

TRAIL: Trajectory-Aware Visual Place Recognition against Unordered Databases

Dominik A. Kloepfer^{1*}  and Patrick Wenzel² 

¹ Visual Geometry Group (VGG), University of Oxford

² Helsing

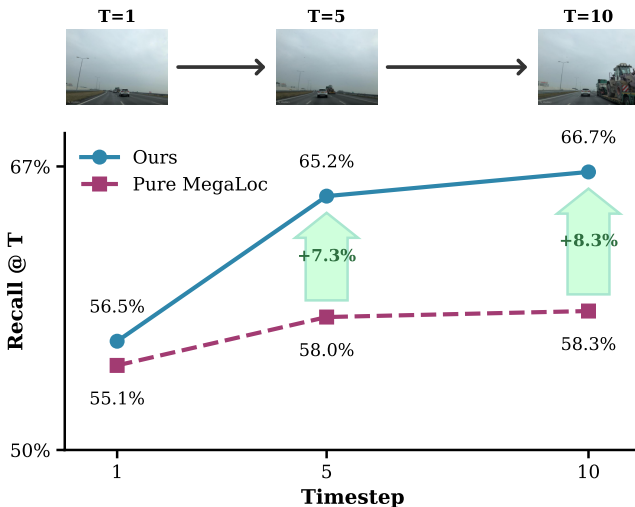


Fig. 1: Incorporating information from previous images in a sequence boosts Visual Place Recognition (VPR) performance in challenging settings. We compare the top-1 Recall of the state-of-the-art VPR algorithm MegaLoc [6] with our system, which makes use of previous images in the sequence (see Sec. 4 for full results).

Abstract. Modern Visual Place Recognition (VPR) methods excel on standard benchmarks yet remain brittle in feature-poor environments. By treating each query image in isolation, they discard the sequential context in any real trajectory. We formalize a task that exploits this context: given a query sequence, localize the *final* image against an *unordered* reference database—which, unlike sequence-to-sequence methods, requires no sequential structure in the database. We propose TRAIL (**TR**ajjectory-**A**ware **I**mage **L**ocalization), a principled framework based on Conditional Random Fields (CRF) that combines learned functions for visual similarity and for camera-motion consistency, refining a distribution over candidate references as each query arrives. A lightweight post-processing layer atop any pre-trained VPR backbone, TRAIL improves a state-of-the-art baseline by up to 8.3 percentage points on our primary benchmark, transfers to unseen datasets without retraining, and delivers its largest gains where visual cues are scarce.

* Work partially done during an internship at Helsing.

1 Introduction

Visual Place Recognition (VPR), the task of matching query images against a reference database to determine camera location, has seen remarkable progress in recent years. Leveraging large-scale datasets and powerful pre-trained visual encoders such as DINOv2 [28], modern methods achieve impressive performance on established benchmarks [2, 6, 17, 19].

Yet, this progress masks a fundamental brittleness. In visually challenging scenarios such as rural landscapes, forests, or uniform highway environments, performance degrades sharply due to a lack of distinctive visual cues. Recent work has further exposed fragility in seemingly robust systems, revealing sensitivity to minor variations in evaluation parameters like retrieval distance thresholds [16].

This brittleness is particularly troubling given VPR’s role in many important applications: camera relocalization, loop closure detection in SLAM, and hierarchical localization systems where VPR provides the initial coarse estimate.

We observe that standard VPR formulations exacerbate this brittleness through an artificial constraint: treating each query image in isolation. This single-image paradigm, while convenient for benchmarking, discards critical information available in any real navigation scenario. Cameras follow continuous trajectories through space, and each new observation arrives with strong geometric priors from preceding frames. Such information dramatically constrains the search space, but the standard VPR formulation is unable to leverage it.

Prior sequence-based VPR methods [7, 11, 21, 26, 42] operate in a *sequence-to-sequence* paradigm requiring the reference database to be organized as ordered sequences covering the same routes as potential queries. This imposes a significant structural constraint: it demands exhaustive coverage of all possible trajectories through the environment. In practice, however, reference imagery is typically collected opportunistically, from mapping vehicles, crowd-sourced platforms, or satellite data, yielding unordered sets of geo-tagged images rather than structured sequences.

In this paper, we formalize a task that addresses these limitations: given a sequence of query images, the goal is to localize the *final* query image against an *unordered* reference database. This setting captures the practical scenario of an agent that needs to determine its current position, exploiting the context from its recent trajectory without imposing any structural requirements on the database. We develop *TRAIL* (**TR**ajectory-**A**ware **I**mage **L**ocalization), a principled probabilistic framework for this task, leveraging Bayesian filtering to maintain and update a probability distribution over reference locations as new query images arrive. Crucially, this framework operates as a post-processing step over existing VPR methods, requiring no architectural modifications and minimal computational overhead.

Specifically, our contributions are as follows:

- (a) We formalize a Sequential Visual Place Recognition task that operates on unordered reference databases and targets localization of the final query image,

clearly distinguishing it from the sequence-to-sequence paradigm adopted by prior work (Sec. 3.1).

- (b) We introduce *TRAIL*, a probabilistic framework for incorporating sequential context into place recognition, designed to maximize retrieval accuracy while imposing minimal computational overhead (Sec. 3.2, Sec. 3.3, Sec. 3.4).
- (c) We empirically validate our proposed framework and demonstrate its effectiveness on challenging datasets (Sec. 4).

By exploiting the temporal structure of navigation sequences, our method achieves notable improvements in both robustness and accuracy. Its simplicity and compatibility with existing systems make it well-suited for integration into real-world localization pipelines.

2 Related Work

Visual Place Recognition. Visual Place Recognition (VPR) has evolved from handcrafted local features and global descriptors [4, 10, 18, 33] to deep-learning techniques that brought remarkable performance gains, with architectures such as NetVLAD [3], Generalized Mean Pooling [29], and MixVPR [1] marking key milestones. More recently, large pre-trained foundation models like DINOv2 [28] have further boosted performance by acting as local feature extractors, with methods such as AnyLoc [19], SALAD [17], and Bag-of-Queries [2] used to aggregate these features into global image descriptors. MegaLoc [6] is a recent model that builds on the SALAD architecture and achieves state-of-the-art performance by combining training data from multiple datasets.

These advances have led to the saturation of several previously challenging benchmarks. However, challenges persist; for example, Izquierdo and Civera [16] have illustrated the heavy reliance of recall performance on the distance threshold that determines correct retrievals.

Sequential Visual Place Recognition. Sequence-based localization originates from loop closure detection, where SeqSLAM [27] and its variants [5, 31] match query sequences against previously visited locations. This *sequence-to-sequence* paradigm has since been extended to general VPR; SeqNet [11] learns descriptors for sequence-based hierarchical retrieval, Mereu *et al.* [26] aggregate single-image descriptors into sequence-level representations, JIST [7] jointly trains image and sequence descriptors, Zhao *et al.* [42] apply spatio-temporal attention to construct sequence descriptors, and CaseVPR [21] performs correlation-aware alignment between query and database sequences. Closely related, Lynen *et al.* [24] match continuous query image streams against previously recorded trajectories, and SeqMatchNet [12] learns single-image descriptors that are explicitly shaped by a sequence-matching metric, collapsing to standard single-image retrieval when no ordered references are available. Despite CaseVPR framing its task as *sequence-to-frame*, its retrieval stage, too, operates over database sequences. All of these methods thus require the reference database to be organized as ordered sequences, which demands exhaustive route coverage of the environment.

The setting most closely related to ours is the *seq2im* task described by Warburg *et al.* [36], which pairs query sequences with an unordered reference database. However, their formulation considers a retrieval correct if it is near *any* image in the query sequence, rather than targeting a specific one. We formalize a stricter variant that reflects the practical localization objective: determining the position of the *final* query image, and develop *TRAIL*, a CRF-based framework, to address it.

Probabilistic Models for Sequence Localization. Some methods employ Hidden Markov Models (HMM) for sequence localization, such as the work by Rudić *et al.* [30] in indoor environments, and Gui *et al.* [14] who used LiDAR scans for unmanned ground vehicle localization in enclosed spaces. These approaches typically use discretized camera locations as hidden states, creating large state spaces that pose challenges in expansive environments. In contrast, we estimate directly the probability that database images are close to the true camera location, offering a novel approach to efficiently managing large environments.

Other approaches such as Cadena *et al.* [8] use a formulation based on Conditional Random Fields (CRF), which we, too, use in our theoretical treatment (see Sec. 3.2 and Appendix ??). However, existing works do not address the problem of visual place recognition with a very large database but instead focus on loop closure in SLAM pipelines, which have a significantly smaller set of reference images. Most directly related, Xu *et al.* [40] perform probabilistic topometric localization with a discrete Bayes filter whose forward recurrence coincides with the one we derive (Appendix ??). Their setting differs substantially, however, assuming an ordered reference map with per-edge odometry; transferring it to our unordered databases is non-trivial and forces simplifications that strip away much of its odometry-driven motion model, so that the resulting baseline fails to consistently improve over plain single-image retrieval (Appendix ??).

More broadly, sequence and trajectory priors are widely used in the robotics-localization literature for loop-closure detection and verification in SLAM, with recent examples such as ROVER [41] exploiting trajectory priors to reject false loop closures in repetitive environments. These methods, however, operate under far milder appearance variation than VPR (generally no day-night or season shifts) and assume the existence of *loops* that revisit previously mapped locations; they therefore do not transfer straightforwardly to our setting, in which queries follow open trajectories with no guaranteed revisits and visual appearance can change drastically between query and reference.

3 TRAIL: Sequential VPR with Conditional Random Fields

3.1 Task Definition

Similarly to the standard Visual Place Recognition task, our goal is to localize an image with respect to a database set of N *unordered* reference images with

associated locations, $\mathcal{R} = \{(r_i, x_i), i = 1 \dots N\}$, where $r_i \in \mathbb{R}^{3 \times H \times W}$ is an RGB-image of dimensions $H \times W$ and $x_i \in \mathbb{R}^3$ is a position coordinate.

In contrast to Visual Place Recognition, each query is not just a single query image, but a query *sequence* of length T , so that $Q = (q_1, \dots, q_i, \dots, q_T) \in \mathcal{Q}_T$ with $q_i \in \mathbb{R}^{3 \times H \times W}$ images typically taken in order by an agent traversing the environment and $\mathcal{Q}_T$ the set of all query sequences of length T . Each image q_i in the query sequence also has an associated location $y_i \in \mathbb{R}^3$, the agent’s location when collecting this query image.

Given such a query sequence, a Sequential-VPR algorithm retrieves a single reference image, so we can formally write $a : \mathcal{Q} \rightarrow \mathcal{R}$. A retrieval (r_i, x_i) is defined to be a *correct* retrieval if $|x_i - y_T| \leq \delta$ with $\delta > 0$. In other words, the algorithm attempts to localize the *final* query image in the query sequence by retrieving a reference image that was collected from a location within a certain threshold distance δ from the location of the query image.

This reflects the practical goal of localizing an agent *now*, *i.e.*, at the end of its trajectory. We assume that reference images carry known positions (GPS or geo-tags), which is standard in VPR [6, 36]. At inference time, reference positions are used for the distance cutoff Δ (Eq. (8)) and the aggregation function κ (Eq. (4)).

We measure performance using a *Recall@T* metric: the fraction of query sequences of length T for which the retrieved image is correct. The standard VPR *Recall@1* metric corresponds to the special case $T = 1$. Note that T denotes the sequence length, not the number of retrieved candidates as in the standard *Recall@K* metric; the two coincide only for $T=K=1$.

We now introduce *TRAIL*, which leverages a pre-trained VPR algorithm to compute the probability that each database image is a correct retrieval for the final query image in a sequence, returning the location of the highest-probability reference image.

3.2 Inference

Our goal is to localize the final query image in the sequence, q_T , as reliably as possible, maximizing the *Recall@T*. In other words, we want to return the reference image that has the highest probability of having been taken from a position within the threshold δ of the query image’s position.

We proceed in two stages: First, we model $P(c_t^i | (q_1, \dots, q_t))$ where c_t^i is a random variable with $c_t^i = 1$ if the reference image r_i is the retrieval image closest to query image q_t and $c_t^i = 0$ otherwise. Second, we use these probabilities to estimate $P(s_t^i | (q_1, \dots, q_t))$, where s_t^i is a random variable with $s_t^i = 1$ if the reference image r_i is within the distance threshold δ of the query image q_t and $s_t^i = 0$ otherwise.

In the appendix ??, we show that under some light locality assumptions we can model this process using a Conditional Random Field (CRF) [20] and derive the following recurrence relation for the first stage:

$$P(c_t^i | Q) = \frac{\alpha_T(i)}{\sum_j \alpha_T(j)}, \quad (1)$$

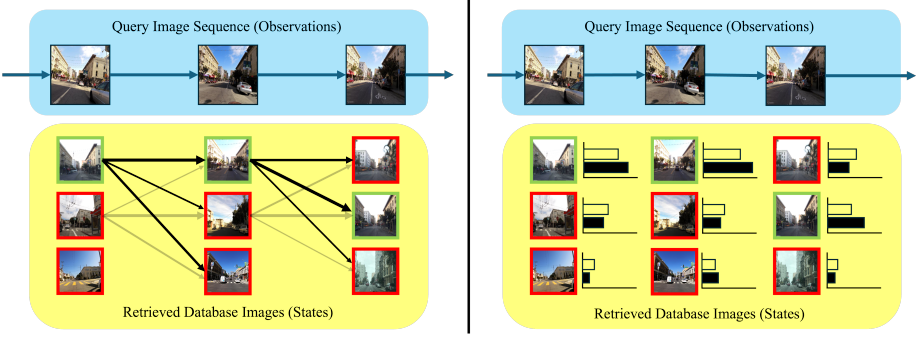


Fig. 2: Diagram illustrating *TRAIL*, our approach for Sequential Visual Place Recognition. For each query image q_t in the sequence (top row), a pre-trained VPR model retrieves the top- K database images, shown below in order of decreasing emission potential Ψ_e (green border = correct retrieval, red = incorrect). The emission potential measures visual similarity between a query and a candidate reference image, while the transition potential Ψ_{tr} (arrows, **left**) measures whether the spatial displacement between two candidate reference images is consistent with the observed camera motion between consecutive queries; thicker arrows indicate higher transition potentials. **Right:** Compared to using only emission potentials (top bars, not filled), fusing emission and transition potentials via the CRF recurrence (Eq. (2)) yields substantial accuracy gains (bottom bars, filled; see Sec. 4.4).

where

$$\begin{aligned}\alpha_1(i) &= \Psi_e(c_1^i, q_1) \\ \alpha_t(i) &= \sum_j \Psi_{tr}(c_t^i, c_{t-1}^j, q_t, q_{t-1}) \Psi_e(c_t^i, q_t) \alpha_{t-1}(j).\end{aligned}\tag{2}$$

$\Psi_e(c_t^i, q_t)$ and $\Psi_{tr}(c_t^i, c_{t-1}^j, q_t, q_{t-1})$ are two probability potentials. Intuitively, Ψ_e measures the probability that the state where $c_t^i = 1$ would ‘emit’ query image q_t , effectively measuring the similarity of the query image and a given retrieved image. Ψ_{tr} in turn measures the probability that the state where $c_{t-1}^j = 1$ transitions to the state where $c_t^i = 1$, conditioned on the query images at $t-1$ and t .

Both of these probability potentials can be parameterized using neural networks $\phi_e(r_i, q_t) \approx \Psi_e(c_t^i, q_t)$ and $\phi_{tr}(r_i, r_j, q_t, q_{t-1}) \approx \Psi_{tr}(c_t^i, c_{t-1}^j, q_t, q_{t-1})$.

In the second stage, we use the computed $P(c_t^i|Q)$ to compute $P(s_t^i|Q)$, the probability of reference image r_i being within the distance threshold of the query image q_t . We approximate

$$P(s_t^i|Q) = \sum_j P(s_t^i|Q, c_t^j) P(c_t^j|Q) \approx \sum_j P(s_t^i|c_t^j) P(c_t^j|Q).\tag{3}$$

This approximation allows us to model $P(s_t^i|c_t^j) = \kappa(|x_i - x_j|)$ using some appropriate function $\kappa: \mathbb{R}^+ \rightarrow [0, 1]$.

The complete forward algorithm is given in Algorithm 1. To obtain the predicted query image location at time step t , we return the location of the highest-probability reference image: $\hat{y}_t = x_{i^*}$ where $i^* = \arg \max_j P(s_t^j | Q)$.

Algorithm 1 Forward Algorithm for Sequential VPR

Require: ϕ_e, ϕ_{tr} # neural networks
Require: $\mathcal{R} = \{(r_i, x_i)\}$ # reference image database
Require: $Q = (q_1, \dots, q_T)$ # query sequence

- 1: $\alpha_1(i) \leftarrow \phi_e(r_i, q_1) \forall r_i \in \mathcal{R}$
- 2: **for** $t = 2 \dots T$ **do**
- 3: $\alpha_t(i) \leftarrow \sum_j \phi_{tr}(r_i, r_j, q_t, q_{t-1}) \phi_e(r_i, q_t) \alpha_{t-1}(j)$
- 4: **end for**
- 5: **for** $i : (r_i, x_i) \in \mathcal{R}$ **do**
- 6: $P(c_T^i | Q) \leftarrow \frac{\alpha_T(i)}{\sum_j \alpha_T(j)}$
- 7: $P(s_T^i | Q) \leftarrow \sum_j \kappa(|x_i - x_j|) P(c_T^j | Q)$
- 8: **end for**
- 9: **return** $P(s_T^i | Q), i = 1 \dots |\mathcal{R}|$

3.3 Network Architecture and Modeling Choices

In principle, we need to compute α_t for all images in the retrieval database. The large size of the database, however, makes this intractable, so we set $\Psi_e(c_t^i, q_t) = 0 \forall i \notin v_K(q_t)$ and only compute $P(c_t^i | Q)$ for the top- K reference images retrieved by a pre-trained VPR algorithm v_K .

Unless K is very large, there is a non-negligible probability that none of the top- K retrievals are actually within the distance threshold δ of a given query image. To account for this case, we add a dustbin state r_{dustbin} with $s_t^{\text{dustbin}} = 1$ iff $s_t^i = 0 \forall i \in v_K(q_t)$. We found improved performance by setting $\Psi_{tr}(c_t^i, c_{t-1}^{\text{dustbin}}, q_t, q_{t-1}) = \Psi_e(c_t^i, q_t)$, which corresponds to the case where the algorithm “lost track” of the query location.

Choice of κ . We require $\kappa(0) = 1$ and $\kappa(x) = 0$ for $x > \delta$, with smooth interpolation between these points. Intuitively, κ models the distribution of the minimum localization error $|x_i - y_t|$ when $c_t^i = 1$, *i.e.*, reference image r_i is the best possible reference image; if this error was always zero and the location of the best possible reference image always coincided with the query image location, the optimal κ would be $\kappa(x) = \mathbf{1}(x \leq \delta)$ with $\mathbf{1}$ as the indicator function.

We choose a scaled sigmoid:

$$\kappa(x) = \mathbf{1}(x \leq \delta) \cdot \sigma\left(\frac{\gamma - x}{\tau}\right) / \sigma\left(\frac{\gamma}{\tau}\right) \quad (4)$$

with the offset γ fixed and the bandwidth τ learnable. The scaling factor $\sigma(\frac{\gamma}{\tau})$ ensures that $\kappa(0) = 1$, while the indicator function $\mathbf{1}$ ensures that $\kappa(x) = 0 \forall x > \delta$. This function consists of an initial flat section followed by a quick decay towards zero. The boundary condition $\kappa(x)=0$ for $x>\delta$ directly mirrors the evaluation criterion (a retrieval is correct iff it lies within δ). Just as any retrieval method must define what counts as correct, κ encodes the same notion into the probability model. Our ablation (??) shows that replacing the scaled sigmoid with a supergaussian yields similar performance, confirming that the framework is robust to the particular functional form of κ .

Network Architectures. Finally, we also need to make some modeling choices for the network architectures of ϕ_e and ϕ_{tr} . Since we already use a pre-trained visual place recognition algorithm v_K to constrain the candidate reference images at each time step, it would be efficient to use intermediate results computed by that algorithm. In our experiments, we use the standard form of visual place recognition algorithms that compute global image descriptors $d_{\text{global}} \in \mathbb{R}^{d_g}$ by first extracting local image features $d_{\text{local}} \in \mathbb{R}^{h \times w \times d_l}$ that are then aggregated into a single vector. We make the following choices:

First, we notice that the task performed by ϕ_e is similar to pure Visual Place Recognition without sequential information, and so to the task that the global image descriptors are trained to do. We therefore set

$$\phi_e(r_i, q_t) = \text{MLP}(d_g(r_i) * d_g(q_t)) \quad (5)$$

with $*$ corresponding to element-wise multiplication and MLP denoting a small multi-layer perceptron, as a sort of ‘generalized cosine-similarity’.

To compute the probability potentials Ψ_{tr} , we first want to compute descriptors d_{tr} for the movement between the reference images r_j and r_i and between the query images q_{t-1} and q_t , and then assign a high probability potential if these transitions are similar. For the latter step, we again use a generalized cosine-similarity

$$\phi_{tr}(r_i, r_j, q_t, q_{t-1}) = \text{MLP}(d_{tr}(r_i, r_j) * d_{tr}(q_t, q_{t-1})). \quad (6)$$

We compute the transition descriptors d_{tr} by computing the cosine-similarity between each pair of local image features and arranging them into a tensor of shape $\mathbb{R}^{h \times w \times (h \cdot w)}$, which we pass through a convolutional neural network:

$$d_{tr}(a, b) = \text{CNN}(d_l(a)d_l(b)^\top). \quad (7)$$

Finally, we notice that the probability potential Ψ_{tr} should be zero for pairs of retrieval images that are very far apart, as a transition between those would be physically impossible. We therefore set $\phi_{tr} = 0$ when the positions x_i, x_j of the retrieval images r_i, r_j are too far apart. This leaves us with

$$\phi_{tr}(\dots) = \begin{cases} 0 & \text{if } |x_i - x_j| > \Delta \\ \text{MLP}(d_{tr}(r_i, r_j) * d_{tr}(q_t, q_{t-1})) & \text{otherwise.} \end{cases} \quad (8)$$

Intuition. Intuitively, the recurrence Eq. (2) suppresses candidate reference images whose locations are inconsistent with earlier observations: if an earlier query image was confidently matched to database image D , for the current query image the transition potentials will down-weight candidate images at locations relative to D 's location that do not match the query images' trajectory.

Even when no single image is easy to match (*e.g.*, along a feature-poor highway), the transition potentials implicitly compare the relative camera motion inferred from reference image pairs with the motion observed in the query sequence, progressively constraining the set of plausible locations.

3.4 Training

End-to-End Training. The entire algorithm Algorithm 1 is differentiable, so given a pre-trained algorithm v_K we train the entire system end-to-end by applying a binary cross-entropy loss on the computed $P(s_t^i|Q)$:

$$\begin{aligned} \mathcal{L}_t = \sum_i & -\log(P(s_t^i|Q)) \cdot \mathbf{1}(|x_i - y_t| \leq \delta) \\ & -\log(1 - P(s_t^i|Q)) \cdot \mathbf{1}(|x_i - y_t| > \delta). \end{aligned} \quad (9)$$

To improve training dynamics, we apply this loss at every timestep, not just T :

$$L = \sum_t \mathcal{L}_t. \quad (10)$$

In our implementation, we perform all calculations in log-space for improved numerical stability. The relevant derivations and equations can be found in the appendix in ??.

Pre-Training. End-to-end training is sped up and stabilized by first pre-training ϕ_e and ϕ_{tr} individually with cross-entropy losses.

For ϕ_e , given query images with K candidate retrievals from v_K , we train using:

$$L_e = - \sum_i \log \frac{\exp(\phi(r_i^c, q_i))}{\sum_k \exp(\phi(r_i^k, q_i))} \quad (11)$$

where $r_i^c = \arg \min_{r_i^k} |x_i^k - y_i|$ is the closest candidate to the query position.

For ϕ_{tr} , we sample consecutive query pairs (q_i^t, q_i^{t+1}) and, for efficiency, train transitions *from* the correct retrieval $r_i^{t,c}$ to K candidates $r_i^{t+1,k}$. That is, for each pair of query images we also select $r_i^{t,c}$, the candidate retrieval with the position closest to q_i^t , and K candidate retrievals $r_i^{t+1,k}$. We then compute $\phi_{tr}(r_i^{t,c}, r_i^{t+1,k}, q_i^t, q_i^{t+1})$ and again apply a cross-entropy loss:

$$L_{tr} = - \sum_i \log \frac{\exp(\phi_{tr}(r_i^{t,c}, r_i^{t+1,c}, q_i^t, q_i^{t+1}))}{\sum_k \exp(\phi_{tr}(r_i^{t,c}, r_i^{t+1,k}, q_i^t, q_i^{t+1}))}. \quad (12)$$

4 Experiments

4.1 Datasets

Mapillary Street-Level Sequences To train and evaluate TRAIL, we need datasets containing images with associated camera locations, organized into individual image sequences. A large and diverse dataset that fits these requirements is the *Mapillary Street-Level Sequences (MSLS)* [36] dataset, and accordingly we use it as our main training dataset. This dataset contains 1.6 million images collected from short sequences of dashcam footage in 30 cities around the globe, providing a large and diverse set of images for training and evaluation.

Since individual sequences are split across the original ‘query’ and ‘database’ sets for each city, we merge them and construct a new database set for each sequence, against which localization is performed. For each sequence, the database set consists of all images from the same city that do not appear in the query sequence, and all results for MSLS cited in this paper refer to this setting.

To avoid degenerate sequences with little camera movement, we skip frames in a sequence that have a distance of less than 25.0 meters from the previous frame.

Evaluation Protocol. Prior work [17, 19] holds out the images collected in San Francisco and Copenhagen for validation. However, we notice that these two cities are among the easiest in the entire dataset, with some of the highest *Recall@1* performance for state-of-the-art Visual Place Recognition algorithms such as MegaLoc [6]. We therefore instead hold out the images collected in Amsterdam and Boston for validation, which are more challenging. Whereas MegaLoc has a *Recall@1* performance of 89% for images from Copenhagen and 84.5% for images from San Francisco, the *Recall@1* performance for images from Amsterdam is 54.0% and 71.5% for images from Boston.

We continue to use the distance threshold of $\delta = 25.0$ meters used in the literature [6, 17, 19].

Two main factors account for the slightly worse *Recall@1* performance on the Mapillary-SLS dataset in our setting compared to the performance reported on the validation split of the dataset elsewhere in the literature even for images from Copenhagen and San Francisco (*e.g.*, Berton *et al.* [6] reach 91.0% *Recall@1*). Firstly, we ensure that no images from the same sequence as a query image are included in the database. Secondly, we use *all* other sequences to create the database instead of a more curated sub-set of images. While this leads to a higher theoretically possible localization performance due to a denser coverage of a given environment, it also increases the number of distractor images among which a VPR-algorithm must discriminate.

Nordland Another dataset containing image sequences that can be used as query sequences in our setting is the *Nordland* dataset [32]. This dataset comprises videos of a train traversing a 728-kilometer route in Norway across four seasons.

The largely rural landscape, with few distinctive features that remain consistent across seasons, makes this a challenging setting. While its lack of visual diversity makes it unsuitable for training, we evaluate our model on Nordland without fine-tuning, using weights trained on MSLS.

Evaluation Protocol. The standard test split of the Nordland dataset discards the sequential nature of the source data. We therefore define the query sequences by uniformly sub-sampling 256 sequences of 10 frames each from the winter-video.

As the database, we choose all the images of the summer-video, which is the season with the largest domain shift to the query season.

We filter out degenerate cases with little camera movement by skipping frames if they were not collected at least 10 meters from the previously sampled frame. Like for the MSLS dataset, we define $\delta = 25.0$ meters as the threshold for correct localization.

4Seasons As a third dataset, we evaluate on the *4Seasons* [38,39] dataset, which consists of gray-scale videos recorded using a sensor suite on a car following a number of different routes throughout different seasons. Most of these routes lead through urban environments with a large number of visual features, which causes non-sequential VPR algorithms to saturate them. We therefore use the “countryside” loop, which is the most challenging route that also supplies ground-truth camera poses for each frame for evaluation. Like Nordland, this dataset is used for evaluation only, again using weights trained on MSLS.

Evaluation Protocol. We follow a similar protocol as for the Nordland dataset, extracting 252 query sequences from the “winter” recording, which we localize against the “summer” recording that serves as the database. As before, we ensure that subsequent query frames are separated by at least 25.0 meters and that database images are separated by at least 10.0 meters, and use $\delta = 25.0$ meters as the threshold for correct localization. The comparatively small loop leaves a total of 672 reference images.

Other Datasets Few other datasets meet the requirement of sequential queries covering sufficiently large areas. Standard VPR benchmarks such as Pittsburgh-30k [3, 13], Tokyo 24/7 [34], and St. Lucia [37] do not preserve the sequential structure of their source data in the standard evaluation splits, making them unsuitable for our task without non-trivial re-processing. The Oxford Robot-Car dataset [25] contains sequential information but covers a small, feature-rich loop where im2im methods are near-saturated. Other domains such as off-road or aerial data are promising future directions where sequential priors may be especially valuable. We hope that clearly defining our task (Section 3.1) and framework (Section 3) will encourage future dataset collection and benchmark design in these domains.

4.2 Architecture and Training

We train on MSLS, holding out the Amsterdam and Boston images for evaluation and resize all images to squares with side-length 224 pixels.

As our pre-trained VPR model v_K we use MegaLoc [6], which computes 8448-dimensional global descriptors. The emission network ϕ_e is a small MLP operating on the element-wise product of global descriptors (Eq. (5)). The transition network ϕ_{tr} computes pairwise correlations between DINOv2 [28] local features (ViT-14-Base, 768-dim) and processes them with a ResNet-style CNN to obtain transition descriptors (Eq. (7)). We set the distance cutoff $\Delta = 75.0$ meters.

We keep MegaLoc frozen and only train ϕ_e and ϕ_{tr} , first individually and then end-to-end using AdamW [22] with RandAugment [9] data augmentation (excluding geometric transforms, as they interfere with the learning of geometric transitions between images).

For pre-training, we use $K = 10$ retrievals per query, with learning rate 10^{-3} and batch sizes of $B = 56$ (emission) and $B = 256$ (transition). For end-to-end training, we use batches of 6 sequences of length 10 with $K = 10$ retrievals and learning rate 10^{-4} .

All training is done on a single NVIDIA A100 GPU with 48GB memory. Full architecture details are provided in ??.

Computational and Memory Cost. TRAIL adds little overhead on top of the underlying VPR backbone. The DINOv2 local features used by ϕ_{tr} are produced by MegaLoc *en route* to its global descriptor, so ϕ_{tr} requires no additional backbone pass, only the small CNN and MLP heads. Taking TRAIL (MegaLoc) as the unit of inference wall-clock per query, raw MegaLoc and the hand-crafted heuristics cost $0.2\times$ and $0.6\times$, while the SelaVPR and PairVPR re-rankers cost $7.2\times$ and $9.3\times$ respectively. TRAIL is therefore substantially lighter than the re-rankers it matches or exceeds in accuracy (Sec. 4.4). On the memory side, the cached local features for a single image occupy roughly $1.3\times$ the storage of the raw reference image (5.3, 14.8, 14.6, and 0.5 GB for MSLS-Amsterdam, MSLS-Boston, Nordland, and 4Seasons, respectively). Since only the $2K = 20$ feature sets for the candidate retrievals of two consecutive frames are needed at once, these can be streamed from disk for arbitrarily large maps.

4.3 Baselines

We evaluate the performance of our approach against a number of baseline algorithms. We use the authors’ code and pre-trained weights.

Global Descriptor Algorithms. MegaLoc [6] and FoL [35] are two recent, state-of-the-art Visual Place Recognition algorithms that compute global image descriptors and use these to retrieve matches from a database. We apply them to our setting by ignoring the sequential information and simply localizing the final image in the sequence. Note that while MegaLoc uses a ViT-B-based feature

backbone, the authors of FoL only release weights for a ViT-L-based backbone. FoL therefore benefits from significantly larger model capacity, which should be kept in mind when interpreting its results.

Sequence-Retrieval Algorithms. We also compare to methods trained to use sequences for Visual Place Recognition: SeqNet [11], JIST [7], and CaseVPR [21]. These methods assume that the database contains sequences of images, whereas our database consists of *unordered* collections of images. We apply them by using the full query sequence descriptors to retrieve unordered database images, treating each as a single-image sequence. We adapt CaseVPR’s two stages as follows: we first retrieve the top- $K = 5$ database images using the full query sequence descriptor, then choose among them the image with the highest similarity to the final query image’s descriptor. Further details are given in ??.

Re-Ranking Algorithms. The third type of Visual Place Recognition Algorithm we evaluate are two recent re-ranking algorithms, SelaVPR [23] and PairVPR [15]. These im2im methods first retrieve $K = 10$ candidate database images using global image descriptors, before then re-ranking these images using more expensive methods. Like for the Global-Descriptor-based algorithms, we apply these algorithms to our setting by ignoring sequential information and only localizing the final image in the query sequence.

Hand-crafted Algorithms. Finally, we compare to three hand-crafted methods that use the MegaLoc algorithm for computing image descriptors for retrieval.

The first, *Maximum Similarity*, is briefly discussed in Warburg *et al.* [36] as the +MIN approach for their seq2im task. Given a window size $W = 2$, this algorithm computes the top-1 retrieved image for the final W query images and returns the reference image that was retrieved with the highest similarity.

The second, *Majority Voting*, also appears in Warburg *et al.*, under the name +MODE. Given a window size $W = 3$, this algorithm uses the top- $K = 10$ images retrieved for each of the final W query images and returns the reference image that was retrieved most often among these $W \cdot K$ images.

Finally, *Location Filtering* returns the database image that is most similar to the final image in the query sequence after filtering out any database image that was not collected within a distance threshold $d = 75.0$ meters of at least one of the top- $K = 10$ retrievals for the previous $W = 2$ images. This corresponds roughly to the maximum distance between two correctly retrieved images when their respective query images are separated by approximately 25 meters. Formal definitions for these hand-crafted heuristics can be found in ??.

4.4 Results

Our main results are given in Tab. 1. We report the *Recall@T* metric for $T = 1, 5, 10$ timesteps on the Amsterdam and Boston subsets of the Mapillary-SLS

Algorithm	MSLS			Nordland			4Seasons		
	$R@T=1$	$R@T=5$	$R@T=10$	$R@T=1$	$R@T=5$	$R@T=10$	$R@T=1$	$R@T=5$	$R@T=10$
<i>Global Descriptor</i>									
MegaLoc [6]	55.1	58.0	58.3	73.8	73.8	68.8	69.4	69.1	68.6
FoL (ViT-L) [35]	63.8	62.3	66.7	53.5	53.9	55.5	74.6	74.6	74.2
<i>Sequence-Retrieval</i>									
SeqNet [11]	17.4	5.8	8.3	3.9	2.7	2.3	23.8	18.3	17.7
JIST [7]	17.4	11.6	16.7	10.9	1.6	0.4	34.5	15.1	6.5
CaseVPR [21]	33.3	27.5	25.0	18.4	7.0	7.8	58.7	21.8	22.2
<i>Re-Ranking</i>									
SelaVPR [23]	71.0	65.2	66.7	64.8	64.1	60.9	<u>77.0</u>	76.6	76.2
PairVPR [15]	59.4	52.2	50.0	69.5	69.5	66.0	92.1	91.7	91.5
<i>Hand-Crafted</i>									
Max. Similarity	55.1	39.1	66.7	73.8	43.0	39.1	69.4	66.3	65.7
Majority Voting	55.1	37.7	50.0	73.8	19.5	15.2	69.4	46.0	46.0
Loc. Filtering	55.1	60.9	58.3	73.8	<u>79.7</u>	<u>76.2</u>	69.4	69.4	66.9
<i>Ours</i>									
TRAIL (MegaLoc)	56.5	65.2	66.7	75.3	81.3	78.1	71.0	<u>78.2</u>	<u>79.0</u>

Table 1: $Recall@T$ for $T = 1, 5, 10$ on the Amsterdam and Boston subsets of Mapillary SLS [36], on Nordland [32], and on the 4Seasons [38, 39] countryside loop (Nordland and 4Seasons: winter queries localized against a summer database). $T=1$ corresponds to standard single-image VPR. Best results in **bold**, second-best underlined; FoL uses a larger ViT-L backbone than MegaLoc (ViT-B).

dataset, using all sequences that contain at least five images with successive query images separated by at least 25 meters, on the Nordland dataset, localizing query-sequences drawn from the winter recording against the images drawn from the summer recording, and on the 4Seasons countryside loop under the analogous winter-to-summer protocol.

Whereas the improvements over vanilla MegaLoc are negligible for the first image in a sequence (when no sequential information is available), they are substantial for later frames. On MSLS at $T=10$, SelaVPR reaches 66.7%, tying with our method. However, SelaVPR applies per-frame re-ranking using cross-attention over local features, which is considerably more expensive than our small MLPs operating on $K=10$ pre-retrieved candidates. FoL, despite using a larger ViT-L backbone, does not outperform our ViT-B-based method at $T \geq 5$ on MSLS and falls clearly behind on Nordland.

Sequence-retrieval methods that assume a database consisting of *sequences* perform very poorly in our setting. The core issue is that these methods treat all query images equally rather than focusing on localizing the final image: their sequence descriptors average across all frames, so when database “sequences” consist of a single image the descriptor distributions are mismatched. TRAIL avoids this by explicitly modeling transitions and returning the location estimate for the final timestep. A more detailed analysis is given in ??.

For $T=1$, the handcrafted heuristics reduce to pure MegaLoc. Among them, only Location Filtering consistently improves over vanilla retrieval. This heuristic performs well on Nordland’s predictable train trajectory but is less effective on

the more varied driving data in MSLS, where learned transition modeling is more robust.

On 4Seasons, the same patterns hold: our method again improves substantially over the raw MegaLoc backbone as well as over the hand-crafted heuristics and SelaVPR, remaining effective despite the compounded domain shift from snowy winter queries to a summer database and from colour training data to grayscale test images, and despite never being trained on grayscale imagery. The notable exception is PairVPR, which performs exceptionally well on this dataset. We attribute this to the comparatively small reference set of the countryside loop, which leaves far fewer distractor images, combined with PairVPR being more robust to the season and colour domain shift than SelaVPR.

In ?? we report the performance of these baselines using different values for their hyper-parameters.

For an ablation analysis investigating the impact of some of the individual design choices for our method, refer to ?? in the supplement.

5 Conclusion

In this work, we presented *TRAIL*, a novel framework for Sequential Visual Place Recognition that exploits temporal context across a sequence of query images. Specifically, TRAIL uses localization estimates from earlier images in the sequence to improve place recognition for the current query image within a large reference database. This approach was empirically shown to improve the *Recall@T* metric by up to 8.3 percentage points on MSLS, 9.3 on Nordland, and 10.4 on 4Seasons, with minimal computational overhead.

Several directions for future work remain: (i) extending to domains beyond street-level driving, such as aerial or off-road imagery; (ii) exploring richer architectures for ϕ_e and ϕ_{tr} , including fine-tuning the backbone v_K itself; (iii) directly modeling the evolving distribution of the query image’s position rather than classifying the closest reference image, which could yield smaller localization errors; and (iv) further reducing the framework’s compute and memory footprint, for instance through a learned down-projection of the redundant DINOv2 local features, or by exploiting odometry data to drive lighter-weight transition potentials.

Acknowledgements

Dominik A. Kloepfer gratefully acknowledges support from the EPSRC (VisualAI, EP/T028572/1).

References

1. Ali-Bey, A., Chaib-Draa, B., Giguere, P.: Mixvpr: Feature mixing for visual place recognition. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2998–3007 (2023)

2. Ali-Bey, A., Chaib-draa, B., Giguère, P.: Boq: A place is worth a bag of learnable queries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17794–17803 (2024)
3. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5297–5307 (2016)
4. Arandjelovic, R., Zisserman, A.: All about vlad. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1578–1585 (2013)
5. Bai, D., Wang, C., Zhang, B., Yi, X., Yang, X.: Sequence searching with cnn features for robust and fast visual place recognition. *Computers & Graphics* **70**, 270–280 (2018)
6. Berton, G., Masone, C.: Megaloc: One retrieval to place them all. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2861–2867 (June 2025)
7. Berton, G., Trivigno, G., Caputo, B., Masone, C.: Jist: Joint image and sequence training for sequential visual place recognition. *IEEE Robotics and Automation Letters* **9**(2), 1310–1317 (2023)
8. Cadena, C., Gálvez-López, D., Tardós, J.D., Neira, J.: Robust place recognition with stereo sequences. *IEEE Transactions on Robotics* **28**(4), 871–885 (2012)
9. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 702–703 (2020)
10. Cummins, M., Newman, P.: Fab-map: Probabilistic localization and mapping in the space of appearance. *The International journal of robotics research* **27**(6), 647–665 (2008)
11. Garg, S., Milford, M.: Seqnet: Learning descriptors for sequence-based hierarchical place recognition. *IEEE Robotics and Automation Letters* **6**(3), 4305–4312 (2021)
12. Garg, S., Vankadari, M., Milford, M.: SeqMatchNet: Contrastive learning with sequence matching for place recognition & relocalization. In: Proceedings of the 5th Conference on Robot Learning. pp. 429–443 (2022)
13. Gronat, P., Obozinski, G., Sivic, J., Pajdla, T.: Learning and calibrating per-location classifiers for visual place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 907–914 (2013)
14. Gui, L., Zeng, C., Luo, J., Yang, X.: Place recognition through multiple lidar scans based on the hidden markov model. *Sensors* **24**(11) (2024). <https://doi.org/10.3390/s24113611>, <https://www.mdpi.com/1424-8220/24/11/3611>
15. Hausler, S., Moghadam, P.: Pair-vpr: Place-aware pre-training and contrastive pair classification for visual place recognition with vision transformers. *IEEE Robotics and Automation Letters* **10**(4), 4013–4020 (2025). <https://doi.org/10.1109/LRA.2025.3546512>
16. Izquierdo, S., Civera, J.: Close, but not there: Boosting geographic distance sensitivity in visual place recognition. In: European Conference on Computer Vision. pp. 240–257. Springer (2024)
17. Izquierdo, S., Civera, J.: Optimal transport aggregation for visual place recognition. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 17658–17668 (2024)
18. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 3304–3311. IEEE (2010)

19. Keetha, N., Mishra, A., Karhade, J., Jatavallabhula, K.M., Scherer, S., Krishna, M., Garg, S.: Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters* **9**(2), 1286–1293 (2023)
20. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning* (2001)
21. Li, H., Peng, G., Zhang, J., Wen, M., Ma, Y., Wang, D.: Casevpr: Correlation-aware sequential embedding for sequence-to-frame visual place recognition. *IEEE Robotics and Automation Letters* (2025)
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: *International Conference on Learning Representations* (2019), <https://openreview.net/forum?id=Bkg6RiCqY7>
23. Lu, F., Zhang, L., Lan, X., Dong, S., Wang, Y., Yuan, C.: Towards seamless adaptation of pre-trained models for visual place recognition. In: *The Twelfth International Conference on Learning Representations* (2024)
24. Lynen, S., Bosse, M., Furgale, P., Siegwart, R.: Placeless place-recognition. In: *2014 2nd International Conference on 3D Vision*. vol. 1, pp. 303–310 (2014)
25. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)* **36**(1), 3–15 (2017). <https://doi.org/10.1177/0278364916679498>, <http://dx.doi.org/10.1177/0278364916679498>
26. Mereu, R., Trivigno, G., Berton, G., Masone, C., Caputo, B.: Learning sequential descriptors for sequence-based visual place recognition. *IEEE Robotics and Automation Letters* **7**(4), 10383–10390 (2022)
27. Milford, M.J., Wyeth, G.F.: Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In: *2012 IEEE international conference on robotics and automation*. pp. 1643–1649. IEEE (2012)
28. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023)
29. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence* **41**(7), 1655–1668 (2018)
30. Rudić, B., Pichler-Scheder, M., Schmidt, R., Helm, C., Efrosinin, D., Kastl, C., Auer, W.: A geometry-aware hidden markov model for indoor positioning. In: *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. pp. 547–552. IEEE (2020)
31. Siam, S.M., Zhang, H.: Fast-seqslam: A fast appearance based place recognition algorithm. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 5702–5708. IEEE (2017)
32. Sünderhauf, N., Neubert, P., Protzel, P.: Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In: *Proc. of workshop on long-term autonomy, IEEE international conference on robotics and automation (ICRA)*. p. 2013. Citeseer (2013)
33. Sünderhauf, N., Protzel, P.: Brief-gist-closing the loop by simple means. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 1234–1241. IEEE (2011)
34. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1808–1817 (2015)

35. Wang, C., Chen, S., Song, Y., Xu, R., Zhang, Z., Zhang, J., Yang, H., Zhang, Y., Fu, K., Du, S., et al.: Focus on local: Finding reliable discriminative regions for visual place recognition. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. pp. 7536–7544 (2025)
36. Warburg, F., Hauberg, S., López-Antequera, M., Gargallo, P., Kuang, Y., Civera, J.: Mapillary street-level sequences: A dataset for lifelong place recognition. In: *Computer Vision and Pattern Recognition (CVPR)* (June 2020)
37. Warren, M., McKinnon, D., He, H., Upcroft, B.: Unaided stereo vision based pose estimation. In: *Proceedings of the 2010 Australasian Conference on Robotics and Automation*. pp. 1–8. Australian Robotics & Automation Association (2010)
38. Wenzel, P., Wang, R., Yang, N., Cheng, Q., Khan, Q., von Stumberg, L., Zeller, N., Cremers, D.: 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In: *German Conference on Pattern Recognition (GCPR)* (2020)
39. Wenzel, P., Yang, N., Wang, R., Zeller, N., Cremers, D.: 4Seasons: Benchmarking visual SLAM and long-term localization for autonomous driving in challenging conditions. *International Journal of Computer Vision (IJCV)* (2024)
40. Xu, M., Fischer, T., Sünderhauf, N., Milford, M.: Probabilistic appearance-invariant topometric localization with new place awareness. *IEEE Robotics and Automation Letters* **6**(4), 6985–6992 (2021)
41. Yu, J., Yang, J., Hu, A., Wang, J., Tan, P., Zhang, H.: ROVER: Robust loop closure verification with trajectory prior in repetitive environments. *IEEE Transactions on Automation Science and Engineering* **23**, 8476–8489 (2026)
42. Zhao, J., Zhang, F., Cai, Y., Tian, G., Mu, W., Ye, C., Feng, T.: Learning sequence descriptor based on spatio-temporal attention for visual place recognition. *IEEE Robotics and Automation Letters* **9**(3), 2351–2358 (2024)