the pose with the probabilistic solver used during training.

where one could focus on improving MicKey's predictions under such strict thresholds.

Pose Solvers. Arnold *et al.* [2] propose different strategies for recovering the metric scale from the keypoint correspondences. In the first strategy, authors first compute the essential matrix and then rely on the depth estimation to obtain the scaled translation vector (Ess. Scale) [9, 18]. Their second strategy consisted of using the depth maps to lift the 2D keypoints to 3D, and then applying the Perspective-n-Point (PnP) algorithm [10]. We refer to [2] for more details. We compare such strategies in Table 6. Moreover, we also show MicKey's results with the two different proposed solvers. We demonstrate that obtaining poses with the same probabilistic approach we use during training yields the best results, proving the effectiveness of both, the end-to-end strategy and our probabilistic formulation of the metric relative pose estimation.

Cross-dataset evaluation. We also test the generalization capability of MicKey when trained and tested in different scenarios. We use MicKey trained in ScanNet dataset and evaluate it in the Map-free evaluation. Even though this experiment involves a significant distribution gap (indoor vs. outdoor), MicKey achieves an AUC (VCRE) score of 0.55, still outperforming DISK [24], SiLK [11], and DeDoDe [8], which all were trained on outdoor datasets (see Table 5 for all AUC (VCRE) results).

Monocular depth estimation. In Table 7, we further eval-

	DIML Outdoor [14]			DIODE Outdoor [25]		
	$\delta_1 \uparrow$	REL $\downarrow$	RMSE $\downarrow$	$\delta_1 \uparrow$	REL $\downarrow$	RMSE $\downarrow$
ZoeDepth [4]	0.29	0.64	3.61	**0.21**	0.76	**7.57**
MicKey	0.65	0.20	4.30	0.04	0.67	15.18
MicKey-Sc	**0.70**	**0.17**	**2.39**	0.04	**0.66**	13.76

Table 7. **Outdoor monocular depth evaluation.** We report the zero-shot generalization on outdoor datasets and see that MicKey provides competitive results even though it was not designed for this task.

	Map-free Dataset	
	VCRE	Median Errors
	AUC / Prec. (%)	Rep. / Trans / Rot
Hard pairs		
SuperGlue [6, 22]	0.07 / 3.5	271.8 / 4.5 / 81.6
LoFTR [23]	0.06 / 2.6	255.7 / 4.8 / 88.5
RoMa [7]	0.08 / 7.3	241.7 / **3.1** / **58.3**
MicKey (ours)	**0.15** / **11.1**	**233.7** / 3.7 / 75.2

Table 8. **Hard examples in the Map-free dataset.** We evaluate image pairs from the validation set that are taken under large viewpoint changes. We define such examples as image pairs that are at least 3m apart and have a 45° change in the camera direction. We report the VCRE metrics and the median errors of the estimated poses.

uate the generalization capabilities of our network also in the zero-shot monocular depth estimation task. We compute its accuracy in the DIML Outdoor [14] and the DIODE Outdoor [25] datasets. As a reference, we also provide ZoeDepth (NK) metrics. Similar to Section 3.1 (Table 4), we also show results for MicKey-Sc, where we evaluate depth prediction on the positions where MicKey is most confident (50% top scoring positions). MicKey has been trained with pedestrian smartphone images, and still, it can generalize and produce valid depth maps in datasets with different statistics, or visual conditions.

Inlier correspondences. In this last section, we show different visual examples in Figure 3. We plot the inlier correspondences that every method returns after computing the relative pose estimation. We observe that MicKey finds correct correspondences even though images were taken from extremely different viewpoints. Moreover, we see that MicKey detects and tries to match the object of interest within the image instead of relying on local patterns that might not appear in the two images. For instance, in images from row 1 or row 4, the object of interest is shown in the images from opposite views, *i.e.*, images were taken with almost a 180° difference. Even so, MicKey is able to match the correct side of the object to its corresponding part in the other image. Thus, we observe that the network is able to reason about the shape of the object and establish correspondences beyond local patterns. RoMa [7] is also able to find good matches, but it fails when the images do not have direct visual overlap. Contrary to LoFTR [23] or RoMA [7], MicKey and SuperGlue [22] are sparse feature methods, and then they only have access to a single image when computing their keypoints and descriptors. Contrary to MicKey, we note that state-of-the-art sparse feature methods (*e.g.*, SuperPoint [6]-SuperGlue [22]) do not find any good correspondences under such extreme cases.

We report the VCRE metrics and median errors on image pairs that have large and challenging viewpoint differences in Table 8. We use the validation scenes, where ground truth data is provided and hence, we can define the difficulty of an image pair. We rely on the pose difference between the reference and the query frame instead of the overlap score, such that unsolvable pairs are also evaluated. We define a hard example as a pair that is taken at least 3 meters apart and with their camera directions being at least rotated by 45°. We see that on those challenging examples, MicKey obtains the highest number of relative poses under the VCRE threshold (90 pixels).

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375*, 2018. 1

[2] 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. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2, 4

[3] Axel Barroso-Laguna, Eric Brachmann, Victor Adrian Prisacariu, Gabriel J Brostow, and Daniyar Turmukhambetov. Two-view geometry scoring without correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8979–8989, 2023. 2

[4] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. ZoeDepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 2, 5

[5] 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 *European Conference on Computer Vision*, pages 20–36. Springer, 2022. 4

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2, 4, 5, 6

[7] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Revisiting Robust Losses for Dense Feature Matching. *arXiv preprint arXiv:2305.15404*, 2023. 4, 5

[8] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. DeDoDe: Detect, Don't Describe – Describe, Don't Detect for Local Feature Matching. *International Conference on 3D Vision (3DV)*, 2024. 4

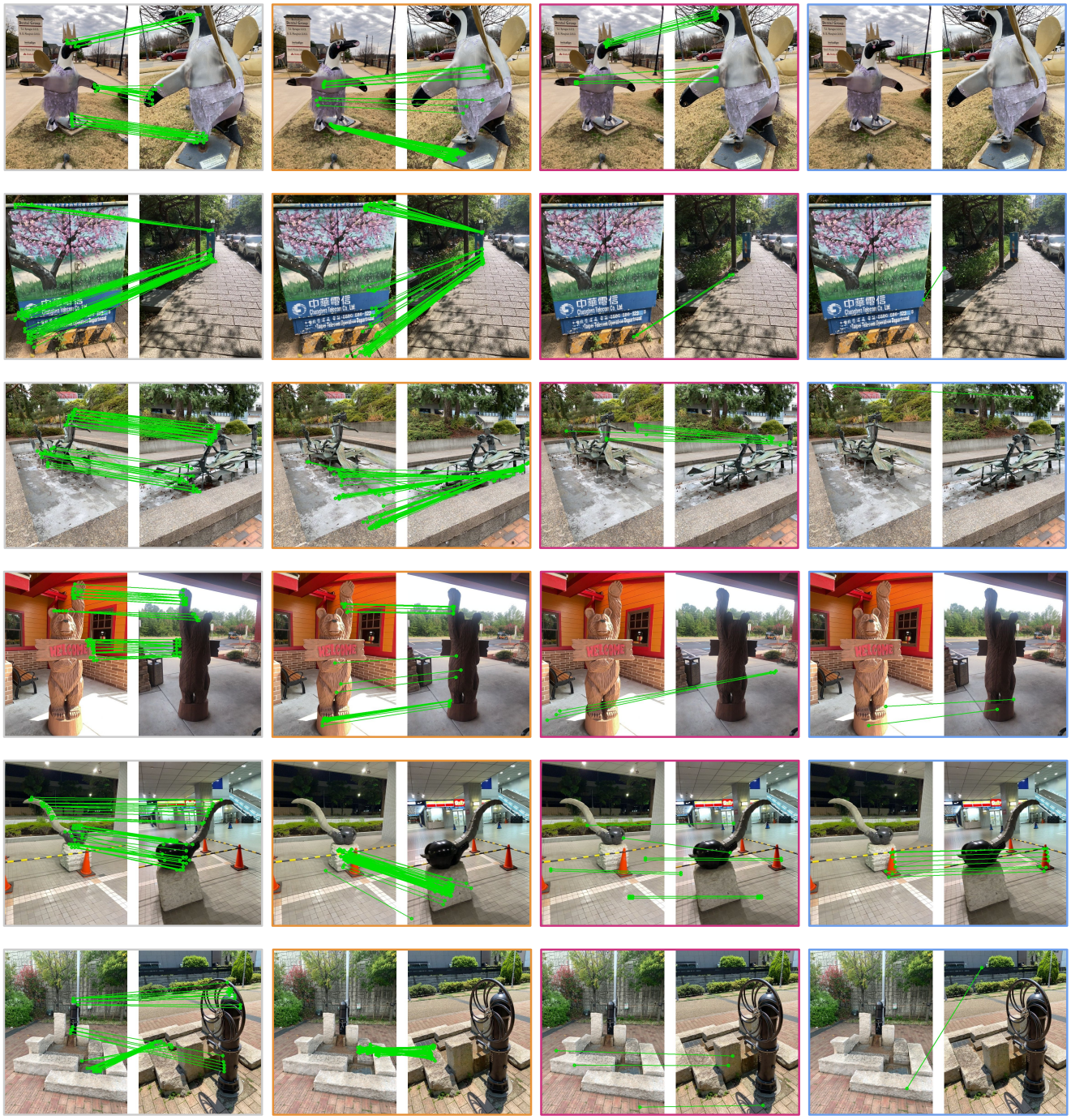

Figure 3. **Inlier correspondences on Map-free dataset.** We show the inlier correspondences returned by different feature extractors and their pose solvers. MicKey outperforms the other matchers when there are strong viewpoint changes between the two input images. MicKey embeds in a single neural network a feature representation of the keypoints, as well as their 3D geometry, allowing it to match keypoints where little overlap is observed. We see that SuperGlue [6, 22], a sparse feature method like MicKey, struggles to compute good correspondences when images present this kind of extreme viewpoint differences.

[9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[10] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003. 4

[11] Pierre Gleize, Weiyao Wang, and Matt Feiszli. SiLK–Simple Learned Keypoints. *Proceedings of the IEEE/CVF international conference on computer vision*, 2023. 4

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 1

[13] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 1

[14] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn. Deep monocular depth estimation via integration of global and local predictions. *IEEE transactions on Image Processing*, 27(8):4131–4144, 2018. 5

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[16] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local feature matching at light speed. *arXiv preprint arXiv:2306.13643*, 2023. 4

[17] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. PlaneRCNN: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019. 2

[18] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 4

[19] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[20] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 2

[21] Barbara Roessle and Matthias Nießner. End2End Multi-View Feature Matching with Differentiable Pose Optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 477–487, 2023. 2

[22] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 4, 5, 6

[23] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 2, 4, 5

[24] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 4

[25] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. DIODE: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 5