UniCorrn: Unified Correspondence Transformer Across 2D and 3D

Supplementary Material

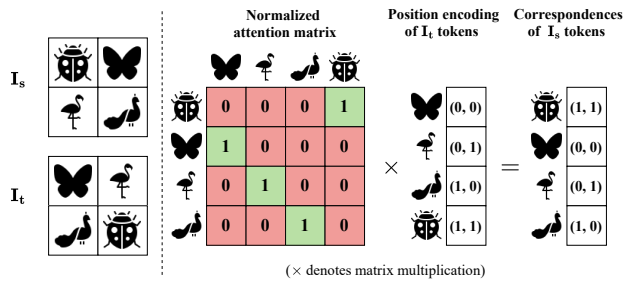


Figure 1. **Illustration of estimating correspondence with attention**. Here each animal symbol denotes a pixel (so both $\mathbf{I}_s$ and $\mathbf{I}_t$ have 2×2 pixels.).

A. Attention as a Learnable Matching Cost

In figure 1, we show an illustration of using attention to estimate correspondences with a toy example. Let's consider two input images $\mathbf{I}_s$ and $\mathbf{I}_t$. The attention map $\mathbf{A}$ between them is computed as the `Softmax`-normalized dot product of the flattened inputs. The attention map is row-normalized and one-hot in each row in an ideal case, where the position of 1 corresponds to the correct matching pixel. If we set the vector $\mathbf{V}$ in Transformer to the *absolute positional encoding* of every pixel in $\mathbf{I}_t$, as shown in Fig. 1, the output $\mathbf{AV}$ contains the positional encoding of the correct corresponding pixels in $\mathbf{I}_t$ for every pixel in $\mathbf{I}_s$.

The attention matrix is similar to the normalized version of the learnable cost volume studied in [26]. In practice, while features may not be perfectly discriminative as demonstrated in this example, the methodology of using attention matrix as a matching cost function still applies.

B. Further Details on Matching Decoder

B.1. Gaussian Attention

In the paper, we propose Gaussian attention in replace of vanilla attention [19] within our matching decoder. The attention logits are computed using pairwise squared `L2` distance is formulated as:

$$a_{ij} = -\frac{\|Q_i - K_j\|^2}{D}, \tag{1}$$

where Q and K are query and key tokens and D is the embedding dimension. Furthermore, if we took `Softmax` function into consideration to get the normalized attention scores, the equation becomes:

$$\mathbf{A}_{ij} = \frac{\exp(a_{ij})}{\sum_k \exp(a_{ik})}. \tag{2}$$

Here, $\exp(a)$ fits into the general formulation of the Gaussian kernel.

B.2. InfoNCE Loss

We provide further details of computing InfoNCE [18] loss. For a given pair of source and target feature descriptors $\mathbf{F}_s^{desc}$ and $\mathbf{F}_t^{desc}$, respectively, the InfoNCE loss over the set of ground-truth correspondences $\mathcal{M} = \{\bar{\mathbf{K}}_s(i), \bar{\mathbf{K}}_t(i)\}_{i=1}^N$ is given by:

$$\begin{aligned}\mathcal{L}_c(\mathbf{F}_s^{desc}, \mathbf{F}_t^{desc}) = &-\sum_{i=1}^{N} \log \frac{d(\bar{\mathbf{K}}_s(i), \bar{\mathbf{K}}_t(i))}{\sum_{j=1}^{N} d(\bar{\mathbf{K}}_s(j), \bar{\mathbf{K}}_t(i))} \\ &+ \log \frac{d(\bar{\mathbf{K}}_s(i), \bar{\mathbf{K}}_t(i))}{\sum_{j=1}^{N} d(\bar{\mathbf{K}}_s(i), \bar{\mathbf{K}}_t(j))},\end{aligned} \tag{3}$$

$$\text{with } d(\bar{\mathbf{K}}_s, \bar{\mathbf{K}}_t) = \tau^{-1} ||\mathbf{F}_s^{desc}(\bar{\mathbf{K}}_\mathbf{s}) - \mathbf{F}_t^{desc}(\bar{\mathbf{K}}_t)||_2,$$

where τ is a temperature hyperparameter. Similarly, we compute the InfoNCE loss for $\mathcal{L}_c(\mathbf{F}_k, \mathbf{F}_t^{desc})$.

B.3. Pseudo Point Cloud Data

In Table 1, we show the effectiveness of using pseudo point cloud data for the 2D3D and 3D3D tasks. The pseudo point cloud is generated from dense depth maps, where depth is projected to dense 3D points and sampled with equal strides to resemble the sparse structure of the 3D benchmark datasets. As our approach is data-driven, jointly training with pseudo-point cloud data enables our model to reach SOTA performance.

Table 1. **Effectiveness of pseudo point cloud data** for 2D-3D and 3D-3D task. The pseudo data is sampled from ScanNet++ [28] depth maps.

Pseudo Point Cloud	7Scenes (2D-3D)			3DLoMatch (3D-3D)		
	IR ↑	FMR ↑	RR ↑	IR ↑	FMR ↑	RR ↑
✗	12.9	49.5	15.4	51.1	83.2	73.2
✓	**66.3**	**88.2**	**77.8**	**70.5**	**90.1**	**81.8**

B.4. Auxiliary Supervision

In our training objective, we use intermediate predictions by applying the attention matrix directly over the target coordinates for auxiliary supervision. As shown in Tab. 2, the auxiliary loss produced substantial performance improvement

with a single matching decoder layer and also improved the results while scaling up the number of layers. In Figure. 2 we visualize the attention heatmaps for each decoder layer along with the final predicted coordinates from the model. The heatmaps show a clear difference: without auxiliary supervision, attention patterns are random across layers, while with auxiliary supervision, query tokens consistently attend to their corresponding predicted coordinates. This shows how the dual-stream attention propagates through the matching decoder layers.

Table 2. **Effectiveness of auxiliary loss $\mathcal{L}_{aux}$.**

Number of Layers	$\mathcal{L}_{aux}$	MegaDepth-1500		
		5° ↑	10° ↑	20° ↑
1	✗	28.8	45.3	61.3
1	✓	47.7	64.2	77.2
5	✗	48.5	65.1	77.9
5	✓	50.6	67.1	79.6

B.5. Additional details on model and training

We train two models with two different capacities. For the small-scale model, we employ 12-layer ViT [4] and PTv3 [24] transformers as image and point cloud backbones, respectively, along with an 8-layer shared Transformer for feature fusion encoder. We ablate various configurations of our matching transformer decoder using this setup in Section **??**. The large-scale model extends these architectures to 24 and 14 layers for the ViT and PTv3 backbones, and 12 and 8 layers for the feature fusion encoder and matching decoder, respectively. We train the large-scale unified model (600M parameters) in two stages. In the first stage, the model is initialized with the pre-trained weights of CroCo v2 [22] and jointly trained on 2D-2D and 3D-3D tasks with the AdamW optimizer for 40 epochs. This stages uses $384,000$ 2D-2D pairs and $384,000$ of 3D-3D pairs. The second stage is trained on all three tasks for 30 epochs with $60,000$ samples per task per epoch. The input images are resized to 512×384 for the 2D-2D and 2D-3D tasks. The training runs on $8\times$H100 GPUs with stage 1 taking 7 days and stage 2 taking 4 days.

In Table 4, we provide the configurations of each module for our small and large-scale models. Table 5 contains the hyperparameters used for the two stage large-scale training. Finally, Table 6 shows the mixture of 2D-2D, 2D-3D and 3D-3D datasets along with the pseudo data samples used in each stage of large-scale training. We further oversample the 2D-3D and 3D-3D pairs to match the total number of pairs used for the 2D-2D task so that the model can be jointly trained.

C. Generalization to unseen correspondence tasks

Our model may generalize to other geometry matching tasks, like optical flow without any fine-tuning. On the Sintel final training split, our model achieves an end-point error (EPE) of 5.2 with zero-shot inference (specialist model RAFT reports EPE of 2.71). This is significant because our model was trained exclusively on static, photorealistic imagery, making Sintel's dynamic motion and stylized rendering strictly out-of-distribution. For other correspondence task, like semantic matching, fine-tuning is required. In fact, unifying both geometric and semantic understanding with a single model by training on all different data is an exciting direction to go.

D. Inference time and memory usage

The memory footprint of our unified model is $\sim$ $2.6G$ which is $3.5\times$ less than the combined memory usage of the specialized models. We report the inference time comparisons with specialized models in the table below, measured on an RTX A5000.

Table 3. Inference time in milliseconds(ms) on RTX A5000. Our method uses 5000 keypoint queries. Diff-Reg [23] uses existing models for 2D-3D [10] and 3D-3D [11] feature descriptors.

	ScanNet (2D-2D)	7Scenes (2D-3D)	3DMatch (3D-3D)
Ours	329 ms	390 ms	320 ms
Specialized	203 ms (RoMa)	1140 ms (Diff-Reg)	603 ms (Diff-Reg)

E. Additional visual results

We show qualitative comparison with state-of-the-art 2D-2D matching methods RoMa [5] and MASt3R [9] in Figure 5. Figure 4 shows the correspondences for different confidence thresholds on two examples from the InLoc [17] benchmark. Additionally, we provide visual results for 2D-3D and 3D-3D in Figure 3 and Figure 6, respectively.

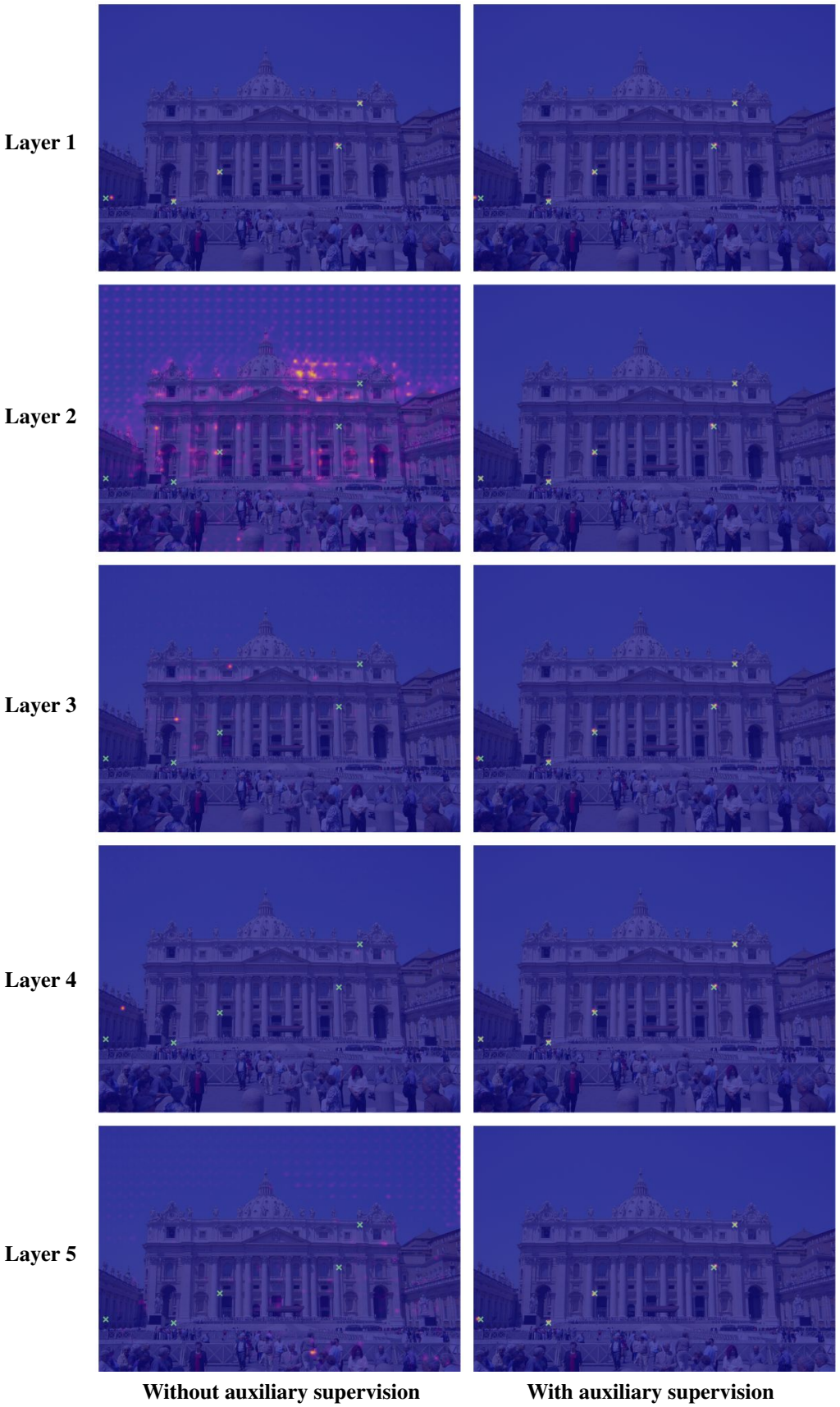


Figure 2. **Per-layer attention heatmap comparison for the effectiveness of auxiliary supervision**. Green markers indicates the model's predicted coordinates. Zoom in for more details.

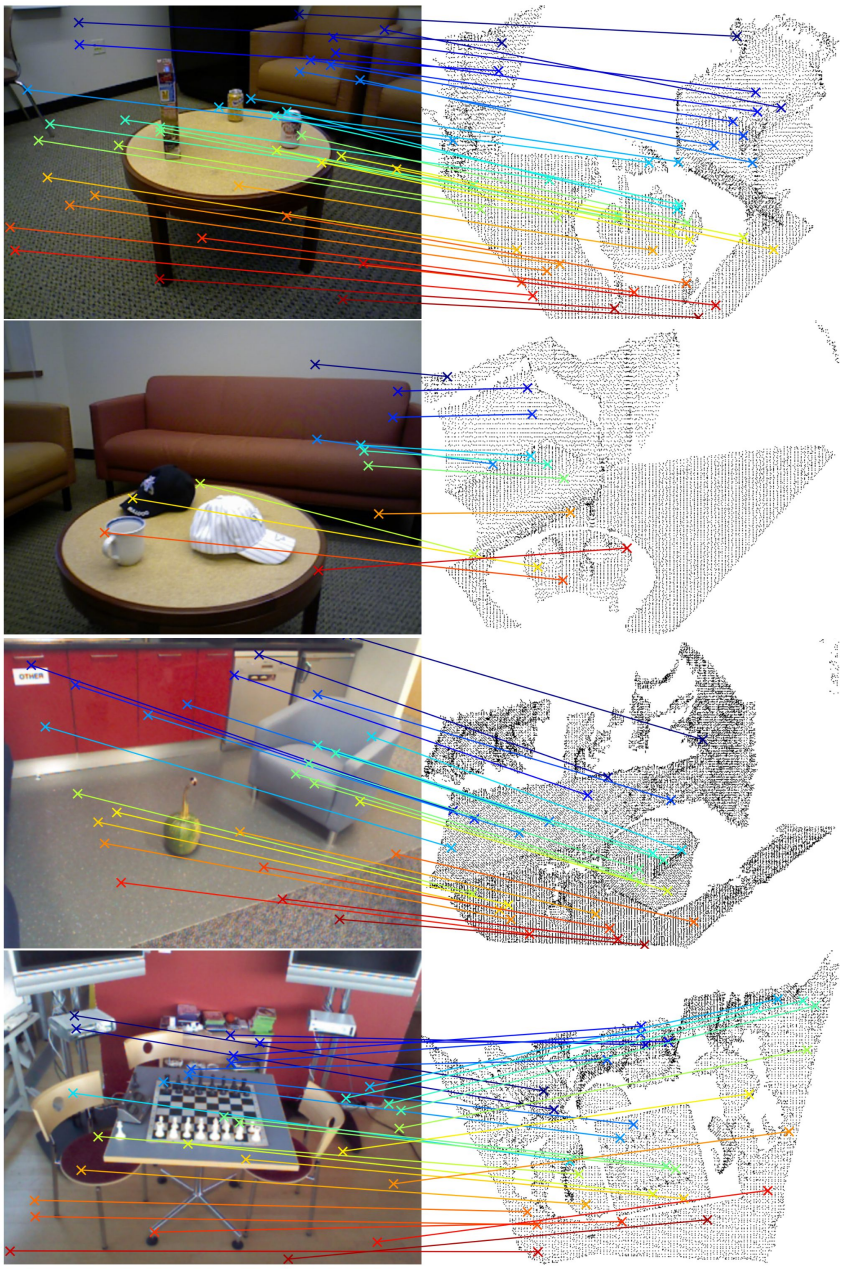


Figure 3. **Visual results of 2D-3D matching on 3DMatch (top) and 3DLoMatch (bottom)**. The top two rows are from the RGB-Scenes V2 [8] and the bottom two rows are from 7Scenes [6].

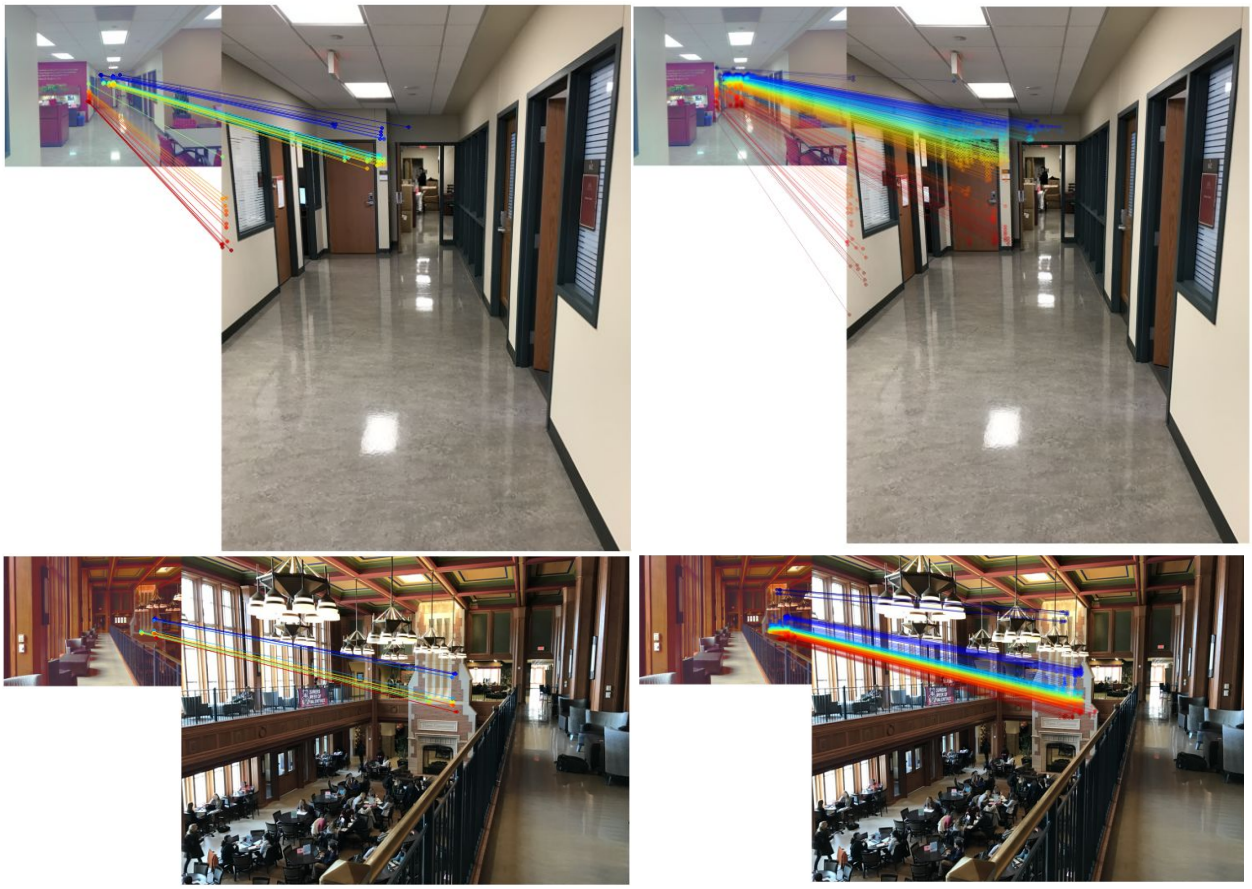
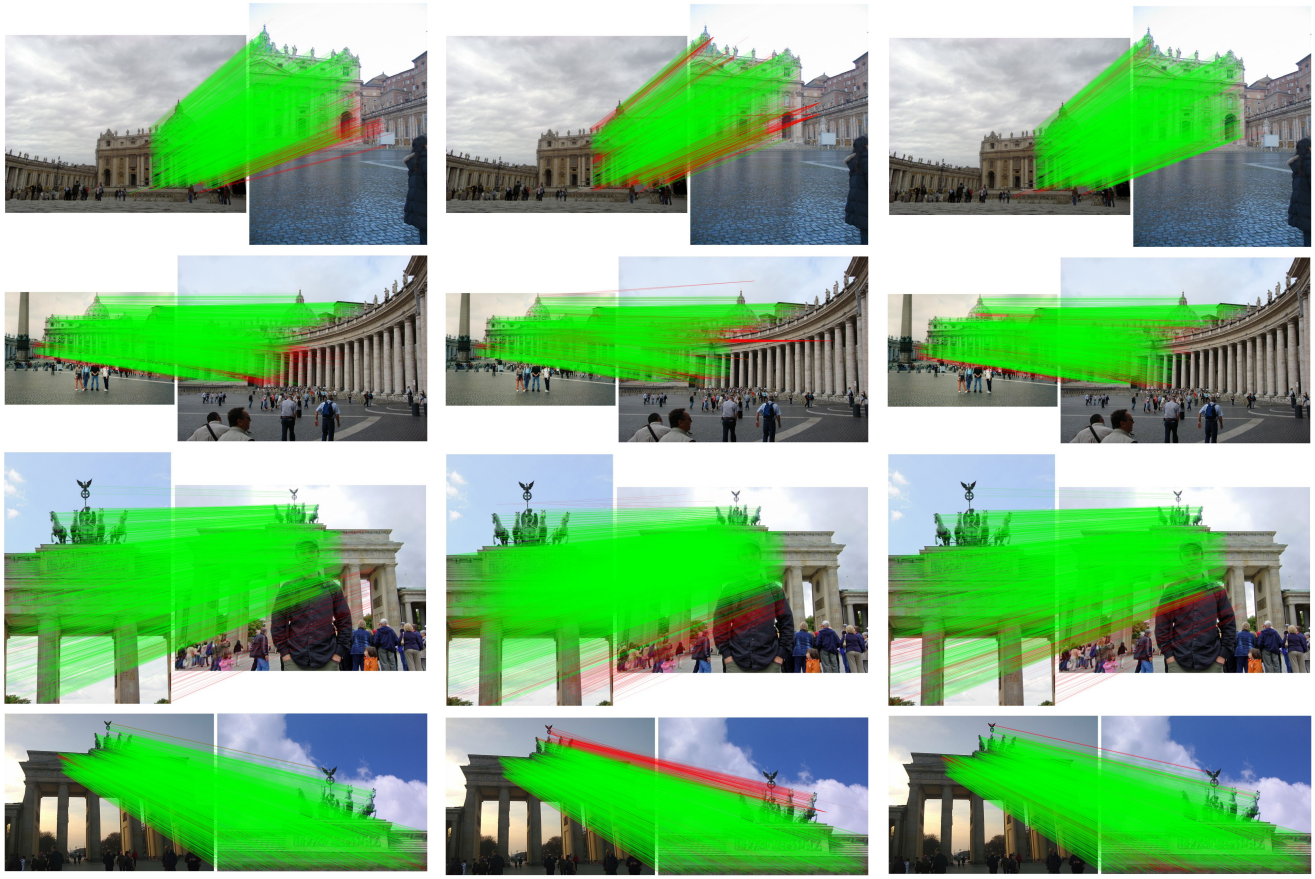


Figure 4. **Visual results on two examples from the InLoc [17]) Benchmark**. We show the correspondences for different confidence thresholds. Zoom in for details.



RoMA [5] MASt3R [9] **UniCorrn (Ours)**

Figure 5. **2D-2D qualitative comparisons on the MegaDepth-1500 benchmark.** Green and red lines indicate accepted and rejected correspondences by the RANSAC essential matrix estimation, respectively. Zoom in for details.

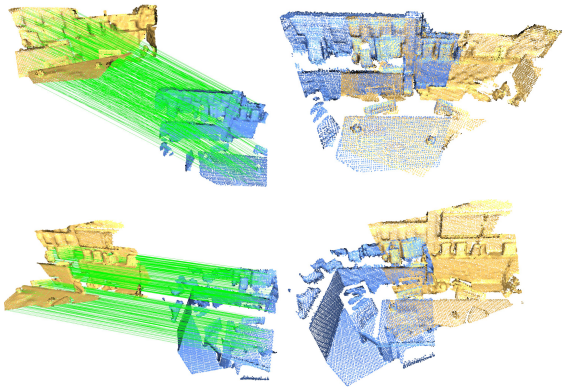


Figure 6. **Visual results of 3D-3D matching on 3DMatch (top) and 3DLoMatch (bottom)**. On the left are point cloud pairs with predicted correspondences, and on the right are registered point clouds using transformations estimated via RANSAC.

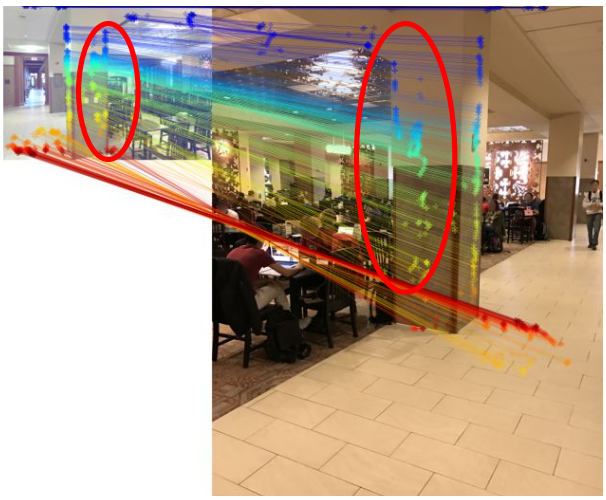


Figure 7. **Failure case on InLoc [17] benchmark.** The correspondences inside the red ellipse are invalid since the pillar's face on the first image is not visible on the second image. Hence these correspondences would yield incorrect geometry.

Table 4. **Detailed architecture** of our small-scale and large-scale model.

Module	Type	Attribute	Size
UniCorrn (small-baseline)			
Image backbone	ViT-B [4]	Depth	12
		Heads	12
		Embedding dims	768
Point cloud backbone	PTv3 [24]	Depth	[2, 2, 6, 2]
		Heads	[4, 8, 16, 32]
		Embedding dims	[64, 128, 256, 512]
Feature fusion encoder	Cross-view [22]	Depth	8
		Heads	16
		Embedding dims	512
Matching decoder	Dual-stream (ours)	Depth	8
		Heads	16
		Embedding dims	256
UniCorrn (small-final)			
Image backbone	ViT-B [4]	Depth	12
		Heads	12
		Embedding dims	768
Point cloud backbone	PTv3 [24]	Depth	[2, 6, 4]
		Heads	[2, 8, 32]
		Embedding dims	[32, 128, 512]
Feature fusion encoder	Cross-view [22]	Depth	8
		Heads	16
		Embedding dims	512
Matching decoder	Dual-stream (ours)	Depth	8
		Heads	1
		Embedding dims	256
UniCorrn (large)			
Image backbone	ViT-L [4]	Depth	24
		Heads	16
		Embedding dims	1024
Point cloud backbone	PTv3 [24]	Depth	[3, 6, 6]
		Heads	[2, 8, 32]
		Embedding dims	[32, 128, 512]
Feature fusion encoder	Cross-view [22]	Depth	12
		Heads	16
		Embedding dims	768
Matching decoder	Dual-stream (ours)	Depth	8
		Heads	1
		Embedding dims	256

Table 5. **Hyper-parameters** for large-scale stage 1 and stage 2 training.

Hyperparameters	Stage 1	Stage 2
Tasks	2D-2D, 3D-3D	2D-2D, 2D-3D, 3D-3D
Optimizer	AdamW [13]	AdamW [13]
Base learning rate	1e-4	2e-5
Minimum learning rate	1e-7	1e-7
Weight decay	0.01	0.01
Adam β	(0.9, 0.95)	(0.9, 0.95)
Batch size (per task)	24	16
Epochs	40	30
Warmup epochs	4	5
Learning rate scheduler	Cosine decay	Cosine decay
Gradient norm clipping	1.0	1.0
Pre-trained weights	CroCo V2 [22]	Ours (Stage 1)

Table 6. Dataset sample sizes for large-scale training.

Dataset	Type	Pairs per epoch
2D-2D (stage 1)		
ArkitScenes [1]	Indoor / Real	45,600
BlendedMVS [27]	Mixed / Synthetic	68,400
CO3Dv2 [15]	Object-centric / Real	22,800
MegaDepth [12]	Outdoor / Real	68,400
Static Things 3D [14]	Object / Synthetic	22,800
ScanNet++ [28]	Indoor / Real	60,000
Waymo [16]	Outdoor / Real	60,000
3D-3D (stage 1)		
3DMatch [29]	Indoor / Real	20,586
ModelNet [25]	Object-centric / Synthetic	5,112
ArkitScenes [1]	Indoor / Real	80,000
MegaDepth [12]	Outdoor / Real	80,000
ScanNet++ [28]	Indoor / Real	80,000
2D-2D (stage 2)		
MegaDepth [12]	Outdoor / Real	20,000
ScanNet++ [28]	Indoor / Real	20,000
2D-3D (stage 2)		
7Scenes [6]	Indoor / Real	4,048
RGB-D Scenes V2 [8]	Indoor / Real	1,748
ScanNet++ [28]	Indoor / Real	10,000
3D-3D (stage 2)		
3DMatch [29]	Indoor / Real	20,586
ModelNet [25]	Object-centric / Synthetic	5,112
ArkitScenes [1]	Indoor / Real	10,000
ScanNet++ [28]	Indoor / Real	20,000

Table 7. Evaluation results on RGB-D Scenes V2 [8]. **Boldfaced** numbers highlight the best and the second best are underlined.

Model	Scene-11	Scene-12	Scene-13	Scene-14	Mean
Mean depth (m)	1.74	1.66	1.18	1.39	1.49
Inlier Ratio(IR) ↑					
FCGF-2D3D [3]	6.8	8.5	11.8	5.4	8.1
P2-Net [20]	9.7	12.8	17.0	9.3	12.2
Predator-2D3D [7]	17.7	19.4	17.2	8.4	15.7
2D3D-MATR [10]	32.8	34.4	39.2	23.3	32.4
B2-3Dnet [2]	36.4	32.7	43.8	27.4	35.1
FreeReg [21]	36.6	34.5	34.2	18.2	30.9
Ours (stage 2)	**85.7**	**86.7**	**92.5**	**69.1**	**83.6**
Feature Matching Recall (FMR) ↑					
FCGF-2D3D [3]	11.1	30.4	51.5	15.5	27.1
P2-Net [20]	48.6	65.7	82.5	41.6	59.6
Predator-2D3D [7]	86.1	89.2	63.9	24.3	65.9
2D3D-MATR [10]	98.6	98.0	88.7	77.9	90.8
B2-3Dnet [2]	**100.0**	**99.0**	92.8	85.8	94.4
FreeReg [21]	91.9	93.4	93.1	49.6	82.0
Ours (stage 2)	98.6	97.2	**100.0**	**92.0**	**97.0**
Registration Recall (RR) ↑					
FCGF-2D3D [3]	26.4	41.2	37.1	16.8	30.4
P2-Net [20]	40.3	40.2	41.2	31.9	38.4
Predator-2D3D [7]	44.4	41.2	21.6	13.7	30.2
2D3D-MATR [10]	63.9	53.9	58.8	49.1	56.4
B2-3Dnet [2]	58.3	60.8	74.2	60.2	63.4
FreeReg [21]	74.2	72.5	54.5	27.9	57.3
Diff-Reg [23]	95.8	**96.1**	88.7	69.0	87.4
Ours (stage 2)	**98.6**	95.3	**99.0**	**76.9**	**92.5**

Table 8. Evaluation results on 7Scenes [6]. **Boldfaced** numbers highlight the best and the second best are underlined.

Model	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Mean
Mean depth (m)	1.78	1.55	0.80	2.03	2.25	2.13	1.84	1.77
Inlier Ratio (IR) ↑								
FCGF-2D3D [3]	34.2	32.8	14.8	26.0	23.3	22.5	6.0	22.8
P2-Net [20]	55.2	46.7	13.0	36.2	32.0	32.8	5.8	31.7
Predator-2D3D [7]	34.7	33.8	16.6	25.9	23.1	22.2	7.5	23.4
2D3D-MATR [10]	72.1	66.0	31.3	60.7	50.2	52.5	18.1	50.1
B2-3Dnet [2]	73.8	66.7	33.1	61.7	50.8	52.3	18.1	50.9
Diff-Reg [23]	79.2	71.0	54.1	70.4	55.8	60.2	22.9	59.1
Ours (stage 2)	**93.7**	**91.2**	**92.3**	**94.8**	**80.5**	**87.3**	**36.7**	**82.4**
Feature Matching Recall (FMR) ↑								
FCGF-2D3D [3]	99.7	98.2	69.9	97.1	83.0	87.7	16.2	78.8
P2-Net [20]	**100.0**	99.3	58.9	99.1	87.2	92.2	16.2	79.0
Predator-2D3D [7]	91.3	95.1	76.7	88.6	79.2	80.6	31.1	77.5
2D3D-MATR [10]	**100.0**	99.6	98.6	**100.0**	92.4	95.9	58.1	92.1
B2-3Dnet [2]	**100.0**	**100.0**	98.6	**100.0**	**92.7**	95.6	**64.9**	**93.1**
Diff-Reg [23]	**100.0**	**100.0**	**100.0**	**100.0**	91.3	98.1	58.1	92.5
Ours (stage 2)	**100.0**	**100.0**	**100.0**	**100.0**	90.5	**99.9**	60.7	93.0
Registration Recall (RR) ↑								
FCGF-2D3D [3]	89.5	79.7	19.2	85.9	69.4	79.0	6.8	61.4
P2-Net [20]	96.9	86.5	20.5	91.7	75.3	85.2	4.1	65.7
Predator-2D3D [7]	69.6	60.7	17.8	62.9	56.2	62.6	9.5	48.5
2D3D-MATR [10]	96.9	90.7	52.1	95.5	80.9	86.1	28.4	75.8
B2-3Dnet [2]	98.3	90.5	56.2	96.4	84.0	86.1	32.4	77.7
Diff-Reg [23]	**100.0**	94.0	90.4	99.3	81.2	94.6	27.0	83.8
Ours (stage 2)	**100.0**	**99.3**	**98.6**	**100.0**	**88.8**	**98.5**	**51.9**	**91.0**

References

[1] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. ARKitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 7

[2] Zhixin Cheng, Jiacheng Deng, Xinjun Li, Baoqun Yin, and Tianzhu Zhang. Bridge 2d-3d: Uncertainty-aware hierarchical registration network with domain alignment. In *AAAI*, pages 2491–2499. AAAI Press, 2025. 7, 8

[3] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, pages 8958–8966, 2019. 7, 8

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021. 2, 7

[5] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *CVPR*, pages 19790–19800. IEEE, 2024. 2, 6

[6] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179, 2013. 4, 7, 8

[7] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4267–4276, 2021. 7, 8

[8] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3d scene labeling. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3050–3057, 2014. 4, 7

[9] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV (72)*, pages 71–91. Springer, 2024. 2, 6

[10] Minhao Li, Zheng Qin, Zhirui Gao, Renjiao Yi, Chenyang Zhu, Yulan Guo, and Kai Xu. 2d3d-matr: 2d-3d matching transformer for detection-free registration between images and point clouds. In *ICCV*, pages 14082–14092. IEEE, 2023. 2, 7, 8

[11] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5544–5554. IEEE, 2022. 2

[12] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 7

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 7

[14] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 7

[15] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotný. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. *CoRR*, abs/2109.00512, 2021. 7

[16] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2443–2451. Computer Vision Foundation / IEEE, 2020. 7

[17] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomás Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7199–7209. Computer Vision Foundation / IEEE Computer Society, 2018. 2, 5, 6

[18] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 1

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[20] Bing Wang, Changhao Chen, Zhaopeng Cui, Jie Qin, Chris Xiaoxuan Lu, Zhengdi Yu, Peijun Zhao, Zhen Dong, Fan Zhu, Niki Trigoni, et al. P2-net: Joint description and detection of local features for pixel and point matching. In *ICCV*, pages 16004–16013, 2021. 7, 8

[21] Haiping Wang, Yuan Liu, Bing Wang, Yujing Sun, Zhen Dong, Wenping Wang, and Bisheng Yang. Freereg: Image-to-point cloud registration leveraging pretrained diffusion models and monocular depth estimators. In *ICLR*. OpenReview.net, 2024. 7

[22] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *ICCV*, pages 17923–17934. IEEE, 2023. 2, 7

[23] Qianliang Wu, Haobo Jiang, Yaqing Ding, Lei Luo, Jin Xie, and Jian Yang. Diff-reg v2: Diffusion-based matching matrix

estimation for image matching and 3d registration. *CoRR*, abs/2503.04127, 2025. 2, 7, 8

[24] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: simpler, faster, stronger. In *CVPR*, pages 4840–4851. IEEE, 2024. 2, 7

[25] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 7

[26] Taihong Xiao, Jinwei Yuan, Deqing Sun, Qifei Wang, Xin-Yu Zhang, Kehan Xu, and Ming-Hsuan Yang. Learnable cost volume using the cayley representation. In *ECCV*, 2020. 1

[27] 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. *CoRR*, abs/1911.10127, 2019. 7

[28] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 1, 7

[29] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017. 7