

Long-Term Invariant Local Features via Implicit Cross-Domain Correspondences

Zador Pataki, Mohammad Altillawi, Menelaos Kanakis, Rémi Pautrat,
Fengyi Shen, Ziyuan Liu, Luc Van Gool, and Marc Pollefeys

Abstract—Modern learning-based visual feature extraction networks perform well in intra-domain localization, however, their performance significantly declines when image pairs are captured across long-term visual domain variations, such as different seasonal and daytime variations. In this paper, our first contribution is a benchmark to investigate the performance impact of long-term variations on visual localization. We conduct a thorough analysis of the performance of current state-of-the-art feature extraction networks under various domain changes and find a significant performance gap between intra- and cross-domain localization. We investigate different methods to close this gap by improving the supervision of modern feature extractor networks. We propose a novel data-centric method, Implicit Cross-Domain Correspondences (iCDC). iCDC represents the same environment with multiple Neural Radiance Fields, each fitting the scene under individual visual domains. It utilizes the underlying 3D representations to generate accurate correspondences across different long-term visual conditions. Our proposed method enhances cross-domain localization performance, significantly reducing the performance gap. When evaluated on popular long-term localization benchmarks, our trained networks consistently outperform existing methods. This work serves as a substantial stride toward more robust visual localization pipelines for long-term deployments, and opens up research avenues in the development of long-term invariant descriptors.

Index Terms—Local Features, NeRF, Long-term Localization

I. INTRODUCTION

VISUAL localization acts as a foundation for many applications, ranging from the navigation of autonomous vehicles to the association of the real and virtual worlds inside, such as placing virtual objects in the user’s field of view. A crucial step in tackling the visual localization problem is data association, where salient points in images are detected and described to establish keypoint correspondences. These correspondences facilitate relative pose estimation of images across varied viewpoints and enable the 3D reconstruction of scenes and the estimation of the 6 Degrees of Freedom (DoF) of the camera pose of a query image relative to a 3D map.

Z. Pataki, R. Pautrat, and M. Pollefeys are with the Computer Vision and Geometry lab, Department of Computer Science, ETH Zurich

M. Altillawi is with the Computer Vision Center CVC-Barcelona and the Intelligent Robotics Cloud Technology lab of Huawei-Munich

M. Kanakis is with the Computer Vision Lab, Department Electrical Engineering, ETH Zurich

F. Shen is with the Chair of Robotics, Artificial Intelligence and Embedded Systems at TU Munich (TUM) and the Intelligent Robotics Cloud Technology lab of Huawei-Munich

Ziyuan Liu is with the Intelligent Robotics Cloud Technology lab of Huawei-Munich

L. Van Gool is with the Center for Processing Speech and Images, KU Leuven and the Computer Vision Lab, ETH Zurich

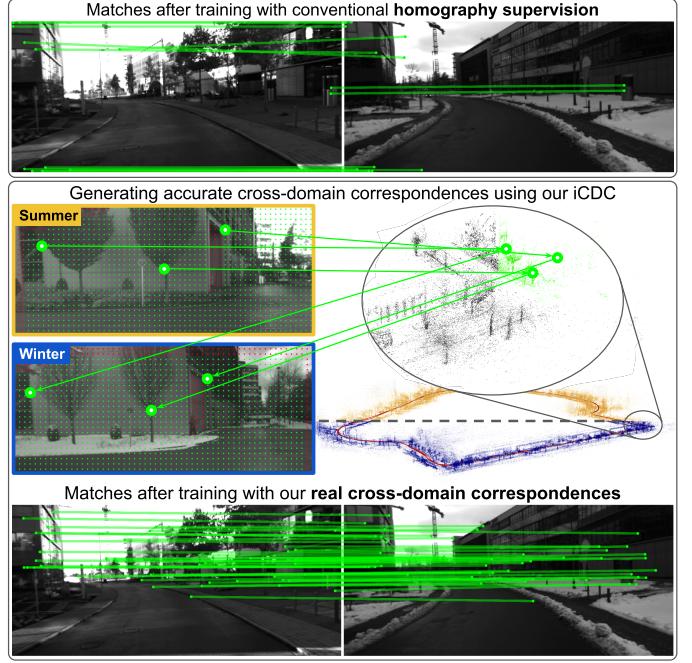


Fig. 1. We propose iCDC, a method to supervise feature extractors across different domains. Traditionally, correspondences are generated by augmenting, and applying homographic transformations to images. iCDC generates accurate correspondences across images captured under different views and in different visual domains.

The quality of data association heavily relies on the accuracy and robustness of local feature extraction and description. Initially, sparse features were identified using hand-crafted methods [1], [2], [3], [4], which have seen success in various applications, including Structure-from-Motion (SfM) [5] and Simultaneous Localization and Mapping (SLAM) [6]. However, significant advancements in Deep Neural Networks (DNNs), have enabled the learning of robust, and highly descriptive image features [7], [8], [9], [10], [11], pushing the boundaries of what was previously possible [12], [13], [14].

With the ultimate goal being the successful life-long deployment in the real world, robustness to visual changes caused by time of day and season is evaluated using benchmarks such as long-term localization [12], or sparse SLAM under different weather conditions [15]. In life-long deployment scenarios, accurate 6 DoF localization relies on localization with respect to a prior 3D map. Herein, the key challenge comes from the often differing visual domains of the descriptors in the map and the query. While model-centric methods explored in the state of the art are crucial, we hypothesize that further

exploration of methods to better exploit available data (also known as data-centric methods) is necessary to achieve accurate localization performance across long-term variations.

In this work, we aim to understand and better optimize the robustness of feature extraction networks to long-term visual domain changes. To conduct a thorough evaluation of long-term localization performance, we enhance the 4Seasons dataset [15] by refining the poses. This dataset contains a large amount of data captured along trajectories across various visual domains and scenes; it lacks consistency in the camera poses across the different trajectories. By refining poses in a cross-trajectory and cross-domain optimization problem, we can accurately compare the relative poses of frames across trajectory and domain pairs of scenes. Using our refined maps, we benchmark current extractors [7], [8], [9], [16], [11].

Our experiments reveal a significant performance gap between intra-domain and cross-domain localization, regardless of the domain types. To bridge this gap, we explore and categorize data-centric approaches by supervising feature extractors using cross-domain correspondences. Through a comparative analysis, we identify the limitations of the approaches and propose a novel data-centric method, Implicit Cross-Domain Correspondences (iCDC). By representing an environment using multiple, independent, and domain-specific Neural Radiance Fields (NeRFs), iCDC leverages their underlying 3D representations and generates highly accurate cross-domain correspondences. We train state-of-the-art networks [8], [11] using iCDC, and improve their cross-domain localization performances on our benchmark, and on other long-term localization benchmarks [17], [18], [19].

In summary, our contributions are the following:

- We refine the ground-truth maps of 4Seasons, and establish a new benchmark for long-term localization.
- Our benchmarking of existing feature extraction networks reveals a significant performance gap between intra- and cross-domain localization.
- We find that supervising extractors using cross-domain correspondences between real viewpoint changes reduces the performance gap.
- We propose iCDC to generate accurate cross-domain correspondences. Training extractors using iCDC further improves cross-domain localization.

II. RELATED WORK

Hand-crafted feature extractors. Hand-crafted extractors such as SIFT [1], SURF [2], BRISK [3], BRIEF [20], ORB [4] aim to identify and describe distinctive structures or patterns in images. SIFT is well-regarded for its invariance to scale, rotation, and affine transformations, and has been widely used in applications such as object recognition [1] and SfM [5]. Inspired by SIFT, SURF improved the computation speed of the feature extraction while maintaining robustness to scale and rotation changes, by approximating the Laplacian of Gaussian with a box filter representation. On the other hand, some methods [3], [20], [4] focus instead on the generation of binary descriptors for use in computationally limited platforms such as robots and Augmented Reality (AR). While

these hand-crafted methods have been successful in a range of applications, they often struggle with perceptual changes caused by variations in illumination, weather, and seasons [12].

Learned feature extractors. Motivated by the limitations of hand-crafted methods and the success of DNNs, learning based feature extractors [7], [8], [9], [10] have improved performance, more noticeably in the aforementioned challenging perceptual changes [12]. A number of works explored better architecture designs to enhance performance. Notably, R2D2 [8] and D2-Net [9] propose networks that share all parameters between detection and description. SuperPoint [7], and KP2D [16], jointly learn interest point detection and description using two decoding branches. Furthermore, other works aimed to improve the optimization of the networks through outlier rejection [16], introducing a novel reliability score to guide the detection process [8], contrastive learning [21], using guidance from SfM to obtain more robust keypoints [22], or by employing reinforcement learning principles [23]. Recent works [10], [24] investigated the possibility of making these networks more efficient for deployment in computationally limited platforms, while others [11] focused on improving retrainability; a limitation of DISK [23] which is notoriously difficult to train.

A common component amongst these works, with the exception of DISK, is the use of self-supervised methods, which facilitate the flexible optimization of the networks, enabling learning from diverse and large-scale datasets, such as COCO [25]. This is accomplished by first applying non-geometric transformations, such as color augmentations, as well as a homographic transformation to every input image, and then optimizing the network to generate identical descriptors for the same regions in the original and transformed images. While this method has proven very powerful in practice, invariance to perceptual changes caused by weather and seasonal variations is not directly enforced, potentially hindering their performance under challenging perceptual changes caused by the time of day, season, and weather conditions. In this work, we aim to evaluate the robustness of state-of-the-art extractors in both intra- and cross-domain scenarios, exploring avenues for further improvement.

Invariance to long-term changes. Due to the importance of life-long deployment of computer vision systems, a number of different approaches have been proposed over the years to close the cross-domain performance gap for different applications. This includes the optimization of networks by directly supervising cross-domain information through contrastive learning [26], [27], [28]. This has been explored in image retrieval and place recognition networks that output a single descriptor to represent the image, where generating ground-truth labels is made possible through GPS signals available in large datasets [29], [26]. For a number of applications, however, generating ground-truth labels is expensive. This has given rise to domain adaptation [30], [31], [32] and generalization [33], [34] methods, allowing the optimization of networks on datasets where labeled data can be more easily acquired, such as simulations, while minimizing the domain gap with real-life data. This technique has been widely adopted

in semantic segmentation [35], [36], [37], [38], [39] and image classification [40], [41], [42] tasks.

More similar to our work, WarpC [43] employs an unsupervised warping consistency to improve correspondence learning. In contrast, other methods [44], [8] use style transfer [45] instead, to alter the domain of the images, such as converting a summer image into a winter image. We, on the other hand, propose to directly enforce cross-domain similarity of descriptors by leveraging a large-scale dataset that captured the same trajectories at different weather conditions [15]. This is facilitated through our novel method iCDC, a correspondence generation method for real cross-domain images.

Long-term datasets. Existing research has shown feature extractor methods to be highly effective for accurate localization within the same domain [12], [46]. A growing body of literature seeks to address the challenge of long-term localization in situations that require matching descriptors across different domains. Benchmark datasets like the Aachen Day-Night dataset [17] focus on localizing query night-time images with respect to a day-time map, while RobotCar Seasons [18] benchmarks visual localization across an array of query domains against a single reference map capturing a single domain. Notably, its images are of relatively low quality, with night-time images suffering from significant motion blur. On the other hand, CMU-Seasons [19] benchmarks localization across many visual domains, albeit primarily in rural areas, with large portions of the images capturing vegetation. Although these benchmarks contribute to the research on cross-domain localization, the range of domain pairs and scenes examined remains limited.

On the other hand, the 4Seasons dataset [15] encompasses data across multiple trajectories, captured under a variety of weather, seasonal conditions, and in diverse scenes across different environments. However, it lacks reliable relative poses between images captured along different trajectories. To establish it as a new long-term localization benchmark, we jointly optimize the trajectories to provide accurate relative poses across a broad spectrum of visual domains.

III. OPTIMIZING THE 4SEASONS DATASET TO BENCHMARK CROSS-DOMAIN VISUAL LOCALIZATION

A. Original 4Seasons Dataset Details

Building on existing datasets [17], [18], [19], Sattler et al. [12] established the benchmark of long-term visual localization. Compared to the existing datasets in this benchmark, the autonomous driving 4Seasons dataset [15] provides data with accurate ground truth poses along multiple trajectories under a more varied set of weather, seasonal, and day-time conditions across a larger number of distinct environments. For our cross-domain localization benchmark, the first criterion for the dataset was having a wide range of categorical variations in visual conditions, which the 4Seasons dataset satisfied. However, for a comprehensive analysis, the dataset must fulfill an additional criterion: ground truth relative poses must be accurate across individual trajectories, i.e. across visual domains. Together, a dataset satisfying these two criteria would provide

TABLE I
DATASET OVERVIEW. TRAJECTORY LABELS ARE USED IN THIS WORK TO REFER TO THE CORRESPONDING TRAJECTORIES. THE DOMAIN IS DESCRIBED FOR EACH TRAJECTORY. FOR EACH SCENE, THE AVERAGE NUMBER OF FRAMES WITH RTK-GNSS MEASUREMENTS PER TRAJECTORY IS PROVIDED, AS WELL AS THE AVERAGE OF THE TOTAL NUMBER OF AVAILABLE FRAMES.

Trajectory Labels	Season	Weather Condition	Time of Day
Business Campus (3255 RTK-GNSS poses, 11765 available frames)			
BC1	Fall	Sunny	Morning
BC2	Winter	Cloudy/snowy	Afternoon
BC3	Winter	Sunny	Afternoon
Office Loop (2775 RTK-GNSS Poses, 14438 available frames)			
OL1	Spring	Sunny	Afternoon
OL2	Spring	Sunny	Afternoon
OL3	Spring	Sunny	Morning
OL4	Summer	Sunny	Morning
OL5	Winter	Cloudy/snowy	Afternoon
OL6	Winter	Sunny	Afternoon
Neighborhood (2790 RTK-GNSS poses, 10297 available frames)			
N1	Spring	Cloudy	Afternoon
N2	Fall	Cloudy	Afternoon
N3	Fall	Rainy	Afternoon
N4	Winter	Cloudy	Morning
N5	Winter	Sunny	Afternoon
N6	Spring	Cloudy	Evening
N7	Spring	Cloudy	Evening

the flexibility of evaluating the localization performances of networks across any two visual conditions available in a scene.

In the 4Seasons dataset, Real-time Kinematic and Global Navigation Satellite System (RTK-GNSS) measurements were utilized for a subset of frames to enhance the accuracy of the ground truth poses. For further details, please refer to the 4Seasons paper [15]. Additionally, these measurements provide geolocations for each frame, enabling the representation of individual trajectories within a unified coordinate system. However, the pose optimization within the dataset was conducted solely along individual trajectories, and as a result, the available maps of the 4Seasons dataset do not satisfy our second criterion. The existing cross-trajectory poses in the dataset are not reliable for benchmarking localization performance. Moreover, the unreliable relative poses result in misaligned local 3D representations of the maps, making them unsuitable for our data-centric supervision approach. This underscores the necessity for improved maps.

B. Refining the 4Seasons Dataset

We follow the HLoc [46] pipeline, which integrates and enhances COLMAP [5], to build SfM maps that satisfy our criteria for a subset of the scenes available in the 4Seasons dataset. In Tab. I, we present an overview of the data from these scenes, including trajectory labels referred to later in this paper. To accelerate the map optimization process, we use the available poses in the 4Seasons dataset as initialization. Our refinement pipeline is summarized in Fig. 2.

Map initialization. We initialize our maps using the provided poses of all frames of a 4Seasons scene, across multiple trajectories in a unified coordinate system. All frames with registered geolocations, i.e. with RTK-GNSS positions, are

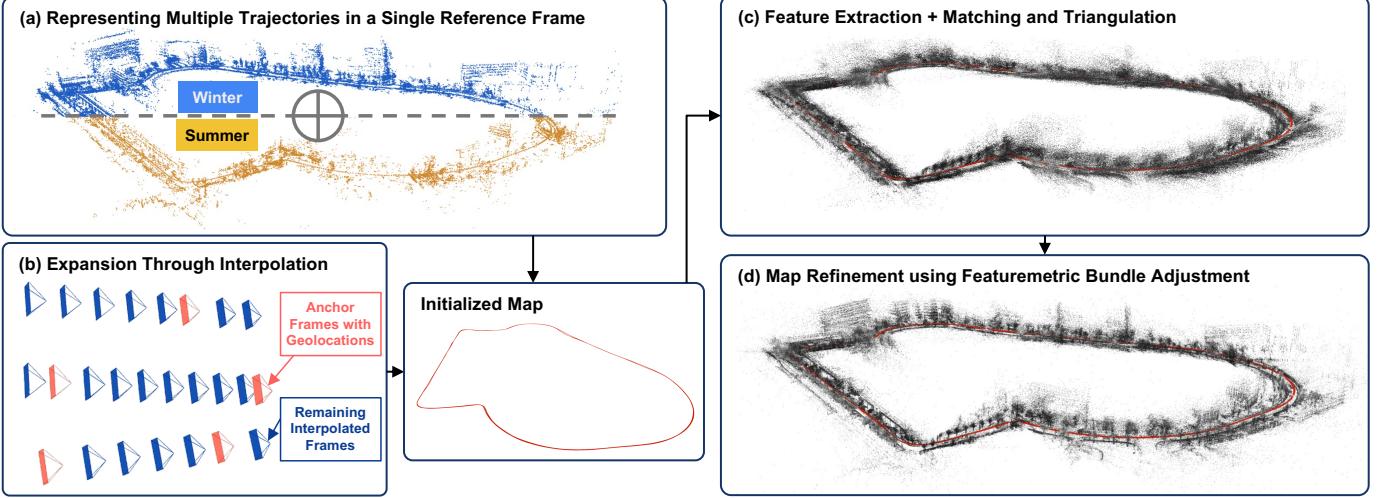


Fig. 2. **Joint Trajectory Map Refinement.** (a) Frames with registered geolocations along multiple trajectories of the 4Seasons dataset are placed, as anchors, into a global map. (b) The remaining frames have poses available only in a local reference frame optimized through stereo visual-inertial odometry. They are integrated into the map by interpolating them around the anchor frames. (c) SuperPoint [7] features are extracted and matched using SuperGlue [47]. Using the poses in the initialized map, the matched keypoints are triangulated into 3D. (d) Pixel-Perfect-SfM [48] is leveraged to optimize the 3D points and poses in the map through featuremetric bundle adjusment.



Fig. 3. **Keypoints and Matches in 4Seasons Dataset and Our Refined Maps.** The figure showcases keypoints from the 4Seasons dataset and from our refined maps. Matches across frames are highlighted in green. For the 4Seasons dataset, keypoints are matched using corresponding depth values and frame poses, while matches from our refined maps are directly extracted from the maps.

placed in the unified coordinate system and serve as anchors for map initialization. For the remaining frames, poses are available only in a local coordinate system, optimized through stereo visual-inertial odometry. Despite suffering from drift, we use the locally accurate relative poses to initialize the global map by placing them around the anchor frames.

Image matching. In the first step of the HLoc pipeline, covisible database images are identified. We generate these covisible image pairs using the ground truth relative poses available in our initialized maps. Per frame, we find the nearest n frames inter- and intra-trajectory.

SfM map triangulation. In the next stage, COLMAP triangulates shared keypoints across images into 3D. Although the provided 4Seasons poses can be used to initialize the maps, the provided keypoints can not be relied on for global map optimization. Specifically, there is only a limited number of available keypoints and the dataset does not provide keypoint matches across image pairs. Per-keypoint depth values, when

combined with relative poses, could theoretically facilitate geometric keypoint matching. However, the unreliability of relative poses across trajectories makes these geometric keypoint matches unreliable as well. Instead, we detect keypoints using SuperPoint [7], match them across identified image pairs using SuperGlue [47], and use them for triangulation. Fig. 3 compares the sparse set of cross-domain 4Seasons keypoints and geometrical matches with the abundance of SuperPoint + SuperGlue matches in our maps. SuperPoint + SuperGlue generates accurate matches across domains [17], and in a SfM pipeline, generates highly accurate maps.

Global map optimization. In the final stage of the HLoc pipeline, maps are globally optimized through a process of bundle adjustment. Conventionally, COLMAP would be used to refine the poses and 3D points in the map by globally minimizing reprojection errors. Taking advantage of the latest advances in SfM, we utilize Pixel-Perfect-SfM (PixSfM) [48]. In contrast to COLMAP, the global refinement is coined as

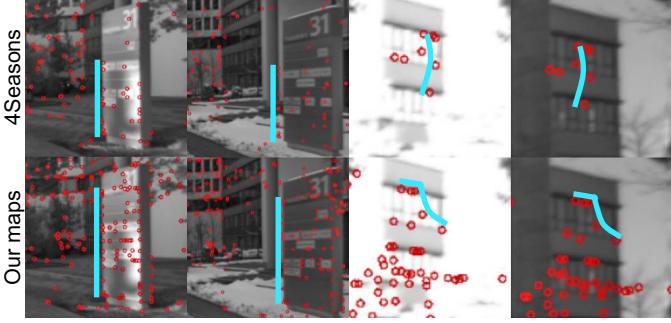


Fig. 4. Keypoint Reprojection for Pose Quality Analysis. Each row presents image pairs captured during different seasons. Keypoints from each left-side image are reprojected onto the right-side image using ground truth poses and keypoint depths. Top and bottom rows are reprojected keypoints from the 4Seasons maps and our maps respectively. Lines have been drawn to highlight reference and corresponding reprojected keypoints. The projected keypoints along the highlighted lines reveal consistent reprojection errors across the 4Seasons frame pairs. These consistent errors indicate inaccurate relative poses. The same can not be observed in the case of our refined maps.

“featuremetric”, because it optimizes the consistency of deep features across frames, and it additionally optimizes keypoints. Instead of relying on reprojection errors, 3D points, keypoints, and poses are optimized by minimizing a featuremetric cost.

C. Refined Map Quality Analysis

Due to the absence of “true” ground truth poses, we can not quantify the true accuracy of our maps, however, we demonstrate the improved quality of our maps over the ground truth poses available in the 4Seasons dataset. We provide both a qualitative and a quantitative analysis. Although we generate relative poses across all frame pairs in a scene, for a comprehensive analysis, we limit this study to the frames with registered geolocation positions; in the 4Seasons dataset, the other frames exist in their own local coordinate system.

Qualitative map quality analysis. We reproject keypoints from a query image onto a reference image using the estimated depths and the relative poses from the corresponding 3D map. The quality of the maps can be visually investigated by studying the reprojection of keypoints of a static object, such as a building. Errors in the map reveal themselves when the reprojected points on the query image do not match the same points in the reference image. These errors can be rooted in errors in both depth and pose, however, in contrast to depth errors, pose errors yield consistent reprojection errors across all keypoints in the image pair.

In Fig. 4, we showcase cross-domain pairs of reference and query images that demonstrate the reprojection of keypoints of static objects and reveal relative pose errors in the 4Seasons dataset. These can be clearly seen along edges of the static objects (we highlight some with cyan lines). In contrast to reprojecting using keypoints and poses from the original 4Seasons dataset, reprojection using the refined poses yields more accurate correspondences.

Quantitative map quality analysis. In the next section, we additionally benchmark the cross-domain localization performances of state-of-the-art feature extraction networks using our refined poses as ground truth, and properly introduce the

metrics. Here, we present a summary of the results in Tab. II, in addition to the same localization results, evaluated against the 4Seasons poses treated as ground truth. We report the median localization errors and, following [47], [49], Area Under the Curve (AUC) of pose errors at error thresholds: 5° , 10° , and 20° , averaged across all scenes and trajectory pairs.

All feature extractors achieve higher performances, i.e. lower median errors and higher AUC under all thresholds, when benchmarked against our refined poses. Due to their inherent imperfections, extractor performances are not a perfect indicator of improved map accuracies, however, the consistent disparity in localization errors emphasizes the enhancements achieved during our refinement. While the initial goal was to improve cross-domain poses, our experiment also reveals enhanced ground-truth poses along individual trajectories.

The qualitative and quantitative analysis combined highlights that the refined poses can be dependably used for evaluating cross-domain localization accuracies. This new dataset is used from here on for the remainder of the paper. With the capability to evaluate localization performances across all visual domains available in each scene, our presented benchmark offers an opportunity for the most comprehensive study of cross-domain localization to date.

IV. BENCHMARKING LOCALIZATION ROBUSTNESS ACROSS VISUAL DOMAINS

Utilizing our refined maps, we benchmark state-of-the-art feature extraction methods in the task of long-term localization. We evaluate localization performance using both relative and absolute pose estimation.

A. Benchmarking Relative Pose Estimation

To benchmark feature extractors, we propose to match keypoints between image pairs, estimate the relative pose between the two frames, and evaluate the quality of the predicted pose. The estimated translation can only be determined up to a scale factor and is therefore reported as a normalized direction vector. Rotation and translation pose errors are the relative angles between estimated and ground truth rotation matrices and translation vectors. Following previous works [47], [49], we report the pose errors, in degrees, as the maximum of translation and rotation angular errors.

Keyframing. For each scene and each trajectory pair, we select a set of reference frames and corresponding query frames. We select frames at increments of 10 meters along the reference trajectory. For each selected reference frame, we select corresponding query frames from the set of frames that are within 8 meters and within 45° of the reference frame. We select these query frames at increments of 2 meters.

Relative pose errors. In some cases, the sampled keyframe pairs have large relative poses. This often results in significant occlusion, particularly across different trajectories, rendering accurate relative localization impossible. To account for such outliers, we calculate median pose errors per trajectory pair. In Tab. III, we report metrics for both intra-domain and cross-domain cases. In intra-domain localization, relative poses

TABLE II
RELATIVE POSE ERRORS EVALUATED AGAINST 4SEASONS POSES VS. OUR REFINED POSES. POSE ERRORS ARE PRESENTED BOTH AS MEDIAN ERRORS AND AUC ACCURACIES. CONSISTENT HIGHER PERFORMANCES WHEN EVALUATED AGAINST OUR REFINED POSES INDICATES HIGHER GROUND TRUTH POSE QUALITY.

		Same Domain					Cross Domain				
		SuperPoint [7]	R2D2 [8]	D2-Net [9]	KP2D [16]	SiLK [11]	SuperPoint [7]	R2D2 [8]	D2-Net [9]	KP2D [16]	SiLK [11]
Median Error ($^{\circ}$) \downarrow	4seasons Refined	1.05	0.90	1.57	1.00	0.97	9.94	13.07	10.13	12.44	17.98
AUC (%) \uparrow	4seasons Refined	71.6/84.4/91.5 73.7/85.7/92.3	76.2/87.4/93.4 78.5/88.7/94.1	60.7/77.7/87.5 62.4/78.7/88.1	73.9/85.7/91.9 76.0/86.9/92.7	74.5/86.1/92.3 76.3/87.0/92.7	27.4/41.8/54.5 29.1/43.0/55.2	28.4/41.8/52.9 30.6/43.5/54.1	21.8/37.4/52.1 22.8/38.1/52.5	26.5/40.1/51.9 28.2/41.3/52.6	23.1/35.6/47.2 24.5/36.4/47.6

TABLE III
MEDIAN RELATIVE POSE ERRORS ON THE 4SEASONS DATASET [15]. FOR EACH SCENE, AGGREGATED MEDIAN ERRORS ARE PRESENTED, EVALUATED AGAINST OUR GROUND TRUTH POSES. IN THE CASE OF CROSS-DOMAIN, THE ERROR UNDER REFERENCE TRAJECTORY IS THE MEAN OF ALL MEDIAN POSE ERRORS BETWEEN THE REFERENCE AND ALL QUERY TRAJECTORIES.

Scenes	Business Campus			Office Loop						Neighborhood							ALL	
	Traj.	BC1	BC2	BC3	OL1	OL2	OL3	OL4	OL5	OL6	N1	N2	N3	N4	N5	N6	N7	
Intra-domain localization																		
SuperPoint [7]		1.28	1.07	1.26	1.06	1.02	0.95	0.91	0.90	0.94	0.85	0.78	0.89	0.90	0.79	0.85	0.88	0.96
R2D2 [8]		1.03	0.93	0.99	0.84	0.81	0.82	0.80	0.76	0.76	0.72	0.71	0.78	0.78	0.71	0.75	0.76	0.81
D2-Net [9]		2.13	1.56	1.70	1.65	1.65	1.55	1.39	1.38	1.45	1.35	1.28	1.41	1.52	1.21	1.26	1.34	1.49
KP2D [16]		1.08	0.91	1.05	0.99	0.96	0.98	0.92	0.84	0.93	0.85	0.79	0.87	0.90	0.83	0.83	0.88	0.91
SiLK [11]		1.11	0.88	0.99	0.96	0.96	0.93	0.83	0.83	0.86	0.82	0.76	0.86	0.90	0.77	0.83	0.83	0.88
Cross-domain localization																		
SuperPoint [7]		13.94	9.51	9.70	14.02	15.08	10.37	35.37	14.86	13.65	2.68	3.02	3.02	2.48	4.13	2.22	2.38	9.78
R2D2 [8]		20.02	12.44	13.29	17.95	19.33	11.40	50.52	18.06	18.79	2.63	3.19	3.56	2.27	4.64	2.07	2.22	12.65
D2-Net [9]		14.99	10.54	10.23	13.66	13.63	8.86	32.66	13.02	12.26	3.99	4.03	4.32	4.13	4.93	3.35	3.60	9.89
KP2D [16]		19.90	13.34	14.37	16.58	16.11	11.36	46.94	16.70	16.52	3.05	3.94	3.62	2.57	5.63	2.40	2.65	12.23
SiLK [11]		19.76	12.03	14.09	33.15	33.36	19.46	60.70	24.80	26.54	3.68	5.08	5.39	3.15	8.78	2.88	3.53	17.84

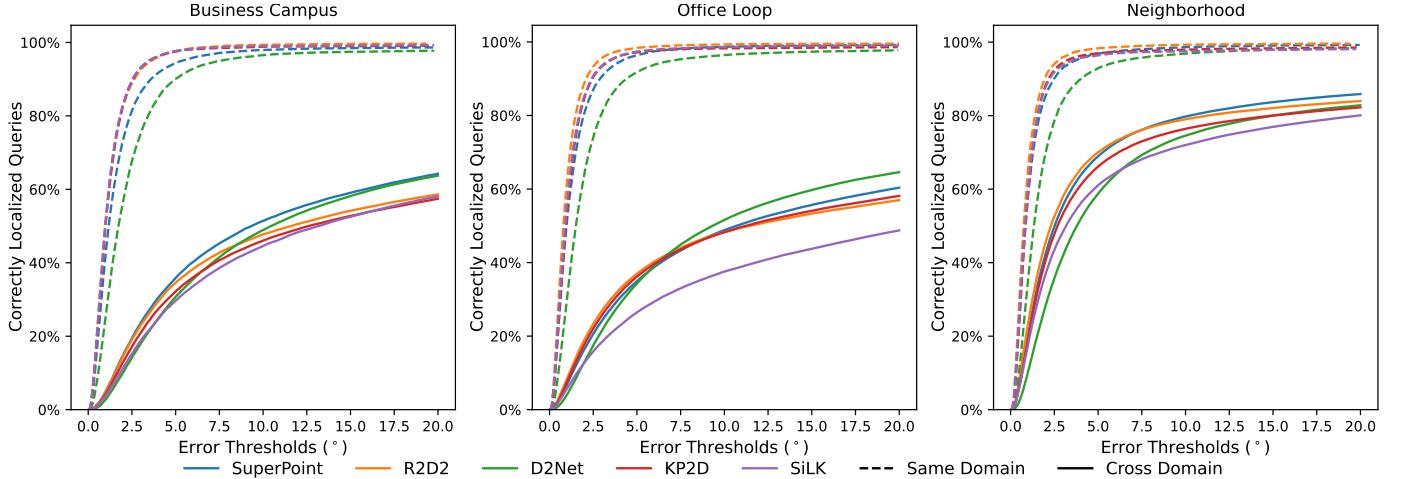


Fig. 5. **Cumulative Distribution of Relative Pose Errors on the 4Seasons Dataset [15].** For each scene, we present the AUC curves separately evaluated against our ground truth poses. Cross-domain plots are marked with filled lines while intra-domain plots are marked with dashed lines. The plots reveal a large performance gap between intra- and cross-domain localization.

are estimated across image pairs captured along the same trajectory, while in cross-domain localization, they are estimated across different trajectories. In the cross-domain case, the reported errors per reference trajectory are the mean of all median pose errors between the reference and all query trajectories. Furthermore, we plot in Fig. 5 the percentage of correctly localized images under different thresholds across all trajectory pairs. While we do not get insight into localization performances along individual trajectories, these plots provide a better understanding of the distribution of localization errors.

In Tab. III, all feature extraction networks exhibit significantly lower localization errors in intra-domain localization.

Furthermore, this performance gap is clearly illustrated in Fig. 5. In intra-domain localization, R2D2 [8] consistently outperforms other methods, while D2-Net [9] struggles. In contrast, D2-Net demonstrates robust results in cross-domain localization for scenes with larger variations, outperforming R2D2, KP2D [16] and SiLK [11]. Particularly SiLK suffers in cross-domain localization, consistently achieving the worst results across the board, and dramatically lagging behind in Office Loop. The results presented in the table suggest that trajectories BC1 and OL4, both characterized by sunny conditions, pose particularly challenging localization tasks within their respective scenes, likely due to the presence of

TABLE IV

CROSS-DOMAIN ABSOLUTE POSE ACCURACIES ON THE 4SEASONS DATASET [15]. FOR EACH SCENE THE PERCENTAGES OF ACCURATELY LOCALIZED QUERY IMAGES UNDER THRESHOLDS (0.25M, 2°), (0.5M, 5°), AND (5M, 10°) ARE REPORTED, EVALUATED AGAINST OUR GROUND TRUTH PSOES. RESULTS ARE PRESENTED FOR THE CASES WHEN GROUND TRUTH POSES AND NETVLAD [26] ARE USED FOR IMAGE RETRIEVAL.

Retriavl		Ground Truth			NetVLAD		
Scenes		Business Campus	Office Loop	Neighborhood	Business Campus	Office Loop	Neighborhood
SuperPoint [7]	66.6 / 82.6 / 99.7	47.2 / 60.7 / 89.9	71.5 / 82.0 / 97.6		64.5 / 80.3 / 97.3	44.4 / 56.1 / 78.6	70.5 / 80.6 / 95.3
R2D2 [8]	62.1 / 80.7 / 99.8	43.2 / 58.06 / 88.3	72.2 / 83.4 / 98.0		60.2 / 78.6 / 98.2	41.3 / 54.7 / 79.9	71.4 / 81.9 / 96.2
D2-Net [9]	61.8 / 83.1 / 99.9	42.9 / 59.1 / 92.3	65.8 / 80.1 / 98.5		59.7 / 80.9 / 98.6	40.6 / 55.3 / 83.1	64.8 / 80.0 / 96.5
SiLK [11]	44.9 / 61.2 / 85.4	36.0 / 46.6 / 68.0	65.5 / 77.7 / 93.7		42.2 / 57.4 / 80.2	33.0 / 42.3 / 59.7	64.3 / 76.0 / 90.7

shadows. These shadows present challenges in cross-trajectory localization by introducing edges in one domain that are absent in the other. The challenges in cross-domain localization also escalate under snowy conditions (as seen in BC3 and OL5), given that snow introduces markedly different textures and leafless trees present new geometries.

B. Benchmarking Absolute Pose Estimation

We further investigate the cross-domain localization performances of the extractors by calculating the absolute 6 DoF pose of query images with respect to a reference map, utilizing the HLoc localization approach [46]. Building on the poses available in our maps, the reference maps are reconstructed using local features extracted by the feature extractor being evaluated. For a given query image, image retrieval is performed to obtain the n most similar images from the database of the reference images. Rather than searching for matches throughout the 3D model, this process effectively narrows down the search space.

In Tab. IV, following [12], we report the percentage of localized images under three thresholds: (0.25m, 2°), (0.5m, 5°) and (5m, 10°). We report the results using two methods of image retrieval. The first method utilizes the ground-truth poses of the images to retrieve the closest reference images to the query. The second method relies on NetVLAD [26]. We compute the localization accuracy per reference-query trajectory pair. For each scene, we present cross-domain localization performances, averaging accuracies over all trajectory pairs.

R2D2 and SuperPoint achieve the highest correctly localized images under the finer thresholds (0.25m/2°, 0.5m/5°) while D2Net reports the best localization result under the coarse threshold (5m/10°). R2D2 [8] performs better than SuperPoint [7] in scenes with smaller domain variations. In contrast, SuperPoint [7] demonstrates greater robustness to larger variations in viewing conditions that are encountered in the Business Campus and Office Loop scenes.

The results that are obtained using NetVLAD for image retrieval mirror those obtained by using the ground truth poses except that the values are lower. This outcome is expected as using ground-truth pose for image retrieval, facilitates the retrieval of the query images closest to the reference image. Utilizing ground-truth poses in this manner helps in isolating the localization errors from image retrieval errors, with the former stemming from the quality of the reconstructed map, as well as the description and matching of local features. Similarly to the case of relative pose estimation, SiLK struggles

in cross-domain localization. KP2D was omitted from this experiment, as it was not implemented in HLoc.

In all experiments, a noticeable performance gap persists between intra-domain and cross-domain localization across all feature extractor methods, regardless of the domain type. While these extractors exhibit outstanding localization performance in intra-domain tasks, the persistent gap underlines that there is still significant room for improvement to achieve true long-term robustness in descriptors. Current state-of-the-art research has made remarkable strides in refining extractors using model-centric approaches such as improving models and optimization schemes. However, in this paper, we argue that these approaches are not sufficient for bridging this gap, as they do not incorporate real cross-domain variations across correspondences. Instead, we advocate for a shift towards data-centric approaches. In the forthcoming sections, we motivate the need for real cross-domain correspondences for supervising feature extractors to enforce cross-domain invariance. Furthermore, we present iCDC in detail and demonstrate a significant reduction in the performance gap when using its correspondences for supervision.

V. NETWORK SUPERVISION FOR LONG-TERM INVARIANT DESCRIPTORS

We can reduce the previously observed performance gap by supervising feature extractors using real cross-domain correspondences. However, manually annotating correspondences is infeasible, as it requires per-pixel annotations for each image pair. As a result, state-of-the-art methods [8], [7], [16], [11] utilize homographic transformations for flexibly generating correspondences between the reference and the transformed image. However, homographies cannot encapsulate real-world geometric variations, and image augmentations must be relied on to generalize across lighting variations.

A. Correspondence Generation Overview

Structure-based methods provide a reliable way to generate correspondences across different views [22], [50]. Relying solely on the sparse SfM maps, however, only results in a sparse set of correspondences. The initial sparse reconstruction using SfM can be further processed using Multi-view stereo to obtain a denser reconstruction. However, the resulting depth is often noisy and yields suboptimal correspondences.

Dense correspondence estimation is a field where networks estimate dense matches between image pairs, and could be used to generate dense cross-domain correspondences for

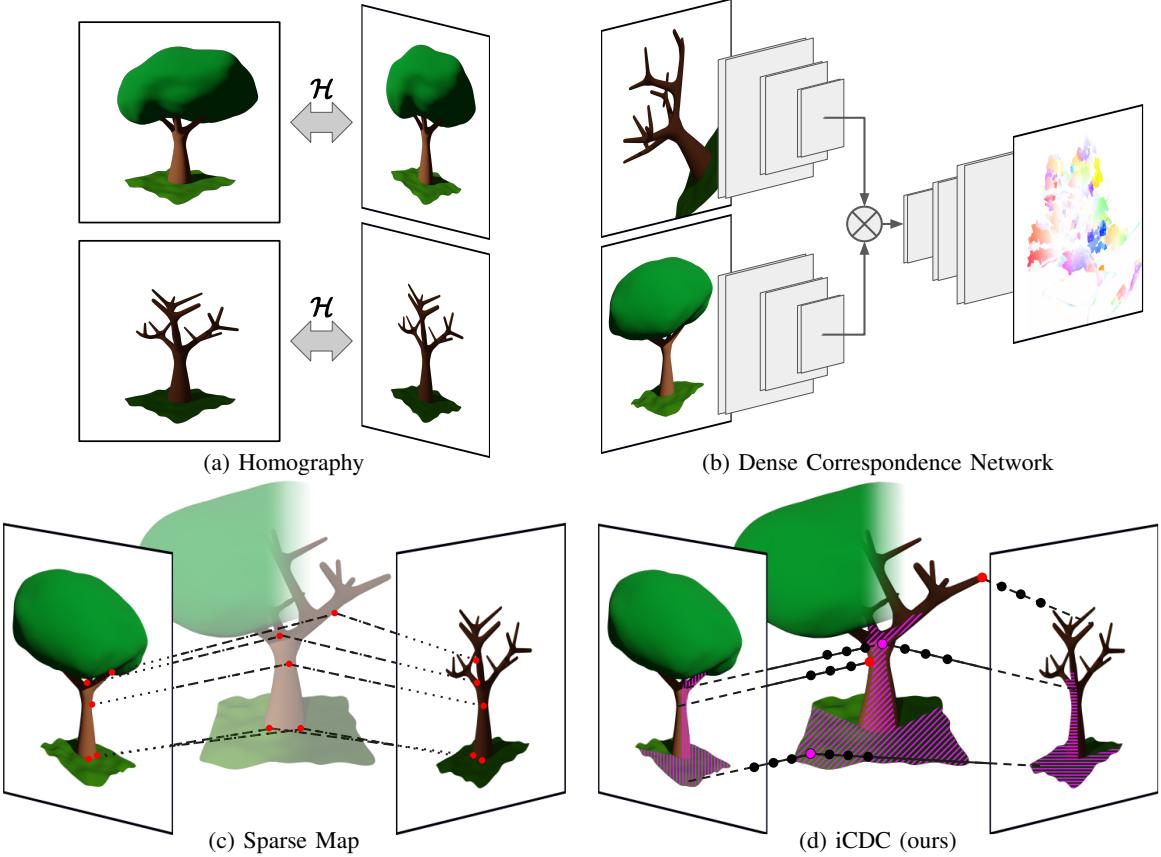


Fig. 6. Overview of Explored Correspondence Methods. Homographic transformations (a) are used to train most extractor methods out of convenience. Dense correspondence estimation networks (b) can estimate correspondences between real-world images across arbitrary viewpoint changes. However, these methods can be inaccurate, particularly across visual domains. Sparse SfM map matches (c) can serve as correspondences, providing highly accurate sparse correspondences across domains and views. In our method iCDC (d), we train domain-specific NeRFs using cross-trajectory SfM maps, and generate correspondences by comparing the implicit 3D geometries. Highlighted regions are where correspondences are extracted.

supervision [8]. However, estimating dense correspondences across long-term viewing conditions is particularly challenging due to significant variations in lighting, texture, and geometry.

In Fig. 6, we present an overview of all correspondence generation methods we explore. In addition to using homographies, we use a dense correspondence network, WarpC [43], to estimate dense correspondences across views, and we leveraged our refined maps to generate sparse correspondences. WarpC was trained using an unsupervised method through warp consistency, and the authors demonstrate state-of-the-art performance and robustness to varying views and domains across image pairs. Our refined maps provide highly accurate relative poses and per-keypoint depth values. Although direct utilization of the matches stored in the maps is possible for correspondence generation, we discovered that the resulting set was too sparse and lacked variation. By reprojecting the keypoints across arbitrary views, we obtained more varied correspondences and better supervision, albeit with more significant occlusion. Lastly, the figure includes iCDC, our method which is detailed in the following subsection.

B. iCDC: Implicit Cross-Domain Correspondences

Generating correspondences with NeRFs. A NeRF [51] learns an implicit continuous 3D representation of a scene from a collection of images with known camera viewpoints.

In the computer graphics community, NeRFs have been used to synthesize novel photorealistic views of a scene. The computer vision community has adopted NeRFs, exploiting the power of the implicit representations in different domains. [52], [53] synthesized data capable of training computer vision models, NeRFs show great promise in the field of scene understanding [54], and they have the potential to revolutionize SLAM algorithms [55], [56]. Beyond synthesizing RGB data, explicit 3D representations of the implicit scenes can be rendered. NeRF-supervision [57] generated correspondences across image pairs, by rendering corresponding depth images and comparing the underlying 3D geometries. The authors generated correspondences across images of an object, using a NeRF trained in a controlled environment with constant lighting conditions, by reprojecting the rendered depth maps from one frame to the other. The authors demonstrated the powerful cross-view information aggregation capabilities of NeRFs, by generating accurate correspondences across reflective objects – a canonical challenge.

iCDC overview. Building on NeRF-supervision [57], we introduce iCDC, capable of generating accurate correspondences across images captured in different visual domains. iCDC leverages the fact that the underlying static 3D geometry of a scene, such as buildings or street signs, remains unchanged, regardless of visual domain changes, such as daytime, weather,

and seasonal variations. For each scene, a single NeRF is trained per trajectory, i.e., per visual domain, using the poses from our multi-trajectory refined maps.

Training a single NeRF using data across different trajectories with different visual domains would result in inferior scene representation. NeRF-W [52] addresses this by learning per-image latent appearance codes. We found that this makes NeRFs robust to lighting variations, however, long-term variations are far more challenging to model. Consider the simple example, where a tree that is standing in one trajectory, and has been cut down in another. If a NeRF was trained on both trajectories, it would learn a representation of a scene that can not be correct for both cases and relying on the NeRF’s underlying 3D representation of the scene would result in false positive correspondences of the tree.

Although in iCDC the per-trajectory NeRFs do not share training data, they are expressed in the same coordinate system. Due to the high accuracy of the maps, the implicit 3D representations of the NeRFs share the same space and can be compared to generate real cross-domain correspondences. More specifically, for a pair of images captured along different trajectories, we first render depth maps using the corresponding NeRFs. Subsequently, we reproject the points utilizing the corresponding map poses.

Scaling NeRFs to represent 4Seasons scenes. The 4Seasons dataset is built from long trajectories that span large areas, and representing entire scenes with a single NeRF is challenging. Following recent works [58], [59], we evenly cluster the scenes into blocks based on frame poses and train a small NeRF per block. Since 4Seasons trajectories can revisit the same place multiple times, a NeRF block can contain multiple segments of the same trajectory. Near the edges of the blocks, this setup leads to poor rendering qualities, because there are only a few observations available to train the NeRFs. To overcome this, blocks are expanded by a buffer region. These buffer regions overlap with other blocks, and the frames in these buffer regions are used only for training purposes. Therefore, while the blocks share training data, each frame has a designated NeRF used to render depth maps. For convenience, we refer to the group of NeRFs that represent an individual trajectory as a trajectory-NeRF. At test-time, such a trajectory-NeRF functions exactly like a normal NeRF: frames are rendered at input camera pose locations.

Supervising NeRFs with sparse depth. DS-NeRF [60] showed that supervising a NeRF with sparse depth values can significantly improve the rendering performances with fewer training views. We found that in our scenario, to render accurate depth maps, sparse depth supervision is not only beneficial but necessary. This is because the 4Seasons data is captured using a forward-facing stereo camera, offering minimal view variation. Moreover, the scenes are unbounded, i.e. the scene points lie at large distance variations from the camera. Finally, the environments are uncontrolled, which means that the camera is often exposed to drastic lighting conditions changes along a trajectory. Each of these factors makes interpreting depth from only RGB images a significant challenge. The additional depth supervision provides sufficient

information for the NeRFs to effectively learn the underlying 3D representations. Following the DS-NeRF approach, we use sparse depth supervision to improve the depth rendering results. We extracted sparse depth values per frame directly from our sparse SfM maps.

Depth-frame rendering. NeRF-Supervision [57] proposed to treat the rays as depth probability distributions, sampling depth values during runtime. Instead, we follow the Instant-ngp [61] implementation. By modeling the transmittance along a ray, we can render depth by estimating the first instance after the light is absorbed by a surface in the scene. Furthermore, we can generate depth data prior to extractor training, instead of having to sample depths during the training.

Correspondence generation. For a pair of images $\mathcal{I} \in \mathcal{U}$ and $\mathcal{J} \in \mathcal{V}$, belonging to trajectories \mathcal{U} and \mathcal{V} respectively. We render depth maps using trajectory-NeRF- \mathcal{U} and trajectory-NeRF- \mathcal{V} . Then, we generate candidate correspondences across the images, by reprojecting the depth map of \mathcal{I} onto \mathcal{J} using the poses stored in the map. In the case where $\mathcal{U} = \mathcal{V}$, the same NeRF will be rendering the depths for both frames.

We use the loop consistency test, similar to WarpC [43], to filter out invalid correspondences. The candidate pixel of \mathcal{J} corresponding to a reference pixel of \mathcal{I} , has its own rendered depth value. We reproject this depth value back onto image \mathcal{I} , arriving at a certain 2D loop consistency distance from the reference pixel. We filter out correspondences with loop consistency distances that are higher than a user-defined loop consistency threshold α .

While the loop consistency test removes most of the invalid correspondences, there can be corner cases with false positives. For example, when the relative pose between the reference and query frame is only a transformation along the frame normal (the case between most frame pairs in the 4Seasons dataset), the corresponding center pixels will always pass the loop consistency test. To filter out false positives, we compare the reprojected depth value of a reference pixel, with the depth value of the query pixel, where the absolute difference between the depth values is the depth consistency difference. We filter out correspondences with depth consistency differences that are greater than the depth consistency threshold β ; another user-defined threshold.

We present in Fig. 7 examples of generated correspondences. In the first example, we filter using only the loop consistency test, while in the second example, we filter using both the loop and the depth consistency tests.

VI. EXPERIMENTAL SETUP DETAILS

A. Train-test-split

We train on two scenes of the 4Seasons dataset: Office Loop and Neighborhood and test on the third scene Business Campus. Both in training and testing splits, we employ spatiotemporal keyframing to minimize image pairs with minor variations. In the training split, along a single loop of a trajectory, consecutive reference frames are keyframed approximately 6 meters apart. For each reference frame, we sample up to two spaced-out query frames per trajectory loop, with maximum distances of 6 meters from the reference. If the trajectory loops

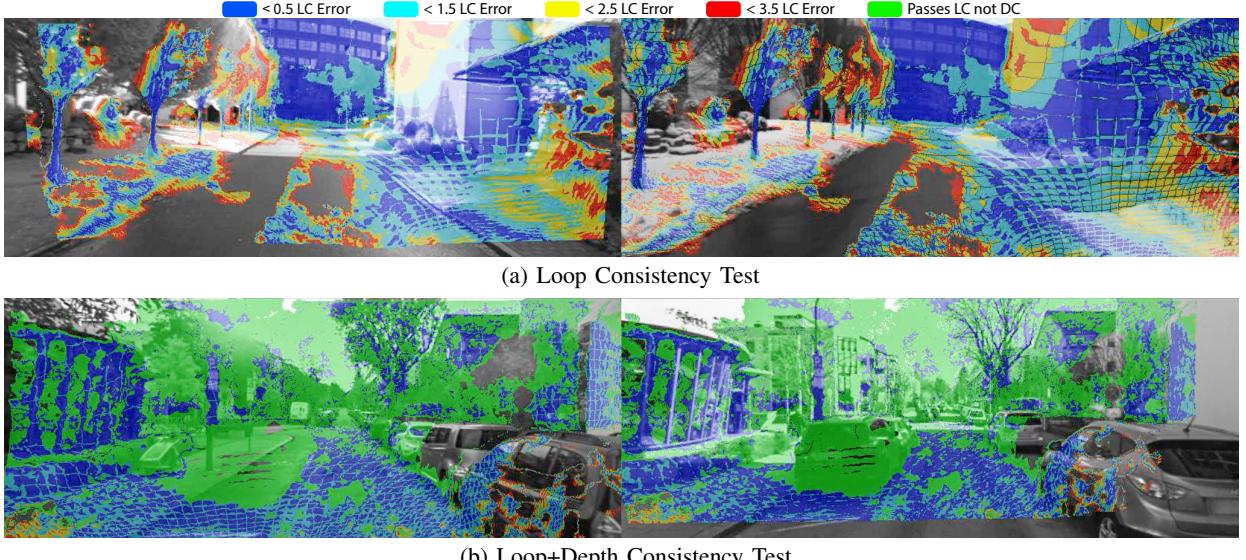


Fig. 7. **iCDC Correspondences.** Both figures display image pairs illustrating iCDC Correspondences. Fig. (a) shows pixels of correspondences satisfying loop consistency (LC) tests colored according to LC thresholds. Fig. (b) depicts the same mask, except marking pixels as green that pass the LC test but not the depth consistency (DC) test with a 15cm DC error threshold.

around, the number of reference keyframes and the number of query frames per reference are both multiplied by the number of loops. This methodology yields 16266 intra-domain and 77428 cross-domain training pairs. We explain the keyframing of the testing data for each experiment.

B. Constructed Maps

We outlined our approach in Sec. III for constructing joint-trajectory maps, to generate accurate relative poses across domains. While we have constructed global maps for scenes, to achieve finer refinements, we broke the problem down into multiple subproblems. As we only ever compare image pairs, we only need relative poses across two trajectories at a time. Therefore, for a given scene, we constructed maps for all trajectory pairs. Furthermore, for each trajectory pair, we construct two types of maps. For benchmarking local features, we construct maps consisting of a subset of the frames available in each trajectory (the frames with registered geolocations). On the other hand, for NeRF training, the more frames the better. Therefore, we generate a second set of maps using all available frames, as described in Sec. III.

C. iCDC Setup

NeRF training. While our multi-trajectory maps were built using only images captured by a single camera, the 4Seasons dataset provides stereo image pairs. To train the NeRFs, we use images captured from both cameras. For efficient training of the NeRFs, we use Instant-npg, as it is designed for fast processing, ensuring quick training and rendering crucial for our block-wise NeRF approach. The scene is divided such that the farthest apart frames in each block are approximately 64 meters apart. For more details about the engineering solutions necessary for representing unbounded outdoor scenes using Instant-npg, check out our supplementary material.

Generating correspondences. We generate correspondences across an image pair using the corresponding rendered depth maps, then filter out invalid correspondences through loop consistency and depth consistency tests, as detailed in section V. For the loop consistency test, we set a loop consistency threshold of $\alpha = 2$ (in pixel units). For the depth consistency test, anchoring poses using geolocations allowed us to use metric units to set our depth consistency threshold. We observed a meaningful improvement in performance with the implementation of the depth consistency test. We selected a threshold of $\beta = 15$ cm.

D. Extractor Training Setup

While our cross-domain correspondences can supervise any feature extractor, R2D2 [8] is the primary network of our investigation. Our choice of R2D2 was influenced by its unique training loss characteristics. Besides optimizing for keypoint repeatability, which is further divided into “peakiness” and “cosimilarity” losses, it also optimizes for descriptor reliability. Each of these losses can be explored, providing an avenue for thorough investigation. In fact, we found that the repeatability loss needed to be adapted for cross-domain correspondence supervision. The loss function was designed for homography-based correspondences, where there are no significant geometrical variations across the matched keypoints. For cross-domain correspondences, we found that the loss needed to be adapted. To train R2D2, for all cross-domain correspondence methods, we employed the cosimilarity loss supervision exclusively for intra-domain training pairs. For more details and insights about the losses, please see our supplementary material.

To demonstrate the importance of real cross-domain correspondences, we trained an additional extractor to demonstrate that the improvements were not network-specific. We train SiLK [11] – we selected it due to the simplicity of its

training pipeline. We trained both networks using their default pipelines, making only the requisite adjustments to accommodate our correspondences. In the case of R2D2, we updated the losses, and in the case of SiLK, the pipeline needed to be updated to support non-homography-based correspondences.

In both the R2D2 and SiLK training pipelines, images are augmented using noise, as well as random cropping and scaling. Additionally, when supervising with real correspondences, we apply homographic transformations as a form of data augmentation. Unlike the SiLK pipeline which only supports homography-based supervision, R2D2 has a function to map ground truth correspondences across an image pair post-application of a perspective homographic transformation to the query image. We adopt this strategy from R2D2 to adapt the SiLK pipeline, enabling it to utilize real correspondences.

To speed up training, we initialize both R2D2 and SiLK with their provided default weights. However, we use a large enough learning rate so that the previous patterns are mostly overwritten. Initializing using the default weights only marginally improves over initializing with random weights.

VII. RESULTS

A. Overview of Experiments

We train R2D2 using multiple correspondence methods on the 4Seasons dataset and observe a significant reduction in the performance gap when utilizing real cross-domain correspondences. To ensure that the performance improvements are not solely attributed to the training data, we establish two baselines for our networks. Firstly, we use the default weights as a baseline. Secondly, we create another baseline by training R2D2 using its default pipeline (homographies) on the 4Seasons dataset. In our first experiment, we train using two straightforward real cross-domain correspondence generation methods. We introduced iCDC to enhance the accuracy of the cross-domain correspondences. In a second experiment, we benchmark the iCDC-supervised extractor. For that, we generate correspondences using different methods and use them to train our feature extraction and description backbone.

We summarize the correspondence generation methods:

- **Homography:** The default R2D2 training scheme is utilized. Individual images are loaded, a perspective homography is applied, and the resulting image pairs are then individually augmented.
- **Dense Correspondence Network:** We use WarpC [43] to estimate dense correspondences between an image pair. Image pairs are loaded, and WarpC estimates are used as ground truth. To filter out inaccuracies, a loop consistency test is used. Both images are individually augmented, one of which is also augmented using homographic transformation. This same augmentation strategy was also implemented for the following two methods.
- **Sparse SfM structure:** We extract sparse depth values from our maps, and reproject them across image pairs to generate correspondences.
- **iCDC:** Per training image pair, depth maps, rendered by corresponding trajectory-NeRFs, are used to generate correspondences. Candidate correspondences are filtered using loop and depth consistency tests.

B. Qualitative Correspondence Analysis

In Fig. 8, we compare the quality of correspondences between the dense correspondences estimator WarpC and iCDC. For a fair comparison, we filter out candidate correspondences (red and purple regions) using only the loop consistency test for both methods. We adjusted the thresholds for both methods to generate a similar number of correspondences per image pair. For an image pair (top row), we generate correspondences using both methods. Using the correspondences, we warp the right-hand side image onto the left-hand side image. Effectively, we use the correspondences to align the two images. The warped images are in the second (iCDC) and third (WarpC) rows, with windowed regions zoomed in.

For both methods, correspondence errors are small-scale and are often visually hard to identify, however, WarpC had significantly more frequent failure cases. In the zoomed-in regions, the warped dense correspondence estimator frames are significantly misaligned.

C. Relative Pose Estimation on Our Benchmark

For relative pose estimation, we apply our benchmark setup described in section IV, evaluating only on Business Campus trajectories. We choose reference frames at 10m increments along the trajectories. Query frames are selected at two-meter increments, a maximum of eight meters from the reference frame. Using extractors, we estimate relative poses between image pairs, evaluating performances using the ground truth poses in our maps. For a comprehensive understanding of error distributions in cross-domain localization, we plot the cumulative error curves in Fig. 9.

Discussing primary extractor results (R2D2). In Fig. 9 (a), we set SuperPoint with default weights as the lower bound – the best-performing pre-trained extractor on cross-domain localization. As an upper bound, we set the intra-domain localization performance of R2D2 with default weights.

Training R2D2 using their default pipeline (homography) does improve the cross-domain localization performance on our benchmark, however, the improvement is only marginal. It is still outperformed by SuperPoint with default weights, indicating that the meaningful improvements are not attributed to the 4Seasons training images. Employing supervision using cross-domain correspondences uniformly achieves at least approximately a 50% reduction of the median error on cross-domain localization. A significant performance gap is observed between the homography-based method and our worst-performing extractor supervised by cross-domain correspondences. This notable performance increase highlights the need for data-centric approaches. Even naive methods, like supervising using sparse map-based correspondences or employing estimates from dense correspondence networks, can already meaningfully enhance robustness against long-term variations.

Despite the high accuracy of the correspondences extracted from the sparse maps, using the estimates from the dense correspondence network yields better results. Although the accuracy of correspondences is crucial, the high variation of positive anchors during training, provided by the dense correspondences, was also an important factor. Through our qual-

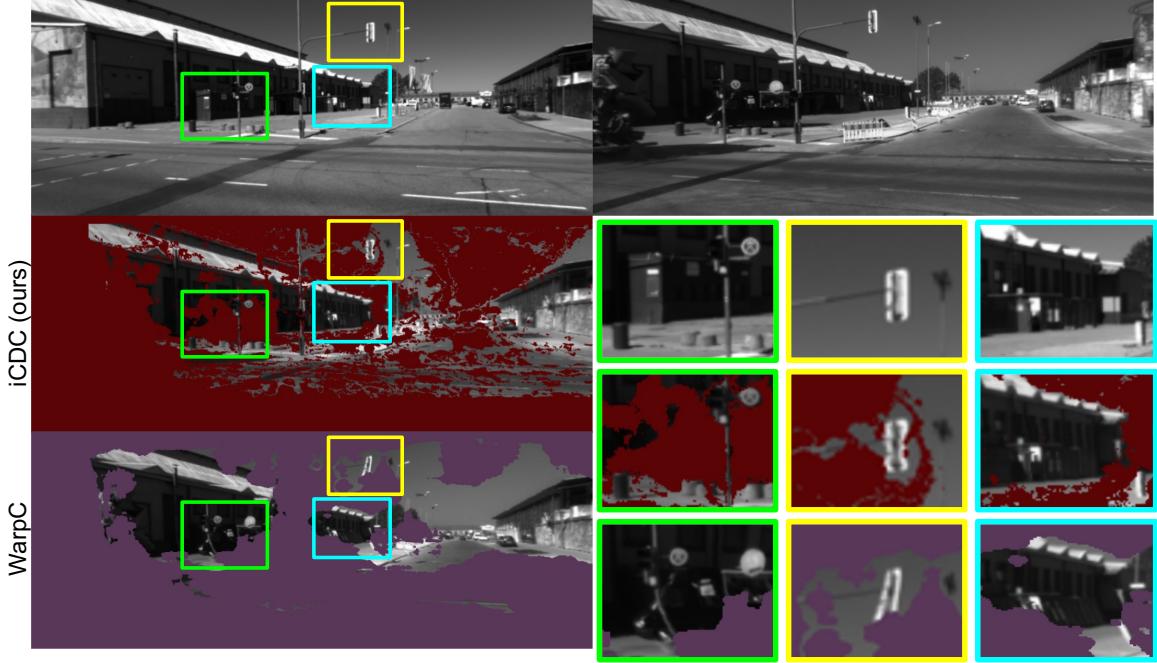


Fig. 8. **Comparison Between Our NeRF Correspondences and WarpC [43] Correspondences.** Correspondences are generated for an image pair (top row). The left column’s second and third rows show WarpC and iCDC correspondences respectively. More specifically, the right-hand side image is warped onto the left-hand side image following the correspondences. We zoom in on windows for detailed examination. In the case of accurate correspondences, the warped frames align with the left-hand side frame.

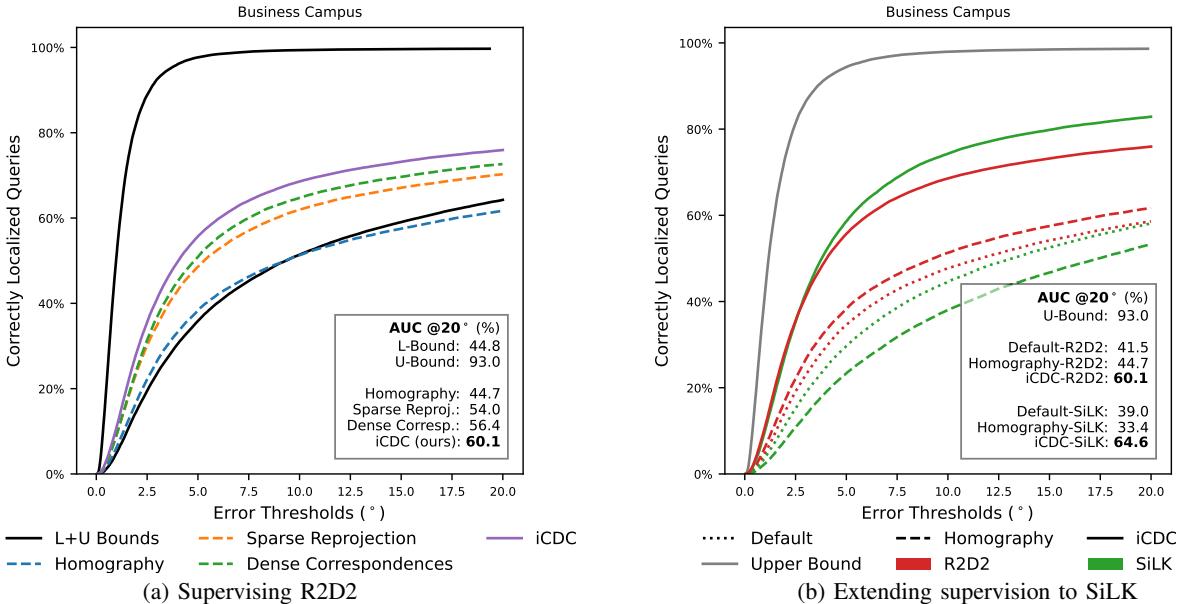


Fig. 9. **Cumulative Distribution of Cross-domain Pose Errors on the 4Seasons Dataset [15].** Our trained R2D2 extractors are benchmarked using our ground truth poses (a). Lower and upper bounds (L+U Bounds) are the error distributions of the best cross-domain localization method (SuperPoint [7]) and R2D2 [8] evaluated on intra-domain respectively. Dashed lines are our trained extractors used to benchmark iCDC. Furthermore, we compare our trained SiLK performances against R2D2 performances (b).

tative analysis, we identified that generating correspondences using our iCDC network results in higher accuracy dense correspondences compared to the dense correspondence estimator WarpC. We achieved the best results when supervising R2D2 using iCDC correspondences, due to the increased density and improved accuracy of the correspondences. Referring to the AUC of the pose errors at a threshold of 20° , we improved the cross-domain localization performance of R2D2 by 44.8%

and reduced the performance gap between the default R2D2 and R2D2 trained using our iCDC method by 36.1%.

Extending discussion to SiLK. In order to validate that the findings of this work are not confined to R2D2, we extended our training to SiLK using a smaller subset of the correspondence methods. For clarity, we have separated the results of SiLK supervision in a separate figure (Fig. (b)). SiLK exhibits some distinct characteristics. As noted in the

benchmarking section (Sec. IV), SiLK achieves state-of-the-art performances within the same domain localization, however, it achieves the worst performance in cross-domain localization. Our experiments indicate that SiLK fits training data more aggressively than R2D2, and if the training data is not varied enough, or is not visually similar enough to the test domain, then SiLK does not generalize well.

We trained on the 4Seasons data utilizing the default SiLK training pipeline (homography). In contrast to the observation with R2D2, the resulting network underperforms compared to SiLK with default weights. Overfitting to the training scenes, with the 4Seasons dataset being less varied than SiLK’s default training data, yields the worst cross-domain performance in this work. Interestingly, however, supervising SiLK with iCDC correspondences leads to the best performance across all our networks. The aggressive fitting of cross-domain correspondences proves highly effective on our test scene, despite the limited training image variation.

D. Absolute Pose Estimation on Our Benchmark

We localize a query image using feature extractors with respect to a reference 3D map. Our approach follows the benchmark setup for absolute pose estimation as previously detailed in sec. IV. For each trajectory in the testing scene, we triangulate an SfM map using an extractor and ground truth poses. For image retrieval, we utilize NetVLAD [26]. The localization results under thresholds (0.25m, 2°), (0.5m, 5°), and (5m, 10°) are reported in Tab. V. Metrics are averaged per reference trajectory and are also globally averaged.

TABLE V

CROSS-DOMAIN ABSOLUTE POSE ACCURACIES ON THE 4SEASONS DATASET [15]. FOR OUR TEST SCENE, THE PERCENTAGES OF ACCURATELY LOCALIZED QUERY IMAGES UNDER THRESHOLDS (0.25M, 2°), (0.5M, 5°), AND (5M, 10°) ARE REPORTED, EVALUATED AGAINST OUR GROUND TRUTH POSES. RESULTS USING DEFAULT WEIGHTS, OUR TRAINED R2D2 NETWORKS, AND OUR TRAINED SiLK NETWORKS ARE GROUPED SEPARATELY.

Traj.	BC1	BC2	BC3	ALL
Pretrained Benchmarks				
SuperPoint [7]	61.4/78.1/96.3	67.1/79.7/98.4	65.1/81.9/97.6	64.5/79.9/97.4
R2D2 [8]	55.0/75.2/97.6	63.6/79.3/99.1	61.9/81.2/98.0	60.2/78.5/98.2
D2-Net [9]	57.4/79.6/98.0	60.7/80.5/99.2	60.9/82.4/98.6	59.7/80.9/98.6
SiLK [11]	34.0/46.9/70.1	50.2/64.9/98.3	42.5/60.5/82.2	42.2/57.4/80.2
Supervising R2D2 [8]				
Homography	61.6/77.5/97.9	67.3/80.1/99.3	64.4/82.6/98.2	64.4/80.1/98.5
Sparse Map	64.2/79.2/ 98.4	70.7 /81.1/99.5	67.5/84.0/98.5	67.5/81.4/98.8
Dense Corresp.	64.4/ 80.8 /98.3	67.4/ 82.0 /99.5	66.8/ 85.4 /98.7	66.2/82.7/98.8
iCDC	66.3 /80.6/ 98.4	69.1/81.8/ 99.6	69.0 /85.3/ 98.8	68.2 / 82.6 / 98.9
Supervising SiLK [11]				
Homography	25.7/38.9/64.1	42.3/58.4/83.1	41.7/59.0/81.1	36.6/52.1/76.1
iCDC	63.0/77.2/97.1	69.2/80.1/98.5	67.4/83.7/98.0	66.5/80.3/97.8

We observe similar patterns as in the relative pose estimation case, both for R2D2 and SiLK. Supervising R2D2 using the default pipeline increases the cross-domain localization performance, however, does not outperform SuperPoint with default weights. Supervising R2D2 using cross-domain correspondences uniformly outperforms the default extractors, and moreover, the accuracy of our iCDC correspondences yields the best results. However, the reference maps contain a very dense set of frames. Consecutive frames are no more than 30

cm apart, and the Business Campus trajectories loop around three times. Therefore, the task of NetVLAD is made easy, and retrieved reference and query frames have very small relative poses, simplifying the localization task. We find that there is less room for improvement than in the relative pose estimation case, with the performance increases being less dramatic. In relative pose estimation, on the other hand, we engineer a broader range of relative poses between test image pairs.

E. Evaluation of Our Correspondences on the Visual Localization Benchmark

To investigate the generalizability of our feature extractors supervised by iCDC, we evaluate their localization performances in the long-term localization benchmark [12]. The benchmark datasets are captured using different camera setups, with each dataset presenting its own set of challenges.

Overview of the benchmark datasets. CMU-Seasons [19] is the setup most similar to our own: a camera mounted on a car with scenes ranging across rural and urban and car trajectories ranging across visual domains. However, while the 4Seasons cameras are forward-facing, the CMU cameras are rotated 45°. Furthermore, a significantly larger portion of the CMU dataset is looking at vegetation than in the 4Seasons dataset.

While the RobotCar-Seasons dataset [18] has images captured by sideways facing cameras, it also has rear-view images which are more similar to the 4Seasons scenario. In fact, the query frames in the benchmark are all rear-view images. Furthermore, RobotCar is captured in an urban scene, with more recognizable geometries than CMU. However, in addition to visual domain variations similar to those of 4Seasons, one of the key benchmarks of RobotCar is night-time image localization against a day-time reference map. We do not have night-time images in our training set. Moreover, the image qualities are significantly lower than those of 4Seasons, in particular, the night-time images with strong motion blur.

Aachen Day-Night [17] also benchmarks night-time image localization and the quality of those images is much higher. However, Aachen Day-Night presents the greatest challenge for our extractors. It is a hand-held camera dataset that presents drastically different relative views between images, and while the 4Seasons images mostly look along roads with buildings along the sides, the Aachen Day-Night images mostly look directly at buildings. Localization results, using NetVLAD for image retrieval, on the long-term visual localization benchmark are listed in Tab. VI.

Discussing primary extractor results (R2D2). On the driving datasets, iCDC-R2D2 correspondences outperformed default SuperPoint and R2D2 in all metrics. In CMU Seasons, there are major performance increases in both the Suburban and Park scenes. In the Urban scenes, the performance increase is less drastic, however, given the already very high performances, the increase is meaningful. In RobotCar-Seasons day-time localization, our extractor only marginally outperforms the baselines. This might be a NetVLAD bottleneck, or due to the lower image qualities. In night-time localization, however, our extractor significantly outperforms the other baselines,

TABLE VI

BENCHMARKING LONG-TERM VISUAL LOCALIZATION. WE EVALUATE THE PERFORMANCE OF OUR EXTRACTORS ON THE LONG-TERM LOCALIZATION BENCHMARK [12], AND COMPARE THEM TO THE PERFORMANCES OF SUPERPOINT, R2D2, AND SiLK WITH DEFAULT WEIGHTS. CMU SEASONS AND ROBOTCAR-SEASONS ARE AUTONOMOUS DRIVING DATASETS, SIMILAR TO OUR TRAINING DATA. AACHEN DAY-NIGHT PRESENTS THE GREATEST CHALLENGE DUE TO THE LARGE DOMAIN VARIATION BETWEEN HAND-HELD AND CAR-MOUNTED CAMERA SETUPS.

Scenes	CMU Seasons Extended [19]			RobotCar-Seasons v2 [18]		Aachen Day-Night v1.1 [17]	
	Urban	Suburban	Park	Day All	Night All	Day	Night
Traj.							
SuperPoint [7]	93.5 / 96.1 / 98.2	84.2 / 86.9 / 92.9	78.2 / 81.9 / 85.8	64.2 / 93.7 / 99.1	14.0 / 27.7 / 33.8	87.1 / 93.3 / 96.7	68.1 / 80.6 / 90.1
R2D2 [8]	94.4 / 97.3 / 99.0	85.3 / 88.9 / 93.5	74.2 / 78.8 / 83.6	64.5 / 94.4 / 99.2	16.8 / 32.6 / 39.4	88.0 / 95.3 / 97.9	67.0 / 86.4 / 96.3
iCDC-R2D2	96.5 / 98.8 / 99.4	93.4 / 94.9 / 97.1	87.5 / 90.3 / 92.9	64.9 / 94.6 / 99.4	24.5 / 45.5 / 52.2	87.7 / 93.7 / 96.6	67.5 / 84.3 / 93.7
SiLK [11]	57.6 / 62.0 / 68.0	42.5 / 47.9 / 57.6	17.0 / 19.6 / 26.6	62.2 / 90.8 / 96.1	12.6 / 18.4 / 21.7	78.8 / 84.3 / 89.8	38.2 / 48.2 / 55.5
iCDC-SiLK	92.2 / 94.9 / 96.9	81.8 / 84.9 / 89.2	63.2 / 67.2 / 72.4	64.5 / 94.0 / 99.2	15.4 / 26.8 / 34.3	83.9 / 90.0 / 94.1	60.7 / 76.4 / 85.3

suggesting that supervising domain variation improves performances in unseen domains. These results underscore the importance of improved data-centric approaches. Despite the lack of view variation in the 4Seasons dataset, we see significant performance increases in long-term localization scenarios. The real-world variations across correspondences even improve localization performances in unseen visual domains, such as in night-time localization.

Despite the large view-point domain shift, our extractor achieves competitive results on the Aachen Day-Night benchmark, outperforming SuperPoint in almost all metrics. R2D2 with default weights achieved the best results in most metrics, however, it was trained on Aachen Day-Night day-time data, as well as on day-to-night style-transferred versions of the same images. For detailed CMU and RobotCar results, please see the supplementary material.

Extending the discussion to SiLK. Our benchmark revealed that SiLK aggressively fits the training data, struggles with generalizing, and with default weights achieves low cross-domain localization performances. Furthermore, training on 4Seasons with their default pipeline reduces the performance by overfitting. On the other hand, we found that supervising SiLK with iCDC correspondences yielded the best results on our benchmark. That being said, generalizing to the long-term localization benchmarks is a separate challenge. SiLK with default weights performs poorly on CMU and Aachen Day-Night, although it has comparable results to the other baselines on RobotCar. Similarly to the case of our benchmark, iCDC supervision significantly improves the performance of SiLK on the long-term localization benchmarks by aggressively fitting the domain variations. This is only enough to make certain metrics comparable in CMU and Aachen. As RobotCar-Seasons' rear-view images present the scenario most similar to that of 4Seasons, iCDC-SiLK achieves competitive results.

VIII. CONCLUSION

Modern feature extraction methods advance the state of the art by developing more powerful network architectures and novel optimization schemes. They supervise their networks by applying perspective homographies and augmenting data to generate training tuples, enabling the training schemes to run flexibly in a self-supervised manner. However, in long-term localization scenarios, visual and perspective variations arise that are beyond the simulation capabilities of the commonly used data augmentation practices. This creates a bottleneck

that model-centric approaches cannot overcome. To achieve successful life-long deployment, a shift towards improved data-centric approaches is imperative.

We established a novel benchmark for the long-term visual localization scenario, enabling flexible evaluation of visual localization performances across a broad spectrum of visual domains. The 4Seasons dataset offers trajectories of images captured in several kinds of scenes under varied weather conditions, seasons, and daytime lighting conditions. Through joint trajectory SfM map refinement, we generated highly accurate maps, providing ground truth relative poses between images captured under different visual domains. We thoroughly evaluated the performances of state-of-the-art extractors on our benchmark, revealing a large performance gap between intra- and cross-domain localization.

To achieve accurate cross-domain localization, real cross-domain correspondences are needed to supervise networks. To accentuate this point, we implement readily available methods to generate correspondences across domains. We train two feature extractors, R2D2 and SiLK, using these correspondences, and significantly reduce the performance gap. We propose a novel data-centric approach to further improve cross-domain localization performances. Implicit Cross-Domain Correspondences (iCDC) is a method to generate accurate correspondences across visual domains by comparing the underlying implicit 3D geometries of domain-specific NeRFs. By supervising networks with real cross-domain correspondences, we reduced the performance gap by 36.1%, as evidenced in our relative pose estimation benchmark.

This work represents a significant step forward in enhancing the robustness and reliability of visual localization pipelines for life-long deployment. It highlights the need for data-centric methods in improving cross-domain localization performance. Future works could explore other data-centric approaches and could refine the efficiency and scalability of the proposed method. The creation of easily deployable cross-domain correspondence generation pipelines, robust against dynamic objects within scenes, would enable scaling to larger amounts and more varied datasets. This advancement, in turn, holds potential for training networks capable of achieving intra-domain localization performances in long-term deployments. The insights gained from this research provide a solid foundation for subsequent studies focused on optimizing the performance of feature extraction networks.

APPENDIX

A. Training Learned Matchers Using iCDC

We additionally explored two related fields that are relevant to this work. (i) NeRFs can be used to generate data. Instead of storing a dataset and sampling from a discrete set of frames, NeRFs could be used to flexibly sample images along a continuous range of camera poses. Furthermore, views can be extrapolated from the original camera trajectory, introducing novel views that have the potential to improve the generalization of networks. Moreover, (ii) learned matchers [47], [62] rely on local features and learn matching patterns across image pairs. They have significantly increased localization performances in complex environments, such as day-night localization in the long-term localization benchmark [12]. We study these two extensions of iCDC.

Training with novel views. We explore the impact of using rendered images in training. In particular, we were interested in two factors: (1) can we maintain performances when training on rendered data, and (2) can we improve generalization to other datasets by using extrapolated views? To this end, we generate a secondary dataset rendered by our NeRFs. The images are rendered at the same XYZ coordinates as the training data, except we incrementally rotated the views around the gravity axis left and right. We found, however, that because of the forward-facing setup, extrapolation was very difficult. We limited the rotation to $\pm \frac{\pi}{8}$, and even then, there was a noticeable decrease in rendering quality. For training, we used iCDC correspondences.

After training, we evaluated on our benchmark. It is important to note that the extrapolated views will not improve performance on our benchmark, because such variations are not present. We measured the cross-domain performance; pose AUC in % under thresholds $5^\circ/10^\circ/20^\circ$ is: 28.1/43.1/55.7. The performance dips in comparison to our best network iCDC-R2D2 (31.5/47.4/60.1), but it did outperform all other feature extractors trained using homography (the best being 20.2/32.7/44.7), and, while being outperformed, achieved comparable results to the best performing naive cross-domain correspondence generation approach, the dense correspondence network supervision: 28.1/43.6/56.4. We present long-term localization benchmark results for many more extractors in Tab. A.1. We get more balanced results for the training with synthesized views (Nov.-iCDC-R2D2+NN). Thanks to the improved generalizability, it performs approximately as well as iCDC-R2D2, and outperforms it in RobotCar and Aachen Day-Night in many metrics, despite the suboptimal quality of the training data.

Learned matchers. LightGlue [62] was recently introduced, improving the trainability, efficiency, and performance of the deep keypoint matcher SuperGlue [47]. We train a vast array of networks using the default setup, with two key adaptations. (a) add new types of augmentation to data in the training pipeline, and (b) train on the 4Seasons dataset using iCDC. (a) is important because we found that, while feature extractors overfit to texture patterns in images, learned matchers overfit to repetitive matching patterns across image pairs. The default LightGlue training pipeline did not apply homographic

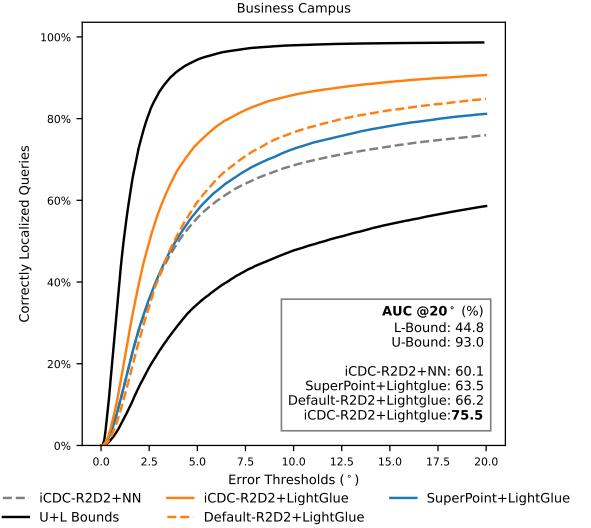


Fig. A.1. **Cumulative Distribution of Cross-domain Pose Errors on Our Benchmark Dataset.** Keypoints are matched using either Euclidean distance-based Nearest Neighbors (NN), or using LightGlue [62]. The lower bound is the SuperPoint [7] with default weights. The upper bound is the intra-domain localization performance of [8] with default weights.

transformations or cropping to the training pairs when training on Megadepth [63], as the types of images are much more varied than in our case. In the 4Seasons dataset, it is very easy to overfit to matching patterns. For example, keypoints on the left will stay on the left in almost all cases, because the car moves forward in a straight line. While overfitting to matching patterns on the 4Seasons dataset leads to the best performances on our benchmark, the resulting matchers can not generalize at all to other datasets. We introduced a small amount of random cropping to explore the effect of overfitting to matching patterns. We only present these results. Furthermore, in a different approach, we also use our rendered extrapolated views for training in an attempt to generalize to other datasets. We only present results for training LightGlue with novel views on the long-term localization benchmark.

In Fig. A.1, we present a small subset of the networks we trained, benchmarking against our extractors trained with iCDC using Euclidean distance-based nearest neighbors (NN) matching, and against pretrained SuperPoint+LightGlue. As expected, matching with LightGlue led to a huge increase in performance over SuperPoint+NN, the lower bound in the plot. While SuperPoint+LightGlue outperforms our network with NN matching, iCDC-R2D2+NN, the results are comparable, highlighting the importance of cross-domain correspondence supervision.

To investigate the importance of descriptor quality when matching with learned matchers, we first trained LightGlue with default-R2D2 features. It marginally outperforms SuperPoint+LightGlue on our benchmark, however, to draw any conclusions about the networks or the training strategy, a further ablation study would be needed. Subsequently, we trained LightGlue using iCDC-R2D2. The resulting network and matcher achieved the greatest reduction of the performance gap with respect to default-R2D2, 66.0% when considering AUC $@20^\circ$, and outperformed SuperPoint+LightGlue by

TABLE A.1

BENCHMARKING LONG-TERM VISUAL LOCALIZATION. PERFORMANCES OF NETWORK MATCHER PAIRS ARE COMPARED. NN MEANS EUCLIDEAN DISTANCE-BASED NEAREST NEIGHBOR MATCHING. LGLUE REFERS TO LIGHTGLUE [62]. NOV. MEANS THE NETWORK WAS TRAINED USING RENDERED NOVEL VIEWS OF OUR NERFS.

Scenes	CMU Seasons Extended [19]						RobotCar-Seasons v2 [18]			Aachen Day-Night v1.1 [17]		
	Traj.	Urban	Suburban	Park	Day All	Night All	Day	Night				
SuperPoint+LGlu		96.1 / 98.8 / 99.6	93.3 / 96.0 / 97.9	89.1 / 92.7 / 95.0	63.9 / 93.8 / 99.3	24.5 / 45.9 / 56.6	90.0 / 96.0 / 99.3	75.9 / 90.6 / 98.4				
R2D2+NN	R2D2+NN	94.4 / 97.3 / 99.0	85.3 / 88.9 / 93.5	74.2 / 78.8 / 83.6	64.5 / 94.4 / 99.2	16.8 / 32.6 / 39.4	88.0 / 95.3 / 97.9	67.0 / 86.4 / 96.3				
iCDC-R2D2+NN	iCDC-R2D2+NN	96.5 / 98.8 / 99.4	93.4 / 94.9 / 97.1	87.5 / 90.3 / 92.9	64.9 / 94.6 / 99.4	24.5 / 45.5 / 52.2	87.7 / 93.7 / 96.6	67.5 / 84.3 / 93.7				
Nov.-iCDC-R2D2+NN	Nov.-iCDC-R2D2+NN	96.5 / 98.7 / 99.4	92.8 / 94.3 / 96.7	85.9 / 89.0 / 91.7	65.1 / 94.7 / 99.4	23.5 / 47.1 / 52.7	86.9 / 93.7 / 96.7	68.1 / 84.3 / 93.7				
Default-R2D2+(iCDC-LGlu)	Default-R2D2+(iCDC-LGlu)	86.4 / 90.5 / 94.9	77.7 / 83.5 / 92.3	53.2 / 60.5 / 74.5	63.9 / 93.8 / 99.0	5.6 / 12.1 / 29.1	70.3 / 81.1 / 89.3	28.3 / 39.3 / 59.7				
iCDC-R2D2+(iCDC-LGlu)	iCDC-R2D2+(iCDC-LGlu)	90.9 / 94.1 / 97.3	87.9 / 91.8 / 96.5	74.0 / 80.0 / 88.6	64.3 / 94.2 / 99.4	13.5 / 23.5 / 39.4	67.1 / 76.6 / 86.4	27.7 / 39.8 / 62.3				
Nov.-iCDC-R2D2+(iCDC-LGlu)	Nov.-iCDC-R2D2+(iCDC-LGlu)	89.8 / 93.5 / 97.3	86.0 / 90.3 / 95.8	71.8 / 78.6 / 88.3	64.4 / 94.6 / 99.6	16.8 / 31.9 / 49.2	69.8 / 80.6 / 88.7	37.2 / 50.8 / 71.7				

18.9%. On top of highlighting the power of learned matchers, the large gap between default-R2D2 and iCDC-R2D2 using LightGlue underscores the importance of data-centric research for training local features.

Using the same setup, i.e. same camera mounted on a car facing forward, these networks and matchers are close to closing the gap even when generalizing to new scenes. This is achieved, on one hand thanks to the features, but on the other hand by overfitting to the matching patterns, and therefore, can not generalize well to new setups. We highlight this by providing an ablation study on the long-term localization benchmark in Tab. A.1. SupPoint+LightGlue outperforms all other network-matcher pairs, with only iCDC-R2D2+NN being competitive in all datasets except Aachen Day-Night. Without random cropping in the LightGlue training, the performances were significantly worse, and introducing cropping only makes the results comparable but not competitive, except on RobotCar day-time data, because RobotCars rear-view camera has similar characteristics to the 4Seasons setup. Similarly to our benchmark, training LightGlue using iCDC-R2D2 instead of default-R2D2 yields a meaningful performance increase, except in Aachen Day-Night. Introducing novel views into training improves performance on Aachen Day-Night, as well as on RobotCar Night. Surprisingly, however, its performance drops in CMU.

B. Additional Insights and Details

NeRF training details. In Sec. V, we detail our approach of dividing each scene into individual blocks along each trajectory and training a smaller NeRF for each block. Per trajectory, merging these smaller NeRFs results in a trajectory-NeRF. In our setup, however, we constructed a separate map for each trajectory pair, to generate more refined relative poses across trajectories. To remove scaling issues when comparing depth maps across trajectories from different maps, we train a trajectory-NeRF per trajectory for every trajectory pair. Furthermore, we expanded each block with buffer regions to improve the rendering quality of the NeRFs near the block edges. The buffer region is a 20% extension of each block along each dimension.

In Instant-npg [61], the representation space is divided into multiresolution hashed boxes, where each box is discretized into 128 steps along each axis. During ray-casting, there is one point sampled in each discrete step. Each box shares the same

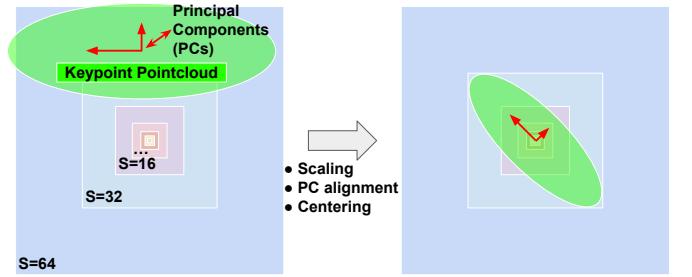


Fig. A.2. **Optimized Bounding Box Alignment.** The keypoint pointcloud is the set of 3D points a map corresponding to a scene block. Initially, the pointclouds are scaled. Using Principal Component Analysis, we calculate the pointcloud's principal components. To fit as much of the scene into the smallest box possible, we center and transform the pointcloud to align the principal components with the body diagonal of the boxes.

center, and are multiples of two larger than the smallest box, therefore the resolution of each discrete step in each box is a multiple of two smaller than in the smallest box. The purpose of this multiscale encoding is to speed up training, sampling points at a much higher rate near the center of the scene, however, at each iteration of training, the sampling point is randomized within each discrete step, therefore, even regions further from the center can be well represented with enough iterations.

For best results, it is important to maximize the number of relevant scene points in the smallest bounding boxes. For each block, we scaled the scene such that the central bounding box (scale 1) spanned 3m of the scene, along all axes. Therefore, the largest bounding box (scale 64) spans 192m of the scene, defining the bounds of the NeRF representation. The scene surfaces in each block should be centered in the boxes. We aggregate the 3D points from the map corresponding to each frame in a block, and filter out points further than 64m from the corresponding camera frames. We center the scene around the median of the resulting pointcloud. Furthermore, we align the body diagonal of the boxes with the principal component of the pointcloud to further maximize the number of points in the smallest boxes.

This process is illustrated in Fig. A.2. Furthermore, in Fig. A.3, we present a collection of examples of the distributions of the pointclouds centered around the bounding boxes. Excluding the points corresponding to the buffer regions, approximately 80% of the points are within the scale 16 bounding box.

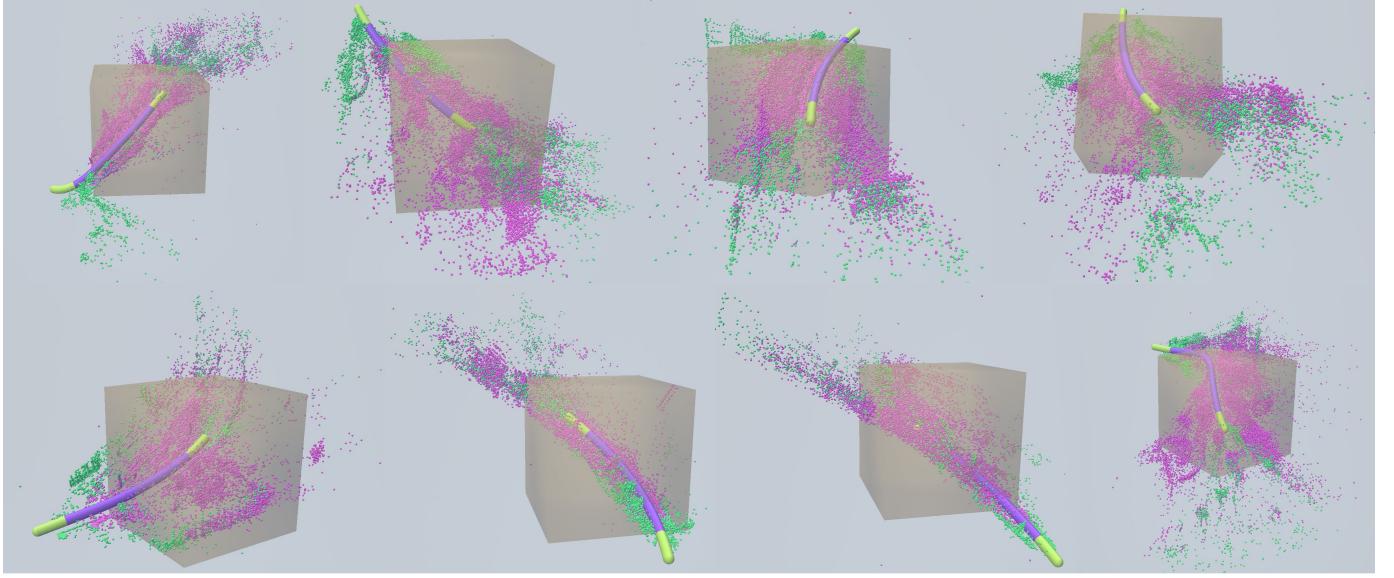


Fig. A.3. **Bounding Box Alignment Illustrations.** Displayed are scenes aligned within scale 16 bounding boxes. Camera trajectories are depicted as tubes, with the pointcloud representing the 3D scene for each camera frame. Frames within blocks are shown as pink pointclouds and purple camera frames, while yellow camera frames and green pointclouds depict those in the buffer region.

Hyperparameters are selected per block to represent each block as accurately as possible. During training, certain training runs got stuck in local minima depending on initialization; the NeRF would estimate high density on the camera frame resulting in zero depth estimation. During a single training run, we reinitialize the training multiple times, each time transforming the scene slightly with respect to the bounding boxes. To evaluate the resulting trained NeRFs, we select the one that estimates depths that best fit the ground truths at corresponding keypoints. These ground truths are the extracted depths from our maps, which were used for depth supervision. If no runs reached an average depth error lower than a certain threshold, all trained NeRFs were discarded. In that case, training was either restarted with new initialization, or abandoned after too many failed training runs. The abandoned blocks accounted for approximately 3%. We train each NeRF for $3 \cdot 10^5$ iterations.

Adapting R2D2 loss for cross-domain correspondences.

To improve robustness to long-term localization performance, we use our generated correspondences to supervise feature extractors. We use R2D2 [8] as the main network for our ablation study. We also trained SiLK [11] to show generalization to other extractors.

For an input image, R2D2 extracts pixel-wise descriptors and two score maps. The first score map estimates the repeatability of the corresponding points, while the second score map estimates the reliability of the corresponding descriptors.

To optimize for repeatability, a cosine similarity loss \mathcal{L}_{cosim} and peakiness loss \mathcal{L}_{peaky} are deployed. The repeatability loss \mathcal{L}_{rep} is defined as the summation of the two losses:

$$\mathcal{L}_{rep} = \mathcal{L}_{peaky} + \mathcal{L}_{cosim} \quad (1)$$

The peakiness loss enforces that the heatmap maintains a “peaky” distribution, an essential addend in the repeatability

loss function. For a more in-depth understanding, please refer to the original paper [8]. The cosine similarity loss, on the other hand, penalizes the network, when the reference repeatability heatmap \mathbf{S} and the query repeatability heatmap \mathbf{S}' are dissimilar at corresponding pixel locations. In R2D2 [8], it is proposed to apply and average the cosine similarity on image patches, however, for our approach with filtered correspondences, we do not follow this patched approach. We instead use the following cosine similarity loss function:

$$\mathcal{L}_{cosim}(\mathcal{C}) = 1 - cosim(\mathbf{S}[p_{1:N}], \mathbf{S}'[p'_{1:N}]), \quad (2)$$

where $\mathcal{C} = \{(p_i, p'_i)\}_{i \in N}$ is the set of N pixel correspondences used for training. $\mathbf{S}[p_{1:N}]$ is a vector of heatmap values \mathbf{S} at reference pixel locations, and $\mathbf{S}'[p'_{1:N}]$ is the vector of heatmap values \mathbf{S}' at corresponding query pixel locations. To optimize for reliable descriptors, the descriptors and the reliability heatmap are jointly optimized, using the reliability loss, which is defined as:

$$\mathcal{L}_{rel} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{AP\kappa}(p_i), \quad (3)$$

where

$$\mathcal{L}_{AP\kappa}(p) = 1 - (AP(p)\mathbf{R}_p + \kappa(1 - \mathbf{R}_p)). \quad (4)$$

\mathbf{R}_p is the estimated reliability value of pixel p , and AP is a differentiable approximation of the Average Precision (AP) metric [64]. $1 - AP(p)$ is then a ranking loss, which is minimized when the descriptor of p is most similar to its positive anchor p' and dissimilar to its negative anchors, randomly sampled from the set of points in the query image that maintain a specified minimum distance from p' . Eq. 4 modifies this loss by introducing the repeatability estimation of the reference pixel. The purpose of optimizing the reliability

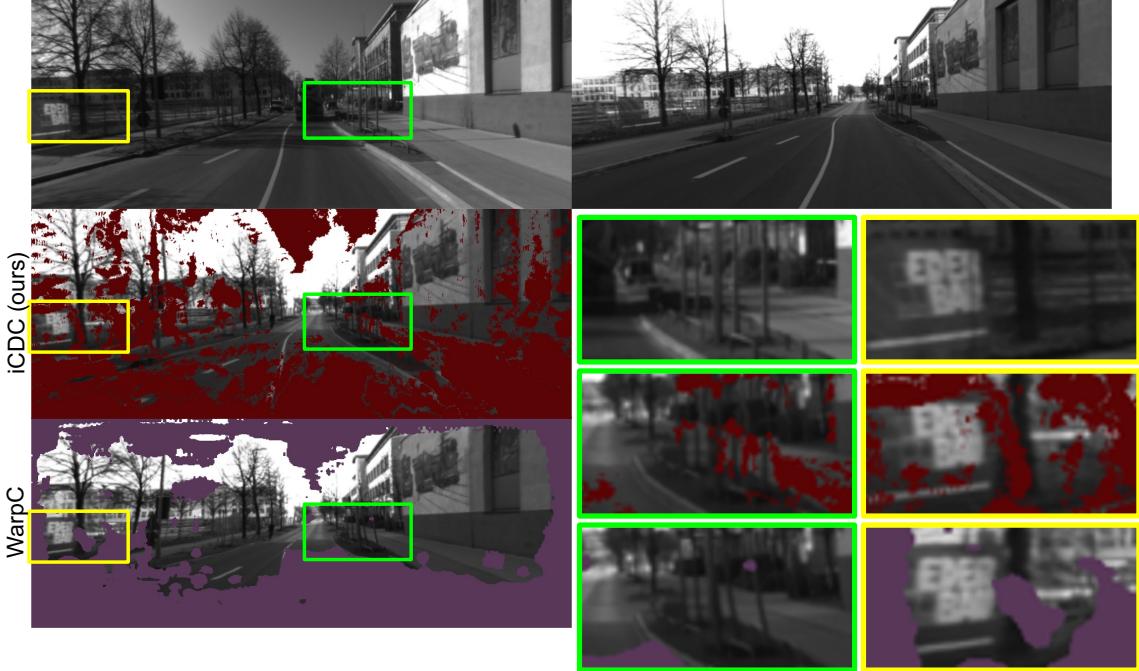


Fig. A.4. **Additional Comparison Between Our NeRF Correspondences and WarpC [43] Correspondences.** Correspondences are generated for an image pair (top row). The left column’s second and third rows show WarpC and iCDC correspondences respectively. More specifically, the right-hand side image is warped onto the left-hand side image following the correspondences. We zoom in on windows for detailed examination. In the case of accurate correspondences, the warped frames align with the left-hand side frame.

and the descriptors together, is to estimate a low reliability if there is an anticipated low $AP(p)$ score.

The overall loss is the then the sum of the repeatability and the reliability loss terms:

$$\mathcal{L}(\mathcal{P}, \mathcal{P}') = \mathcal{L}_{rep}(\mathcal{P}, \mathcal{P}') + \mathcal{L}_{rel}(\mathcal{P}, \mathcal{P}') \quad (5)$$

We introduced iCDC to supervise extractors to generate domain invariant descriptors, and to extract keypoints that are useful for both cross- and intra-domain localization. While supervising with cross-domain correspondences are vital for extracting domain invariant descriptors, we propose to supervise keypoint detection with only intra-domain correspondences. Penalizing extracted keypoints that are not shared across a certain visual domain pair is not valid, because they might be useful keypoints across different domain pairs, or across intra-domain image pairs. Specifically, we turn off the repeatability loss in the cross-domain case and adapt the reliability loss. We adapt the overall loss function 5 to:

$$\mathcal{L}_{\mathcal{D}, \mathcal{D}'}(\mathcal{P}, \mathcal{P}') = \mathbb{1}_{\mathcal{D}=\mathcal{D}'} \cdot \mathcal{L}_{rep}(\mathcal{P}, \mathcal{P}') + \mathcal{L}_{rel, \mathcal{D}, \mathcal{D}'}(\mathcal{P}, \mathcal{P}'), \quad (6)$$

where \mathcal{D} is the viewing condition index of \mathcal{P} , and $\mathbb{1}$ is the indicator function. $\mathcal{L}_{rel, \mathcal{D}, \mathcal{D}'}$ is identical do \mathcal{L}_{rel} , except, in the case where $\mathcal{D} \neq \mathcal{D}'$, the gradients are not propagated through the reliability output of the network.

Additional results. In Sec. VII-B of the main paper, we presented results comparing correspondences generated by a dense correspondence network (WarpC [43]) and our iCDC method. In Fig. A.4 we present an additional example, highlighting the improved performance of our correspondence generation method across domains.

Furthermore, in Sec. VII-E, we presented our key results, evaluating our extractors on the long-term localization benchmark. In Tabs. A.2 and A.3, we present more detailed results for the same networks on the CMU Seasons [19] and RobotCar-Seasons benchmarks [18]. In CMU-Seasons, iCDC-R2D2 significantly outperforms other networks across all visual domains. On the other hand, the results are more balanced in RobotCar-Seasons. Similar to the less detailed results in the main paper, iCDC-R2D2 outperforms other extractors in the day-time scenario, however, not across all visual domains. In the main results, we found that the key improvements were in the night-time scenario. In these detailed results, we see that the greatest performance improvement occurs in the rainy night-time scenario.

REFERENCES

- [1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis. (IJCV)*, vol. 60, 2004.
- [2] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2006.
- [3] S. Leutenegger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in *Int. Conf. Comput. Vis. (ICCV)*, 2011.
- [4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *Int. Conf. Comput. Vis. (ICCV)*, 2011.
- [5] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.
- [6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE transactions on robotics*, vol. 31, 2015.
- [7] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh. (CVPRW)*, 2018.
- [8] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, “R2d2: Reliable and repeatable detector and descriptor,” *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2019.

TABLE A.2
DETAILED CMU SEASONS [19] BENCHMARK RESULTS

	Overcast	Sunny	Foliage	Mixed Foliage	No Foliage	Low Sun	Cloudy	Snowy
SuperPoint [7]	85.1 / 87.9 / 92.7	74.5 / 79.0 / 87.0	78.0 / 81.7 / 88.8	84.7 / 87.7 / 92.4	95.0 / 96.4 / 97.8	86.9 / 89.4 / 93.3	90.3 / 92.2 / 95.9	91.1 / 93.0 / 95.7
R2D2 [8]	87.7 / 90.9 / 94.2	77.2 / 82.1 / 87.5	80.7 / 85.0 / 89.6	85.2 / 89.0 / 92.9	94.8 / 96.6 / 97.9	86.8 / 90.2 / 93.6	91.6 / 94.1 / 96.2	90.8 / 93.5 / 95.8
iCDC-R2D2	93.7 / 95.5 / 97.0	87.6 / 91.4 / 94.7	89.2 / 92.4 / 95.2	94.0 / 95.7 / 97.0	98.1 / 98.9 / 99.1	94.7 / 96.2 / 97.3	95.6 / 96.8 / 98.1	96.9 / 97.7 / 98.2
SiLK [11]	39.3 / 43.8 / 52.3	33.6 / 37.8 / 44.0	35.9 / 40.1 / 47.1	36.3 / 40.7 / 49.4	56.1 / 60.3 / 67.8	42.5 / 46.6 / 54.8	44.8 / 49.4 / 57.8	43.1 / 47.3 / 56.4
iCDC-SiLK	80.4 / 83.3 / 87.1	72.0 / 76.5 / 81.2	74.4 / 78.2 / 82.6	80.1 / 83.3 / 87.4	91.3 / 93.3 / 95.1	83.0 / 85.8 / 89.4	84.4 / 86.9 / 89.6	85.5 / 88.2 / 91.0

TABLE A.3
DETAILED ROBOTCAR-SEASONS [18] BENCHMARK RESULTS

	Night Rain	Overcast Winter	Sun	Rain	Snow	Dawn	Dusk	Night	Overcast Summer
SuperPoint [7]	12.8/23.6/27.1	57.3/97.0/ 100	62.1/87.9/99.1	78.0/94.6/100	68.8/97.7/ 100	56.8/91.6/96.5	79.2/95.4/ 100	15.0/31.4/39.8	47.9/92.9/98.6
R2D2 [8]	16.3/30.0/33.5	58.5/98.2/ 100	60.7/ 89.3/99.1	78.0/94.6/ 100	67.9/98.1/ 100	57.3/91.6/96.5	79.7/ 95.9/100	17.3/35.0/44.7	50.2/94.3/99.5
iCDC-R2D2	31.0/49.8/54.2	59.1/97.6/100	59.8/88.4/98.2	78.0/ 94.6/100	68.8/98.1/ 100	60.4/ 95.2/100	79.7/95.9/ 100	18.6/41.6/50.4	48.8/93.4/98.1
SiLK [11]	10.3/16.7/20.2	55.5/98.2/100	47.3/70.1/84.4	77.6/94.6/100	70.2/97.2/99.1	60.8/93.4/98.7	80.2/95.4/100	14.6/19.9/23.0	44.5/89.6/92.4
iCDC-SiLK	15.3/30.0/36.5	54.3/ 98.2/100	58.9/87.1/98.2	78.5/94.6/100	70.2/98.6/100	61.2/95.2/100	79.7/95.4/ 100	15.5/23.9/32.3	48.3/90.0/96.7

- [9] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, “D2-net: A trainable cnn for joint description and detection of local features,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.
- [10] M. Kanakis, S. Maurer, M. Spallanzani, A. Chhatkuli, and L. Van Gool, “Zippypoint: Fast interest point detection, description, and matching through mixed precision discretization,” in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh. (CVPRW)*, 2023.
- [11] P. Gleize, W. Wang, and M. Feiszli, “Silk—simple learned keypoints,” *arXiv preprint*, 2023.
- [12] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, “Benchmarking 6dof outdoor visual localization in changing conditions,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.
- [13] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls, “Image matching across wide baselines: From paper to practice,” *Int. J. Comput. Vis. (IJCV)*, vol. 129, 2021.
- [14] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, “Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.
- [15] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers, “4seasons: A cross-season dataset for multi-weather slam in autonomous driving,” in *In Proceedings of the German Conference on Pattern Recognition*, 2020.
- [16] J. Tang, H. Kim, V. Guizilini, S. Pillai, and R. Ambrus, “Neural outlier rejection for self-supervised keypoint learning,” in *Int. Conf. Learn. Represent. (ICLR)*, 2020.
- [17] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, “Image retrieval for image-based localization revisited.” in *Brit. Mach. Vis. Conf. (BMVC)*, vol. 1, 2012.
- [18] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 year, 1000 km: The oxford robotcar dataset,” *The International Journal of Robotics Research*, vol. 36, 2017.
- [19] H. Badino, D. Huber, and T. Kanade, “Visual topometric localization,” in *IEEE Intelligent vehicles symposium (IV)*, 2011.
- [20] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2010.
- [21] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, “Universal correspondence network,” in *NIPS*, 2016.
- [22] M. Altilawli, “Pixselect: Less but reliable pixels for accurate and efficient localization,” in *International Conference on Robotics and Automation (ICRA)*, 2022.
- [23] M. Tyszkiewicz, P. Fua, and E. Trulls, “Disk: Learning local features with policy gradient,” *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 33, 2020.
- [24] M. E. Fathy, Q.-H. Tran, M. Z. Zia, P. Vernaza, and M. Chandraker, “Hierarchical metric learning and matching for 2d and 3d geometric correspondences,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2018.
- [25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2014.
- [26] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: Cnn architecture for weakly supervised place recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.
- [27] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, “Deep learning features at scale for visual place recognition,” in *International Conference on Robotics and Automation (ICRA)*, 2017.
- [28] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli, “Learning context flexible attention model for long-term visual place recognition,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, 2018.
- [29] T. Weyand, A. Araujo, B. Cao, and J. Sim, “Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.
- [30] A. Farahani, S. Voghieri, K. Rasheed, and H. R. Arabnia, “A brief review of domain adaptation,” *Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020*.
- [31] H. Daumé III, “Frustratingly easy domain adaptation,” *arXiv preprint*, 2009.
- [32] G. Csurka, “Domain adaptation for visual applications: A comprehensive survey,” *arXiv preprint*, 2017.
- [33] H. Li, S. J. Pan, S. Wang, and A. C. Kot, “Domain generalization with adversarial feature learning,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.
- [34] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, “Domain generalization in vision: A survey,” *arXiv preprint*, 2021.
- [35] F. Shen, Z. Pataki, A. Gurram, Z. Liu, H. Wang, and A. Knoll, “Loopda: Constructing self-loops to adapt nighttime semantic segmentation,” in *Winter Conference on Applications of Computer Vision (WACV)*, 2023.
- [36] X. Guo, C. Yang, B. Li, and Y. Yuan, “Metacorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021.
- [37] F. Shen, A. Gurram, Z. Liu, H. Wang, and A. Knoll, “Diga: Distil to generalize and then adapt for domain adaptive semantic segmentation,” *arXiv preprint*, 2023.
- [38] X. Wu, Z. Wu, H. Guo, L. Ju, and S. Wang, “Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021.
- [39] D. Brüggemann, C. Sakaridis, P. Truong, and L. Van Gool, “Refign: Align and refine for adaptation of semantic segmentation to adverse conditions,” in *Winter Conference on Applications of Computer Vision (WACV)*, 2023.
- [40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.
- [41] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, “Contrastive adaptation network for unsupervised domain adaptation,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.
- [42] G. French, M. Mackiewicz, and M. Fisher, “Self-ensembling for visual domain adaptation,” *arXiv preprint*, 2017.

- [43] P. Truong, M. Danelljan, F. Yu, and L. Van Gool, “Warp consistency for unsupervised learning of dense correspondences,” in *Int. Conf. Comput. Vis. (ICCV)*, 2021.
- [44] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool, “Night-to-day image translation for retrieval-based localization,” in *International Conference on Robotics and Automation (ICRA)*, 2019.
- [45] I. Melekhov, G. J. Brostow, J. Kannala, and D. Turmukhambetov, “Image stylization for robust features,” *arXiv preprint*, 2020.
- [46] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, “From coarse to fine: Robust hierarchical localization at large scale,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.
- [47] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superglue: Learning feature matching with graph neural networks,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.
- [48] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys, “Pixel-perfect structure-from-motion with featuremetric refinement,” in *Int. Conf. Comput. Vis. (ICCV)*, 2021.
- [49] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “Loftr: Detector-free local feature matching with transformers,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021.
- [50] M. Larsson, E. Stenborg, L. Hammarstrand, M. Pollefeys, T. Sattler, and F. Kahl, “A cross-season correspondence dataset for robust semantic segmentation,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.
- [51] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, 2021.
- [52] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021.
- [53] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado, “Gaia-1: A generative world model for autonomous driving,” *arXiv preprint*, 2023.
- [54] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, “Clip-nerf: Text-and-image driven manipulation of neural radiance fields,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022.
- [55] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, “Nice-slam: Neural implicit scalable encoding for slam,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022.
- [56] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys, “Nicer-slam: Neural implicit scene encoding for rgb slam,” *arXiv preprint*, 2023.
- [57] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, “Nerf-supervision: Learning dense object descriptors from neural radiance fields,” in *International Conference on Robotics and Automation (ICRA)*, 2022.
- [58] H. Turki, D. Ramanan, and M. Satyanarayanan, “Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022.
- [59] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, “Block-nerf: Scalable large scene neural view synthesis,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022.
- [60] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, “Depth-supervised nerf: Fewer views and faster training for free,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022.
- [61] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM Trans. Graph.*, vol. 41, 2022.
- [62] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, “Lightglue: Local feature matching at light speed,” *arXiv preprint*, 2023.
- [63] Z. Li and N. Snavely, “Megadepth: Learning single-view depth prediction from internet photos,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.
- [64] K. He, Y. Lu, and S. Sclaroff, “Local descriptors optimized for average precision,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.



Zador Pataki recently completed his Master's degree and has now joined the Computer Vision and Geometry (CVG) lab at ETH Zurich to pursue a PhD. His research interests lie at the intersection of 3D Computer Vision and Deep Learning. During his Master's, he published papers in WACV and IROS.



Ziyuan Liu works as the chief scientist on AI robotics and simulation technologies at Huawei Cloud computing. Before joining Huawei, he was a tech lead on robotics at DOMO academy of Alibaba group. His research interests include simulation technologies and 3D robot vision. He has served as associate editor on IROS. In 2021 He launched the open cloud robot table top organization competition (OCRTOC) which is considered to be one of the first cloud-based robot competitions.



Mohammad Al tillawi is a PhD student at the Computer Vision Center (CVC) and the Autonomous University of Barcelona (UAB). He is conducting his research at Huawei Intelligent Robotics Cloud Computing lab in Munich. His research focuses on improving mapping and localization for robotics. Particularly, he aims to derive solutions that are accurate, real-time, and with low compute resources. Recently, he is focusing on leveraging novel view synthesis techniques towards improved mapping and localization. He has published in renown conferences, such as ICRA, IROS, and RA-L.



Menelaos Kanakis earned his PhD from the Computer Vision Lab (CVL) at ETH Zurich. His research is focused on developing deep learning models under real-life constraints such as algorithmic runtime, limited annotations, or performing inference on-board computationally limited hardware. He has published in several renown conferences, such as CVPR, ICCV, ECCV, and ICML.



Luc Van Gool is a full professor for Computer Vision at ETH Zurich and the KU Leuven. He leads research and teaches at both places. He has authored over 300 papers. Luc Van Gool has been a program committee member of several, major computer vision conferences (e.g. Program Chair ICCV'05, Beijing, General Chair of ICCV'11, Barcelona, and of ECCV'14, Zurich). His main interests include 3D reconstruction and modeling, object recognition, and autonomous driving. He received several Best Paper awards (eg. David Marr Prize '98, Best Paper CVPR'07). He received the Koenderink Award in 2016 and 'Distinguished Researcher' nomination by the IEEE Computer Society in 2017. In 2015 he also received the 5-yearly Excellence Prize by the Flemish Fund for Scientific Research. He was the holder of an ERC Advanced Grant (VarCity). Currently, he leads computer vision research for autonomous driving in the context of the Toyota TRACE labs in Leuven and at ETH, and has an extensive collaboration with Huawei on the issue of image and video enhancement.



Rémi Pautrat is a PhD student at the Computer Vision and Geometry (CVG) lab at ETH Zurich. His research interests are centered on local features and how to apply them to geometry tasks such as visual localization and 3D reconstruction. He focuses in particular on studying the invariance of local descriptors, and on higher-level features such as line segments, exploring ways to make the detection and matching pipelines as accurate and robust as possible. He has published in several renown conferences, such as CVPR, ICCV, ECCV, and ICRA.



Marc Pollefeys is a full professor in the Dept. of Computer Science of ETH Zurich since 2007 where he leads the Computer Vision and Geometry lab. He is also the director of the Microsoft Mixed Reality and Artificial Intelligence Zurich lab where he leads a team of scientists working on perception algorithms for HoloLens and Mixed Reality. His main area of research is computer vision, but he is also active in robotics, machine learning and computer graphics. One of his main research goals is to develop flexible approaches to capture visual representations of real world objects, scenes and events. Prof. Dr. Pollefeys has received several prizes for his research, including a Marr prize, an NSF CAREER award, a Packard Fellowship and a European Research Council Starting Grant. He is the author or co-author of more than 300 peer-reviewed publications. He was the General Chair of ICCV 2019 and ECCV 2014 and Program Co-Chair for CVPR 2009. Prof. Pollefeys has served on the Editorial Board of the IEEE Transactions on Pattern Analysis and Machine Intelligence, the International Journal of Computer Vision and Foundations and Trends in Computer Graphics and Computer Vision. He is a fellow of the IEEE.



Fengyi Shen is a PhD student at the Chair of Robotics, Artificial Intelligence and Embedded Systems at Technical University of Munich (TUM). He also works as a research intern in Huawei Munich Research Center and a collaborated research assistant in Embodied Perception and InterAction (EPIC) Lab at Peking University. His research interests include domain adaptive/generalizable and semi-supervised learning in computer vision, as well as generative models. He focuses in particular on exploring algorithms that can be trained in a label-efficient manner, aiming to reduce the effort of human annotation in computer vision tasks. He has published and served as reviewer in several renowned venues, such as CVPR, ECCV, BMVC, WACV and IEEE T-ITS.