

Self-Supervised Feature Learning for Long-Term Metric Visual Localization

Yuxuan Chen¹, Timothy D. Barfoot¹

Abstract—Visual localization is the task of estimating camera pose in a known scene, which is an essential problem in robotics and computer vision. However, long-term visual localization is still a challenge due to the environmental appearance changes caused by lighting and seasons. While techniques exist to address appearance changes using neural networks, these methods typically require ground-truth pose information to generate accurate image correspondences or act as a supervisory signal during training. In this paper, we present a novel self-supervised feature learning framework for metric visual localization. We use a sequence-based image matching algorithm across different sequences of images (i.e., experiences) to generate image correspondences without ground-truth labels. We can then sample image pairs to train a deep neural network that learns sparse features with associated descriptors and scores without ground-truth pose supervision. The learned features can be used together with a classical pose estimator for visual stereo localization. We validate the learned features by integrating with an existing Visual Teach & Repeat pipeline to perform closed-loop localization experiments under different lighting conditions for a total of 22.4 km.

Index Terms—Localization, Deep Learning for Visual Perception, Vision-Based Navigation

I. INTRODUCTION

LONG-TERM navigation across drastic appearance change is a challenge in visual localization. Traditional point-based localization approaches find correspondences between local features extracted from images by applying hand-crafted descriptors (e.g., SIFT, SURF, ORB [1]–[3]), then recover the full 6-DoF camera pose. However, such hand-crafted features are not robust under extreme appearance changes. To address this, Multi-Experience Visual Teach & Repeat [4], [5], a visual-based route-following framework, proposed to store intermediate experiences to achieve long-term localization.

Due to recent developments in the field of deep learning, many recent works use neural networks to directly predict relative poses [6]–[8] or absolute poses [9] from images for localization. Instead of directly learning poses from images, deep-learned interest-point detectors and descriptors [10]–[19] have gained popularity since they produce more accurate results than direct pose regression when combined with a

Manuscript received: August, 26, 2022; Revised November, 2, 2022; Accepted November, 29, 2022.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. We thank Mona Gridseth for her tremendous help with this project.

¹Yuxuan Chen and Timothy D. Barfoot are with the University of Toronto Institute for Aerospace Studies, University of Toronto, Canada {yuxuansherry.chen, tim.barfoot}@mail.utoronto.ca

Digital Object Identifier (DOI): see top of this page.



Fig. 1: Grizzly ground robot autonomously repeats a taught path with learned features despite severe lighting changes.

classical pose estimator. In addition, Gridseth and Barfoot [10] have proven the effectiveness of deep-learned features in the VT&R framework under different illumination conditions. However, these feature learning networks typically require training with accurate image correspondences that are extracted from known scene geometry or ground-truth camera poses. Although some methods simplify the problem by collecting data using a stationary camera [12], applying known homographic adaptation [13], [17], [18], or rendering synthetic training data [13], [16], the generalizability on unseen data is limited.

In this paper, we propose a novel self-supervised feature learning framework for long-term visual localization that does not require any ground-truth labels while preserving generalizability. We identify that one of the main challenges is to find accurate image correspondences in unstructured outdoor environment without ground-truth data. Since the ultimate goal is to learn features in a self-supervised manner, we do not want to presuppose the existence of local features. Especially, we want to avoid using deep-learned features that are pretrained on other datasets using known image correspondences. Hence, we adopt a sequence-based visual place recognition algorithm that works directly with whole-image descriptors to generate image pairs. Our overall pipeline consists of two stages: place recognition and feature learning. During the first stage, we generate training samples by performing sequence-based visual place recognition on data collected under different seasons and lighting conditions. During the second stage, we train a feature learning network using the sampled training stereo image pairs to predict keypoints with associated descriptors and scores in an Expectation-Maximization loop without any ground-truth pose supervision.

II. RELATED WORK

Traditionally, hand-crafted features have been commonly used for visual localization [1]–[3], but suffer from low re-

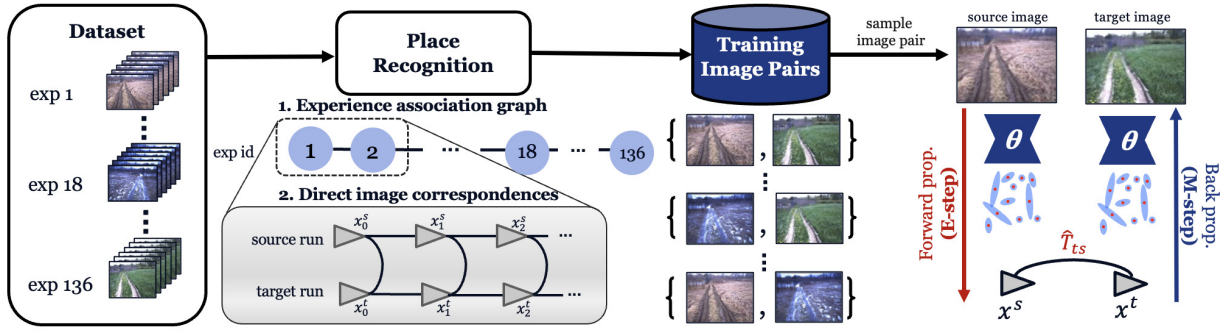


Fig. 2: **Overall pipeline.** The dataset consists of multiple sequences (i.e., experiences) of image data organized by their collection times. During the place-recognition stage, we directly match neighbouring experiences using a place-recognition algorithm, and find indirect image correspondences by constructing an experience-association graph. We sample image pairs using the predicted image correspondences for the subsequent feature-learning stage. During the E-step of feature learning, the network predicts keypoints from the source and target images, which can be used to compute the relative transform. During the M-step, we use the estimated relative transform as a supervisory signal to optimize the network weights.

peatability on images under dramatic appearance changes [20]. Experience-based visual navigation systems [5], [21], [22] attempt to bridge significant appearance changes by storing multiple appearances of the same location (i.e., experiences) and choosing the most relevant experiences for feature matching during online operation.

To address the deficiencies of hand-crafted features, there has been a wide range of deep-learning-based approaches for pose estimation. Some methods tackle camera localization by training a CNN that directly learns relative poses [6]–[8] or absolute poses [9] from images. However, learning pose directly from image data can struggle with accuracy [23].

Other works focus on only learning the visual features or descriptors [10]–[19], [24], which can be integrated with a classical pose estimator for localization. For instance, Gridseth and Barfoot [10] and Sun et al. [18] have proven the effectiveness of deep-learned features against illumination changes by integrating with the VT&R pipeline for long-term visual localization. However, training networks for visual localization generally requires accurate image correspondences extracted from ground-truth camera poses or known scene geometry. Although Kasper et al. [19] proposed an unsupervised transform consistency loss to train features, it still requires GPS to ensure the query and reference images have sufficient visual overlap. Alternatively, Verdie et al. [12] use a stationary camera to observe the same scene under different illumination conditions, whereas other methods obtain image correspondences from either using synthetically rendered data [13], [16] or applying known transformations (e.g., colour shift, homographic transform) to real-world images [13], [17], [18]. However, these approaches limit the generalizability of the learned model on unseen data.

In this paper, we propose to find image correspondences using a visual place-recognition algorithm without any labels. We will discuss some relevant place-recognition approaches in the rest of the section. Since our goal is to sample training data for feature learning in a self-supervised manner, we restrict our discussion to hand-crafted algorithms that do not involve any learning or human annotations.

Lowry et al. [25] provide a thorough overview of existing methods for place recognition. Traditional bag-of-words-based FAB-MAP2 [26] shows good performance, but tends to fail in the presence of dramatic appearance changes [27] due to the limited repeatability of local hand-crafted keypoints [1], [2] in changing environments [20]. In order to improve the performance of place recognition under appearance changes, various approaches have been developed to exploit the sequential nature of the data [22], [28], [29]. SeqSLAM [29] is one of the most recognized sequence-based algorithms that achieves significant performance improvements over FAB-MAP2 under appearance changes. SeqSLAM finds image correspondences by constructing a pairwise similarity matrix between the local query image sequence and a database image sequence without the need for keypoint extraction. Hansen et al. [30] further improve the overall flexibility of the sequence alignment procedure by using a hidden Markov model (HMM) to formulate a graph-based sequence search method in the similarity matrix. In order to match image sequences under severe seasonal changes more efficiently, Naseer et al. [31] formulate sequence matching as a minimum cost flow problem in an offline data-association graph. Vysotska et al. [32] extended this idea to build a data-association graph online using a pretrained CNN for feature extraction, and later proposed a hashing-based relocalization strategy [33] to improve data association for flexible trajectories. Additionally, Neubert et al. [34] detect similarities within the database to avoid unnecessary image matching between query and database images.

For our purposes, we adopt SeqSLAM aided by Visual Odometry (VO) to find direct image correspondences between temporally neighbouring experiences, and indirectly link further experiences by building an offline experience-association graph.

III. METHODOLOGY

We propose a two-stage pipeline as shown in Figure 2, which includes: 1) place recognition and 2) self-supervised feature learning. During the place-recognition stage, we build a locally connected experience-association graph by matching

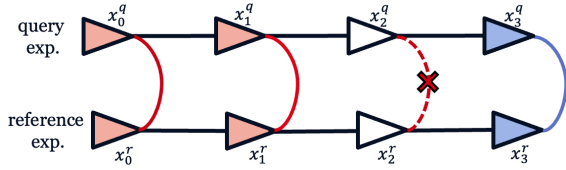


Fig. 3: **Direct image correspondences.** The black horizontal edges represent the relative transforms between frames (known from VO). The red vertical edges represent candidate matches suggested by SeqSLAM. The red solid edges are validated by VO, whereas the dotted edges are rejected by VO. After the candidate match is rejected, VO can also potentially find a new match in the reference sequence that is closest to the query image, which is represented by a blue edge.

neighbouring experiences with gradual perceptual changes, where we can sample image pairs from direct or indirect image correspondences between experiences using a graph-search algorithm. Our fully differentiable feature-learning pipeline takes a pair of source and target stereo images generated from the previous stage, and estimates the relative pose, $\mathbf{T}_{ts} \in SE(3)$, between their corresponding frames. During the feature-learning stage, we train a neural network in an Expectation-Maximization loop. In the E-Step, the network predicts sparse local keypoints with associated descriptors and scores for both input images. We find point correspondences between the predicted keypoints from the source and target images, which are used in a differentiable pose estimator to estimate the relative pose change. In the M-step, we construct a self-supervised keypoint loss using the estimated relative keypoint loss as a supervisory signal to optimize the network weights.

A. Place Recognition

We define the set of training data $\mathcal{D} = (s_0, \dots, s_M)$ as a temporally ordered set of M experiences, where each experience is defined as a sequence of N consecutive images $s_i = (x_0^i, \dots, x_N^i)$. The ultimate goal is to generate direct or indirect image correspondences between any two experiences in the dataset \mathcal{D} . However, matching images of the same place under appearance change is a non-trivial task, as the appearance of the same place is likely to change substantially due to different illumination conditions and seasonal changes. A naive approach is to directly match images between experiences using proximity provided by Visual Odometry (VO). However, this results in poor image correspondences for longer routes due to drift errors in VO. Alternatively, we can exploit the sequential nature of the data and use SeqSLAM [29] to match different image sequences using patch-normalized image representation, but face two main problems:

- SeqSLAM is capable of matching experiences with gradual appearance changes but struggles with significant appearance gap.
- The raw SeqSLAM matching results may contain outliers and discontinuities.

To mitigate these issues, we combine SeqSLAM with VO to improve the direct matching results, and construct an

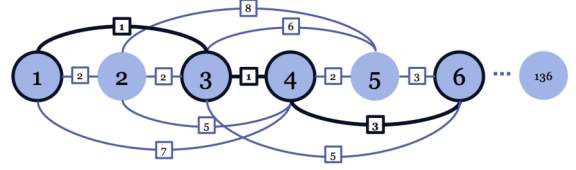


Fig. 4: **Experience-Association Graph.** Each vertex represents an experience, which is locally connected to 3 previous neighbours, and each edge represents direct image correspondences. The assigned cost term on each edge reflects the quality of the match. To compute indirect image correspondences between experience 1 and 6, the minimum cost path is indicated in black.

experience-association graph to extract indirect image correspondences between experiences. To sufficiently limit scope, we make the following assumptions:

- All experiences roughly follow the same trajectory with the same starting and ending positions.
- All experiences are temporally ordered by their collection times. Nearby experiences are collected within a relatively short time frame, thus having similar appearances compared to further experiences.

1) *Direct Image Correspondences:* According to the second assumption, we assume the temporally neighbouring experiences are collected within a shorter time frame, such that the appearance changes are small enough for us to find direct image correspondences using SeqSLAM and VO. For each query experience s_i , it is directly matched with k previous experiences $\{s_{i-k}, \dots, s_{i-1}\}$, which are denoted as the reference experiences.

As shown in Figure 5, SeqSLAM constructs a difference matrix by computing image-by-image dissimilarity scores between the two sequences using full-image descriptors, then finds raw image matches through the full matching matrix with the smallest sum of dissimilarity scores. We further validate the matching results using VO to reject outliers as illustrated in Figure 3.

For each image in the query experience, we use the raw SeqSLAM matching results to suggest a candidate match in the reference experience, then validate the match using VO. Assuming we have successfully matched and validated the image pair $\{q_0, r_0\}$ using SeqSLAM and VO, we consider the image pair as the last validated match m , then proceed to the next image in the query experience, q_1 . After retrieving the candidate match r_1 from the reference experience using SeqSLAM, we use VO to estimate the relative transform between the last validated image and the current image in the query and reference experiences, namely $\hat{\mathbf{T}}_{q_1, q_0}$ and $\hat{\mathbf{T}}_{r_1, r_0}$. Since the last validated image pair $m = \{q_0, r_0\}$ are two images of the same place, we can approximate the relative transform $\hat{\mathbf{T}}_{r_0, q_0}$ with an identity matrix \mathbf{I} . As a result, the relative transform between the query and reference images can

be estimated as:

$$\begin{aligned} \hat{\mathbf{T}}_{r_1, q_1} &= \hat{\mathbf{T}}_{r_1, r_0} \hat{\mathbf{T}}_{r_0, q_0} \hat{\mathbf{T}}_{q_1, q_0}^{-1} \\ &\approx \hat{\mathbf{T}}_{r_1, r_0} \hat{\mathbf{T}}_{q_1, q_0}^{-1} = \begin{bmatrix} \mathbf{C}_{r_1, q_1} & \mathbf{r}_{r_1, q_1} \\ \mathbf{0}^T & 1 \end{bmatrix}. \end{aligned} \quad (1)$$

The absolute distance between q_1 and r_1 can be computed as:

$$d_{r_1, q_1} = \|\mathbf{r}_{r_1, q_1}\|_2^2. \quad (2)$$

We compare the distance d_{r_1, q_1} with a pre-specified threshold value e to check if the candidate match is consistent with VO. We discuss two emerging cases as follows:

- $d_{r_1, q_1} \leq e$. The candidate match suggested by SeqSLAM is validated by VO. Hence, we update the last validated match m to be $\{q_1, r_1\}$ when processing the next image in the query sequence.
- $d_{r_1, q_1} > e$. The candidate match suggested by SeqSLAM is rejected by VO. As a replacement, we retrieve a new image r_j from the reference experience such that the distance between r_j and q_1 is minimized. Note that we do not update the last validated match m in this case.

Since the last validated match m is frequently updated, we usually only need to compute VO for a relatively short time frame starting from the last validated match. Hence, the matching results are less likely to be affected by VO drift.

2) *Indirect Image Correspondences*: According to the second assumption, relatively similar experiences are temporally closer together, whereas dissimilar experiences are further apart. Hence, we can find image correspondences between further experiences indirectly using bridging experiences. After finding direct image correspondences between neighbouring experiences, we can build a locally connected graph $G = \{V, E\}$, where V is a set of vertices representing all experiences in the dataset, and E is a set of edges representing direct image correspondences between the vertices. Each vertex is connected to k previous vertices, and Figure 4 shows an example when $k = 3$.

We assign a cost term to each edge in E to indicate the quality of the match. The cost term is defined as the number of frames rejected by VO out of the total number frames in the source sequence, where a higher cost represents a poorer match and a lower cost represents a better match. To generate image correspondences between two experiences that are not directly connected, we can find the minimum-cost path in the graph using a graph-search algorithm to indirectly extract image correspondences.

B. Feature Learning

1) *Network Architecture*: Our network is adapted from the architecture presented by [10], [24], which is a U-Net style convolutional encoder-multi-decoder architecture to output keypoints, descriptors, and scores based on visual inputs. The encoder is a VGG16 network [35] pretrained on the ImageNet dataset [36], truncated after the *conv_5_3* layer.

The keypoint-locations decoder predicts the sub-pixel locations of each sparse 2D keypoint, $\mathbf{q} = [u_l, v_l]^T$, in the left stereo image. To achieve this, we equally divide the image into 16×16 square cells, with each generating a single candidate

keypoint. We then apply a spatial softmax on each cell and take a weighted average of the pixel coordinates to return the sub-pixel keypoint locations.

In addition, the network computes the scores by applying a sigmoid function to the output of the second decoder branch. The scores s are mapped to $[0, 1]$, which predict how useful a keypoint is for pose estimation. Finally, we generate dense descriptors for each pixel by resizing and concatenating the output of each encoder layer, which results in a descriptor vector, $\mathbf{d} \in \mathbb{R}^{960}$.

2) *E-Step: Pose Estimation*: After generating N keypoint predictions for the source image, we need to perform data association between the keypoints in the source and target images by performing a dense search for optimum keypoint locations in the target image. For each keypoint in the source image, we compute a matched point in the target image by taking the weighted sum of all image coordinates in the target image as follows:

$$\hat{\mathbf{q}}_t^i = \sum_{j=1}^M \sigma(\tau f_{\text{zncc}}(\mathbf{d}_s^i, \mathbf{d}_t^j)) \mathbf{q}_t^j, \quad (3)$$

where M is the total number of pixels in the target image, $f_{\text{zncc}}(\cdot)$ computes the zero-normalized cross correlation (ZNCC) between the descriptors, and $\sigma(\cdot)$ takes the temperature-weighted softmax with τ as the temperature. Finally, we find the descriptor, $\hat{\mathbf{d}}_t^i$, and score, \hat{s}_t^i , for each computed target keypoint using bilinear interpolation.

For the matched 2D keypoints, we compute their corresponding 3D coordinates using an inverse stereo camera model. The camera model, $\mathbf{g}(\cdot)$, maps a 3D point, $\mathbf{p} = [x, y, z]^T$, in the camera frame to a left stereo image coordinate, \mathbf{q} , as follows:

$$\mathbf{y} = \begin{bmatrix} u_l \\ v_l \\ d \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ d \end{bmatrix} = \mathbf{g}(\mathbf{p}) = \begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 0 & f_u b \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (4)$$

where f_u and f_v are the horizontal and vertical focal lengths in pixels, c_u and c_v are the camera's horizontal and vertical optical center coordinates in pixels, $d = u_l - u_r$ is the disparity obtained from stereo matching, and b is the baseline in metres. We use the inverse stereo camera model to get each keypoint's 3D coordinates:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{g}^{-1}(\mathbf{y}) = \frac{b}{d} \begin{bmatrix} u_l - c_u \\ \frac{f_u}{f_v}(v_l - c_v) \\ f_u \end{bmatrix}. \quad (5)$$

Given the corresponding 2D keypoints $\{\mathbf{q}_s^i, \hat{\mathbf{q}}_t^i\}$ from the source and target images, we use (5) to compute their 3D coordinates, $\{\mathbf{p}_s^i, \hat{\mathbf{p}}_t^i\}$. In addition, their descriptors and scores are: $\{\mathbf{d}_s^i, \hat{\mathbf{d}}_t^i\}$, and $\{s_s^i, \hat{s}_t^i\}$.

We then perform RANSAC to reject outliers prior to pose estimation to find features that are geometrically consistent with each other. This step is essential to provide a reasonable guess of the relative pose in the beginning of training to help the algorithm converge.

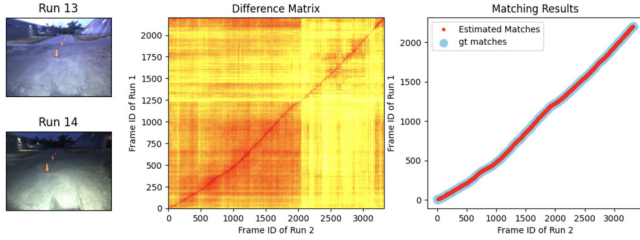


Fig. 5: **Frame-level topological matching results** between two neighbouring runs, run 13 and run 14, in the UTIAS-In-the-Dark dataset. The raw difference matrix generated by SeqSLAM algorithm and the processed matching results is shown.

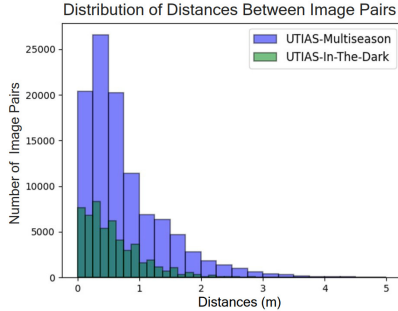


Fig. 6: **Distribution of distances between sampled image pairs** of Multiseason and In-the-Dark datasets.

Given the inlier 3D keypoints, we can estimate the relative pose from the source to the target by minimizing the following cost using Singular Value Decomposition (SVD):

$$J = \sum_{i=1}^N w^i \left\| (\mathbf{C}_{ts} \mathbf{p}_s^i + \mathbf{r}_t^{st}) - \hat{\mathbf{p}}_t^i \right\|_2^2, \quad (6)$$

where the weight for a matched point pair is a combination of the learned point scores and how well the descriptors match:

$$w^i = \frac{1}{2} (f_{z\text{ncc}}(\mathbf{d}_s^i, \hat{\mathbf{d}}_t^i) + 1) s_s^i \hat{s}_t^i. \quad (7)$$

Finally, we obtain the relative transform between the source and target images as:

$$\hat{\mathbf{T}}_{ts} = \begin{bmatrix} \mathbf{C}_{ts} & \mathbf{r}_t^{st} \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (8)$$

3) *M-Step: Feature Optimization*: Barnes and Posner [24] and Gridseth and Barfoot [10] use the ground-truth transform between the source and target images to construct a supervised pose loss. Since we do not rely on any ground-truth data, we directly use the estimated transform as a supervisory signal to construct a keypoint loss without any ground-truth pose information. We transform the predicted source keypoints \mathbf{p}_s^i to the target frame using the estimated relative transform $\hat{\mathbf{T}}_{ts}$, and define a keypoint loss as follows:

$$\mathcal{L} = \sum_{i=1}^N \left\| \hat{\mathbf{T}}_{ts} \mathbf{p}_s^i - \hat{\mathbf{p}}_t^i \right\|_2^2. \quad (9)$$

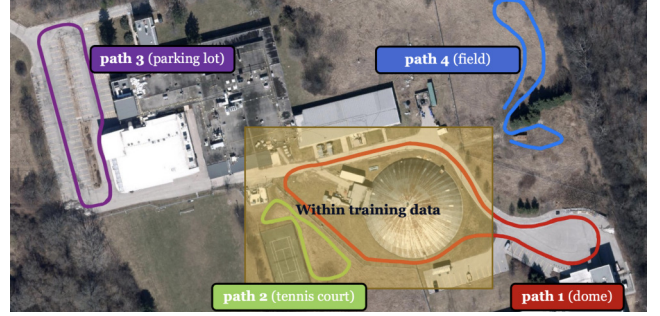


Fig. 7: **Training and testing paths**. The training paths from the datasets are highlighted by the yellow region. Path 1 (tennis court) is collected in the same region as the training data, whereas the bottom right portion of path 2 (dome), path 3 (parking lot) and path 4 (field) are not in training data.

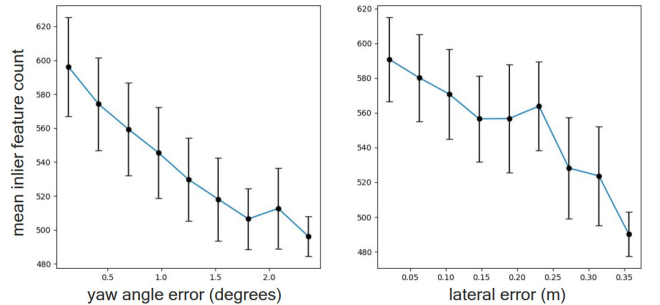


Fig. 8: **Median feature inlier count** (with 1σ bound) for different lateral and yaw angle errors of the UTIAS-In-the-Dark test set. The median feature inlier count is highly correlated with the path-following errors, which serves as a reasonable proxy for localization performance in VT&R.

IV. EXPERIMENTS

A. Datasets

For training, we use the same datasets as described in [10] to test localization for a route-following problem. The training data were collected using a Clearpath Grizzly robot with a Bumblebee XB3 camera. The robot autonomously repeats two different paths across drastic lighting and seasonal change using Multi-Experience VT&R [5]. VT&R stores stereo image keyframes as vertices in a spatio-temporal pose graph, where each edge contains the relative pose between a repeat vertex and the teach vertex. Unlike [10], we do not rely on the spatio-temporal pose graph to extract ground-truth relative transforms between vertices for sampling image pairs or for training.

We use data from two different paths for training, which are included in the In-the-Dark and Multiseason datasets respectively. The In-The-Dark dataset contains 39 runs of a path collected at the campus of UTIAS in summer 2016. The same path was repeated hourly for 30 hours, which captures incremental lighting changes throughout the day. The Multiseason dataset contains 136 runs of a path in an area with more vegetation. The path was consistently repeated from January until May 2017, which captures diverse seasonal and weather conditions throughout the year. Both datasets are publicly available.

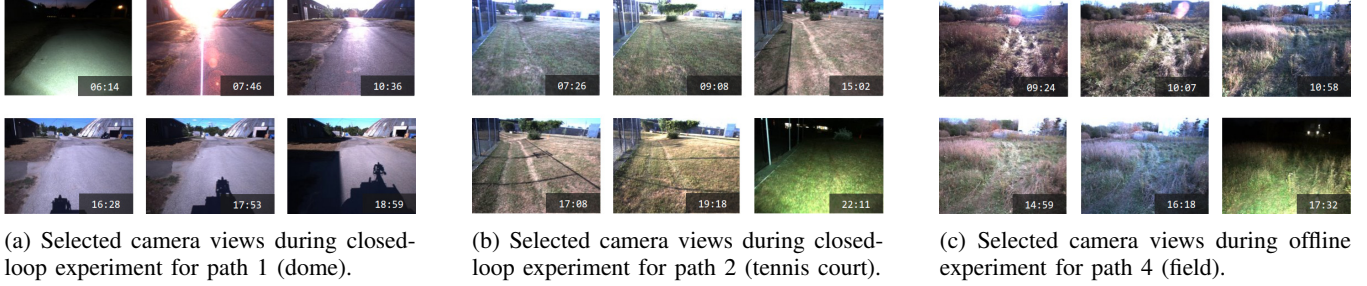


Fig. 9: **Selected camera views** from closed-loop and offline experiments labeled with collection time (hh:mm).

TABLE I: **Comparison of the path-following errors and the median number of inliers** for SuperPoint [13], D2-Net [11], Supervised U-Net [10], and our self-supervised method. We construct 6 teach-repeat pairs under different lighting changes for evaluation from the UTIAS-In-the-Dark test set, where we list the experience IDs as well as their collection time for each pair. We report the lateral errors Δy in meters, and the yaw angle errors $\Delta\theta$ in degrees.

	2-28 09:42 - 06:44			4-17 12:29 - 21:58			11-4 20:44 - 12:29			16-17 21:48 - 21:58			17-23 21:58 - 05:34			28-35 06:44 - 11:26		
	Δy	$\Delta\theta$	inliers	Δy	$\Delta\theta$	inliers	Δy	$\Delta\theta$	inliers	Δy	$\Delta\theta$	inliers	Δy	$\Delta\theta$	inliers	Δy	$\Delta\theta$	inliers
SuperPoint [13]	0.18	0.78	60	0.14	1.2	36	0.22	1.53	72	0.11	0.61	41	0.20	0.89	51	0.16	1.31	72
D2-Net [11]	0.16	1.02	133	0.29	3.68	72	0.05	0.21	278	0.02	0.33	459	0.09	1.29	129	0.18	1.55	122
Sup. U-Net [10]	0.02	0.25	606	0.03	0.47	538	0.03	0.20	617	0.02	0.31	614	0.03	0.35	550	0.03	0.29	588
Self Sup. U-Net (Our)	0.03	0.34	579	0.06	0.50	536	0.03	0.19	619	0.02	0.32	611	0.04	0.38	539	0.04	0.27	582

B. Training Data Generation using Topological Localization

We generate direct and indirect image correspondences across different sequences in the training datasets as described in Section III, which are used to sample 100,000 training image pairs and 20,000 validation image pairs. To evaluate the accuracy of the sampled training data, we extract the relative transform between the source and target image from the spatio-temporal graph for all sampled image pairs, then plot the distribution of the absolute distances in Figure 6. As can be seen from the histogram, majority of the image pairs are within 3 metres of each other, which generally means sufficient overlapping visual field for training.

C. Training and Inference

We use the sampled image pairs to train the network. During training and inference, we discard outlier keypoints using RANSAC. The network is trained using the Adam optimizer with a learning rate of 10^{-5} on an NVIDIA Tesla V100 DGXS GPU. The encoder is a VGG16 network [35] pretrained on the ImageNet dataset [36], truncated after the *conv_5_3* layer. The network is finetuned end-to-end on the UTIAS Multiseason and In-the-Dark datasets for 50 epochs.

After training the network, we integrate the learned keypoint detector and descriptor with the VT&R system to extract features for visual localization. While the user manually drives the robot to teach a path, VT&R creates a spatio-temporal pose graph that stores relative poses between keyframes. The path is autonomously repeated by computing VO and localizing keyframes. VT&R relies on SURF for VO, and utilizes the learned features along with a sparse descriptor matcher for localization.

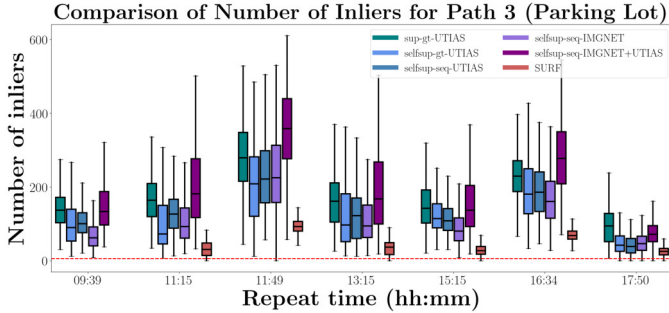
We conduct three experiments to verify the localization performance of the learned features. We compare our methods to other existing learned features under lighting changes by

calculating the path-following errors using the UTIAS-In-the-Dark test set.

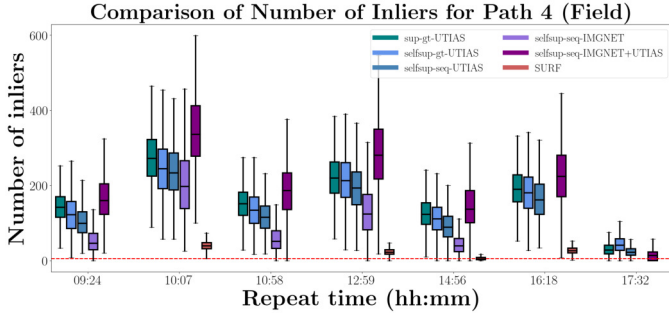
In addition, we perform an offline localization ablation study using pre-collected rosbags [10] for paths 3 and 4. Lastly, we perform closed-loop experiments on three paths (path 1, path 2, and path 3) under different lighting conditions as shown in Figure 1. We set the maximum speed of the robot to 0.6 m/s. Selected camera views for localization experiments are shown in Figure 9. For these two experiments, it was often not feasible to collect accurate RTK GPS ground truth as we drove in area where tree cover blocks the GPS signals. Hence, we report the median number of inliers in feature matching to reflect the performance of localization. We plot the median number of inliers against the path-following errors in both lateral and yaw directions using the UTIAS-In-the-Dark test set in Figure 8, which indicates that the median number of inliers is highly correlated with the path-following performance, showing that a higher inlier count is beneficial for accurate and robust localization.

D. Offline Localization: Learned Features Comparison

We compared the localization performance of our method to its supervised counterpart (Sup. U-Net) [10], SuperPoint [13] and D2-Net [11] in PyTorch. The pretrained networks provided by the authors are used to extract the keypoints, descriptors, and scores. We compute the path-following errors between six teach-repeat pairs sampled from the UTIAS-In-the-Dark test set, then report the lateral errors, yaw angle errors, and the median number of inliers in Table I, where the ground truth poses are obtained from Multi-Experience Localization during data collection. Our method is competitive with its supervised counterpart, and outperforms SuperPoint and D2-Net in all metrics for all six localization experiments.



(a) Plot of feature inliers for path 3 (parking lot).



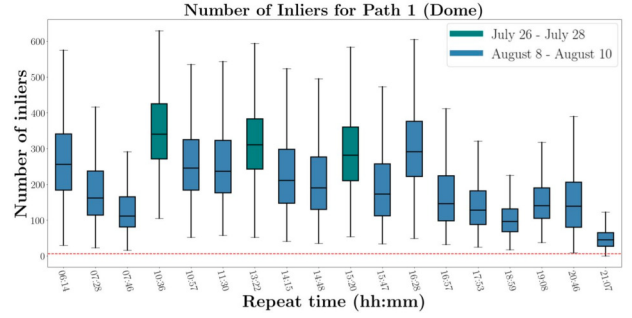
(b) Plot of feature inliers for path 4 (field).

Fig. 10: **The mean number of inliers for each repeat of the offline experiments** using pre-collected rosbag files for paths 3 (a) and path 4 (b), where ‘*sup/selfsup*’ indicates if the model is trained using supervised or self-supervised losses, ‘*gt/seq*’ indicates if the image pairs are generated using ground-truth poses or SeqSLAM-based algorithm, and ‘*IMGNET/UTIAS*’ represents the datasets that the model is pretrained or finetuned on. A missing entry indicates failure in localization. The learned features outperform SURF in all experiments. Under the exact same settings (i.e., trained on UTIAS datasets only), the self-supervised method achieves competitive results in comparison with the supervised method in all experiments. In addition, the performance increases for the self-supervised method when pretrained on ImageNet and finetuned on UTIAS datasets.

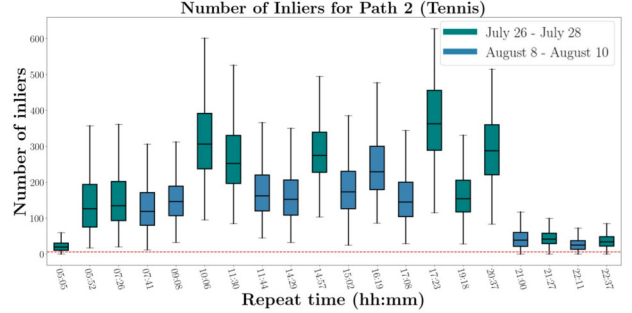
E. Offline Localization: Ablation Study

We perform offline experiments on paths 3 and 4 using pre-collected rosbag files, and present the results in Figure 10. Each path is played back 7 times from morning to nighttime with 100% localization success rate. In the same plot, we compare with different methods. In all experiments, SURF results in a significantly lower number of feature inliers, and fails to localize at the beginning of three different repeats. This shows that the learned features are superior in terms of handling illumination changes.

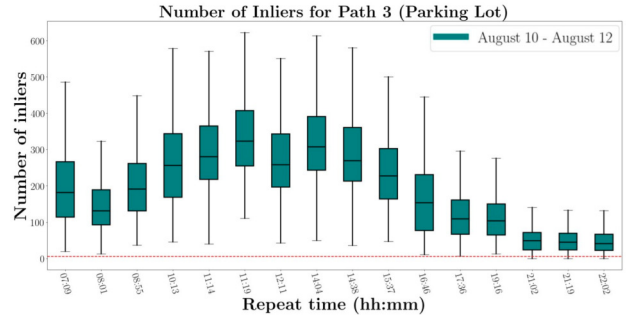
When trained on UTIAS datasets only (i.e., not pretrained on ImageNet), we see a slight performance drop in the self-supervised method when compared to the supervised counterpart. Although SeqSLAM matching is not perfect, using SeqSLAM matching to generate training pairs results in a competitive performance compared to using a ground-truth image matches. This suggests the effectiveness of using a place-



(a) Plot of feature inliers for path 1 (tennis court).



(b) Plot of feature inliers for path 2 (dome).



(c) Plot of feature inliers for path 3 (parking lot).

Fig. 11: **The mean number of inliers for each repeat of the closed loop experiments** for path 1 (a), path 2 (b), and path 3 (c). The red line indicates the minimum number of inliers required for successful localization. The learned features can be integrated with VT&R to successfully localize under different lighting conditions. In addition, the learned features can be generalized to new regions outside the training data.

recognition algorithm to find coarse image correspondences for detailed metric feature learning.

When trained on ImageNet only, the self-supervised method does not perform well compared to other methods, and failed to localize in two runs for path 4. However, the best performance for the self-supervised method is achieved when pretrained on ImageNet and finetuned on UTIAS datasets, which even outperforms the supervised method (i.e., not pretrained on ImageNet). It shows that by training on UTIAS datasets, it allows the model to generalize better to unseen data.

F. Online VT&R: Closed-Loop Experiments

We perform closed-loop experiments using online VT&R on path 1, 2, and 3 from Figure 7 with 100% success rate. Path 1 (dome) and path 2 (tennis court) is repeated 18 and 20 times respectively, from 6 am to 9 pm, and from July 26 to August 10 in 2022. Path 3 (parking lot) is repeated 17 times from 7 am to 10 pm in mid-August in 2022. The feature inliers for each repeat is shown in Figure 11. For every repeat, we obtained sufficient feature inliers for localization. In general, we get the highest number of inliers in the middle of the day or when the weather is cloudy, since there are no shadows on the ground. The number of inliers decreases at dusk or dawn, and is at the lowest during the night. In addition, we test on paths that are not within the training data to evaluate the generalizability of the network. Although part of path 1 and path 3 are outside of the training data, we still obtained sufficient number of feature inliers, which shows that the learned features can generalize well to unseen regions.

V. CONCLUSION

In this paper, we have shown that we can generate image correspondences for image-pair sampling and perform self-supervised feature learning without any ground-truth pose information. We validated the effectiveness of learned features in the VT&R framework on unseen paths under various lighting conditions. In addition, our pipeline can be readily deployed on self-collected image sequences without additional ground truth data. Limitations of the paper are the assumptions that we made to generate image correspondences. In the future, we intend to further relax these constraints and match sequences of image data that might not start or end at the same place. In addition, we plan to run closed-loop experiments over a longer period of time to test the robustness of the learned features against seasonal change.

REFERENCES

- [1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004. [Online]. Available: <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
- [2] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," vol. 3951, 07 2006, pp. 404–417.
- [3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," 11 2011, pp. 2564–2571.
- [4] P. Furgale and T. Barfoot, "Visual teach and repeat for long-range rover autonomy," *J. Field Robotics*, vol. 27, pp. 534–560, 09 2010.
- [5] M. Paton, K. MacTavish, M. Warren, and T. D. Barfoot, "Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat," in *2016 IEEE/RSJ IROS*, 2016, pp. 1918–1925.
- [6] V. Balntas, S. Li, and V. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in *The European Conference on Computer Vision (ECCV)*, September 2018.
- [7] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "Camnet: Coarse-to-fine retrieval for camera re-localization," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2871–2880.
- [8] Z. Laskar, I. Melekhov, S. Kalra, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," 2017.
- [9] A. Kendall, M. Grimes, and R. Cipolla, "Convolutional networks for real-time 6-dof camera relocalization," *CoRR*, vol. abs/1505.07427, 2015.
- [10] M. Gridseth and T. D. Barfoot, "Keeping an eye on things: Deep learned features for long-term visual localization," 2021.
- [11] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable CNN for joint detection and description of local features," *CoRR*, vol. abs/1905.03561, 2019.
- [12] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "TILDE: A temporally invariant learned detector," *CoRR*, vol. abs/1411.4568, 2014.
- [13] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," *CoRR*, vol. abs/1712.07629, 2017.
- [14] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: learned invariant feature transform," *CoRR*, vol. abs/1603.09114, 2016.
- [15] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: Learning local features from images," *CoRR*, vol. abs/1805.09662, 2018.
- [16] L. von Stumberg, P. Wenzel, Q. Khan, and D. Cremers, "Gn-net: The gauss-newton loss for deep direct SLAM," *CoRR*, vol. abs/1904.11932, 2019.
- [17] J. Revaud, P. Weinzaepfel, C. R. de Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2D2: repeatable and reliable detector and descriptor," *CoRR*, vol. abs/1906.06195, 2019.
- [18] L. Sun, M. Taher, C. Wild, C. Zhao, Y. Zhang, F. Majer, Z. Yan, T. Krajnfc, T. Prescott, and T. Duckett, "Robust and long-term monocular teach and repeat navigation using a single-experience map," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2635–2642.
- [19] M. Kasper, F. Nobre, C. Heckman, and N. Keivan, "Unsupervised metric relocalization using transform consistency loss," *CoRR*, vol. abs/2011.00608, 2020.
- [20] C. Valgren and A. J. Lilienthal, "Sift, surf & seasons: Appearance-based long-term localization in outdoor environments," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 149–156, 2010, selected papers from the 2007 European Conference on Mobile Robots (ECMR '07).
- [21] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *International Journal of Robotics Research*, vol. 32, pp. 1645–1661, 12 2013.
- [22] C. Linegar, W. Churchill, and P. Newman, "Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation," in *2015 IEEE ICRA*, 2015, pp. 90–97.
- [23] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixé, "Understanding the limitations of cnn-based absolute camera pose regression," *CoRR*, vol. abs/1903.07504, 2019.
- [24] D. Barnes and I. Posner, "Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar," *CoRR*, vol. abs/2001.10789, 2020. [Online]. Available: <https://arxiv.org/abs/2001.10789>
- [25] S. Lowry, N. Sünderhauf, P. Newman, J. Leonard, D. Cox, P. Corke, and M. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, pp. 1–19, 11 2015.
- [26] M. Cummins and P. Newman, "Highly scalable appearance-only slam - fab-map 2.0," 06 2009.
- [27] P. Neubert, N. Sünderhauf, and P. Protzel, "Appearance change prediction for long-term navigation across seasons," in *2013 European Conference on Mobile Robots*, 2013, pp. 198–203.
- [28] M. Milford, "Vision-based place recognition: How low can you go?" *The International Journal of Robotics Research*, vol. 32, pp. 766–789, 06 2013.
- [29] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE ICRA*, 2012, pp. 1643–1649.
- [30] P. Hansen and B. Browning, "Visual place recognition using hmm sequence matching," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4549–4555.
- [31] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, Jun. 2014. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/9057>
- [32] O. Vysotska and C. Stachniss, "Lazy data association for image sequences matching under substantial appearance changes," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 213–220, 2016.
- [33] —, "Effective visual place recognition using multi-sequence maps," *IEEE RAL*, vol. 4, no. 2, pp. 1730–1736, 2019.
- [34] P. Neubert, S. Schubert, and P. Protzel, "Exploiting intra database similarities for selection of place recognition candidates in changing environments," 2015.
- [35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.
- [36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.