Efficient and Flexible Spatio-Temporal Attention for Sequential Visual Place Recognition

Idan(Yu Kiu) Lau
New York University
idanlau@nyu.edu

***Abstract*—Sequential visual place recognition (seq-VPR) provides better robustness in poor lighting conditions compared to single-framed VPR by leveraging spatio-temporal features across multiple frames. While traditional attention mechanisms enhance seq-VPR's spatio-temporal modeling capabilty, they incur significant computational costs when processing long frame sequences. Moreover, existing spatio-temporal attention-based seq-VPR methods require the same input sequence length for both training and inference, limiting their practical flexibility. To overcome these limitations, we propose Adapt-STformer, which features two key innovations: a spatial deformable attention module that reduces computations, coupled with an iterative recurrent temporal mechanism that enables processing of arbitrary-length sequences. In challenging experiments, Adapt-STformer achieves 39% faster sequence extraction, 35% reduction in memory usage, and up to 15% improvements on recall relative to STformer, our main method of comparison. Adapt-STformer, therefore, balances well between computational efficiency, flexibility, and recall. The code is available at https://github.com/xx/xxxx**

***Index Terms*—Recognition, Localization, SLAM, Visual Place Recognition**

I. Introduction

Visual place recognition (VPR) aims to retrieve frames from a geotagged database that are located at the same place as the queried frame using only visual features, thus bypassing the need for GPS [1]. Single-framed VPR is vulnerable to drastic changes in noise and appearance; therefore, studies have investigated the utilization of sequential frames to address this issue (seq-VPR). Combined with advances in deep learning, particularly the widespread use of attention layers [2] in computer vision, has motivated us to explore their potential to better model spatio-temporal relationships in seq-VPR [3]–[5]

However, we identify two core challenges hindering existing attention-based spatio-temporal seq-VPR methods [5] are their high compute time and memory costs [6], and the enforcement of identical sequence lengths at training and inference [7], even though the optimal sequence length can vary with environment and hardware constraints.

To address these limitations, we introduce Adapt-STformer, a seq-VPR architecture that unifies two key ideas. The first is a spatial deformable attention module: by learning a small set of adaptive sampling offsets over a 2D query grid, it focuses computation on the most informative regions of each frame, drastically reducing overhead compared to standard attention. The use of deformable attention also adaptively samples discriminative local features while maintaining global context awareness. This preserves small but critical structural cues, particularly vital under poor lighting conditions. The second innovation is an iterative recurrent temporal mechanism: at each time step t, we feed the deformable-attention output from frame $t-1$ into the deformable-attention module for frame t as its query. This recurrent connection propagates information forward through time, removes the need for a dedicated temporal encoder, and lets the model handle arbitrary sequence lengths.

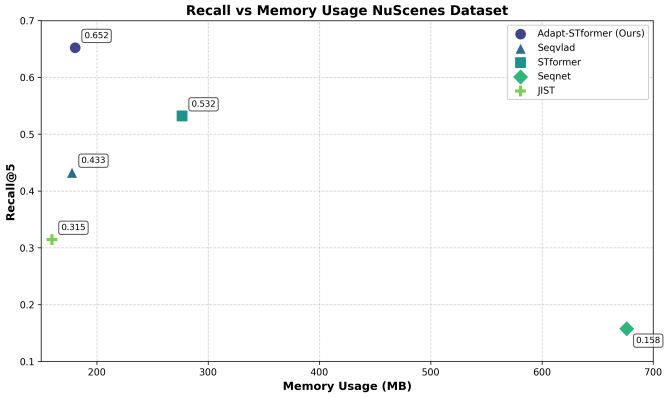


Fig. 1: Adapt-STformer showing a good balance between recall performance and memory usage

The contributions of this paper are threefold:

- A spatial deformable attention module that reduces computational costs while preserving features important under poor lighting conditions
- A recurrent temporal mechanism that effectively propagates information through time while allowing our method to be sequence length agnostic
- Improved recall under poor lighting conditions while utilizing less compute time and resource at inference compared to similar method

II. Related Works

A. Efficient and Flexible Sequential Descriptors for VPR

Early methods exploited temporal continuity by matching individual frame descriptors across time (e.g., SeqSLAM) [8], but these suffered from high computational cost and sensitivity to motion assumptions. To overcome this, SeqNet [9] introduces a lightweight 1D temporal convolutional network that encodes a short image sequence into a single descriptor

for coarse-to-fine retrieval. Building on this idea, SeqVLAD [7] adapts NetVLAD [10] to sequences by clustering and aggregating frame-level features into a fixed-length vector, making the descriptor agnostic to sequence length without retraining. Other length-agnostic methods, such as JIST [11] similarly fuse or pool local features over variable spans to improve robustness under differing traversal speeds.

B. Spatio Temporal VPR

The first fully attention based spatio-temporal seq-VPR approach, STformer [5], leverages spatial attention within individual frames to capture local feature distributions and temporal attention across frames to model dynamic interactions over time. By employing a sliding window mechanism and relative positional encoding, the temporal attention module tracks features across sequences. The resulting patch embeddings from both spatial and temporal branches are combined and aggregated via NetVLAD [10], producing a discriminative sequence descriptor. However, STformer is specifically set to handle sequence lengths of 5, making it not sequence length agnostic, and its use of dense attention for spatial features is computationally expensive. STformer is the main baseline we compare our method to.

C. Image BEV VPR

BEV2PR (BEV-Enhanced image VPR) [12] generates explicit BEV structural cues from a single monocular camera by "lifting" RGB features into a pseudo-point cloud and pooling into a BEV map; these BEV features are fused with visual embeddings through dual CNN streams and late-fusion aggregation to improve place recognition under extreme appearance changes (e.g., day vs. night), achieving substantial gains on hard subsets. We draw inspiration from this work, which suggests that focusing on structural cues is beneficial in poor lighting conditions. We utilize the NuScenes dataset [13] in our experiments for seq-VPR, as described in their work.

D. Deformable Attention

Deformable attention, introduced in Deformable DETR [14], addresses critical limitations of standard Transformer attention in vision tasks—specifically, the slow convergence and quadratic complexity when processing high-resolution image features. This mechanism replaces global attention with adaptive sparse sampling: for each query element (e.g., an object query or feature pixel), it predicts a small set of K sampling offsets (Δp) and attention weights (A) via a lightweight linear projection. We apply this attention module to learn spatial features from our input images.

E. BEV and transformers

BEVformer [15] creates Bird's-Eye-View representations from Multi-Camera Images via Spatiotemporal Transformers. It employs predefined grid-shaped BEV queries that look up spatial features via deformable attention over multi-camera views and fuse history BEV features via temporal self-attention. We take heavy inspiration from BEVformer's recurrent fusion of past BEV features and its usage of [14] for attending to spatial features.

[16] demonstrates the effectiveness of deformable attention in generating BEV features for multi-camera VPR, reinforcing its potential in this domain. However, their method does not leverage temporal information, limiting its applicability to seq-VPR tasks. Additionally, both [15] [16] rely on synchronized multi-camera inputs at each timestep to construct a 360° view and requires camera intrinsics for 3D feature projection . Our work addresses a more constrained yet practical scenario: monocular front-view-only input and unknown camera intrinsics.

F. Efficient Backbone Architectures

The choice of image encoder strongly influences descriptor quality. CCT384 [17]combines convolutional tokenization with transformer layers, preserving inductive bias. CCT384 is also less computationally intensive compared to traditional ViT models [18]. CCT384 was the backbone of choice for both seqvlad [7] and STformer [5].

METHODOLOGY

A. Overall Architecture

The model takes a frame sequence $\mathbf{S} \in \mathbb{R}^{L\times H\times W\times 3}$ as input, composed of L image frames with dimensions $H \times W$. These are first fed into the CCT384 [17] image encoder, which outputs features of shape $\mathbf{L} \times \mathbf{N} \times \mathbf{D}$, where N is the number of tokens and D is the feature dimension. The features then pass through our **spatio-temporal module** (comprising of spatial deformable attention and recurrent temporal mechanism) [15], producing outputs of the same shape $\mathbf{L} \times \mathbf{N} \times \mathbf{D}$. Finally, the aggregation module processes these features and outputs a sequential descriptor of shape $\mathbf{C} \times \mathbf{D}$ consistent with seqvlad and stformer, where $\mathbf{C}$ is the predefined number of VLAD [10] clusters.

Spatial Deformable Attention: Inspired by [14], we utilize their spatial deformable attention layer to learn spatial features. Deformable attention [14] [19] is a resource-efficient mechanism where each query focuses on localized regions of interest, contrasting with standard attention, which samples only $\mathbf{K}$ key points near reference points to compute attention, thereby reducing computational cost.

$$\text{SpatialDeformAttn}(Q, V) = \sum_{h,l,p} A_{h,l,p} \cdot W'_h x(p + \Delta p_{h,l,p})$$

Here, Q represents queries encoding target spatial positions, V denotes image features from a given image encoder, and $\Delta p_{h,l,p}$ are learnable sampling offsets. The attention weights $A_{h,l,p}$ and head-specific transformations W'_h enable efficient aggregation of features across h (attention heads), l (multi-scale feature levels), and p (sampled points).

We derive the reference points from the backbone's output token size. For example, when inputting an image size of 384

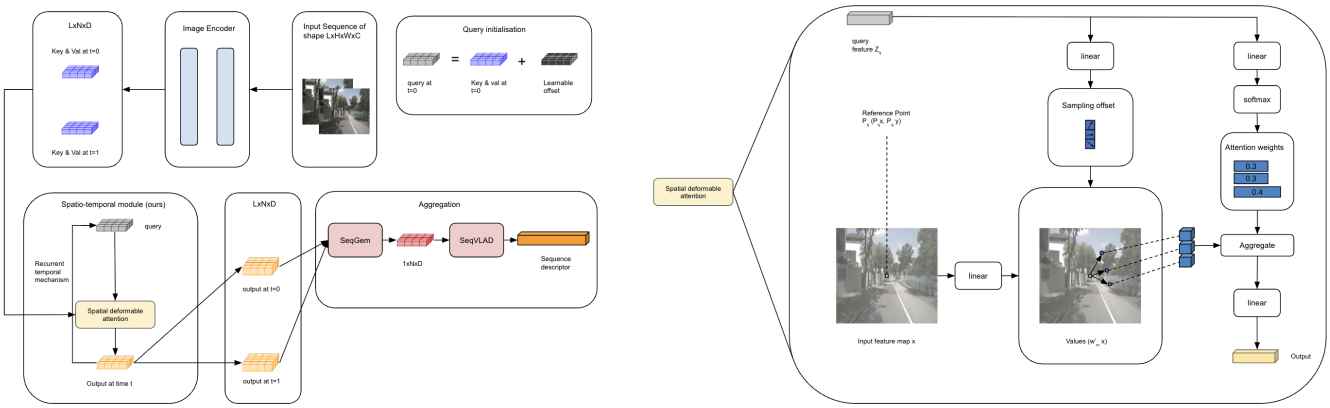


Fig. 2: **Proposed architecture of Adapt-STformer, example setup where sequence length is 2:** An input image sequence of length L (shape H×W×C per frame) is processed by an image encoder to generate token features of shape L×N×D, where N is the token count and D the feature dimension. These features serve as keys and values for the spatio-temporal module. The module operates recurrently: At t=0, the query is initialized and processes key/value from frame 0. For subsequent timesteps (t > 0), the query is dynamically updated using the previous output via a recurrent temporal mechanism, while keys/values correspond to frame t. In total the recurrent temporal mechanism runs t-1 times. Outputs from all L timesteps are aggregated through SeqGem and SeqVLAD to generate a compact 1×N×D sequence descriptor. We also show the inner working of the spatial deformable attention module, where we show a simplified set up of only 3 sampling offsets and 1 attention head, we take figure inspiration from [14]

x 384, the CCT384 encoder outputs 576 tokens, which we interpret as a 24×24 spatial grid. These grid dimensions (height H and width W) are used to calculate reference points using the formula:

$$\mathbf{x}' = \left(x - \frac{W}{2}\right) \times s\,, \quad \mathbf{y}' = \left(y - \frac{H}{2}\right) \times s\,,$$

Recurrent Temporal Mechanism: The recurrent temporal mechanism in our model is also inspired by BEVFormer [15] but modified to operate directly within the input sequence. Given a sequence of frames labeled $t = 0, 1, \ldots, T$, at each timestep t, the query Q_t is the output feature from the spatial deformable attention in previous timestep $t-1$ [20], while the value V_t is derived from the current frame's encoder features. This is formulated as:

$$\text{Output}_t = \text{SpatialDeformAttn}(Q = \text{Output}_{t-1}, V_t)$$

By reusing Q_{t-1} as the query, the model propagates spatio-temporal information across frames. This iterative approach eliminates the need for a separate temporal attention layer, unlike [5], and also ensures sequence-length agnosticism. For example, at $t = 2$, the query Q_1 (from $t = 1$) interacts with K_2 and V_2 (from $t = 2$). The recurrent mechanism iterates $L - 1$ times in total, and the query at t=0 is initialized as the first image frame's feature plus a learnable offset.

Aggregation Module : We save the output from each step of the recurrence to form a input of shape $\mathbf{L} \times \mathbf{N} \times \mathbf{D}$ into SeqGeM [11], permuted to match its expected format. After permutation:

- N becomes the non-batch dimension
- D represents the feature dimension

This treats each token as an independent element for processing. SeqGeM is a learned mean pooling layer from JIST [11] that operates along the temporal axis for a sequence of single-image embeddings. Formally:

$$\text{SeqGeM}(d) = \left(\frac{1}{L}\sum_{i=1}^{L} d_i^p\right)^{1/p}$$

Where p is a learnable parameter, d_i is the descriptor of the i-th frame.

SeqGem has key properties of acting as a learnable mean pooling layer, collapsing the sequence length dimension output from the spatio-temporal module, and further enhancing temporal learning across frames. The SeqGeM output (1 × N × D) feeds into SeqVLAD [7], which treats it as N set of D-dimensional embeddings. SeqVLAD aggregates embeddings via soft-assigned residuals:

$$\text{SeqVLAD}(k) = \sum_{f=1}^{L=1}\sum_{i=1}^{N} a_k(x_{fi}) \cdot (x_{fi} - c_k)$$

where x_i is a single embedding, c_k is the k-th centroid which is a trainable parameter, and $a_k(x_i)$ is a soft-assignment defined as:

$$a_k(x_i) = \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}}$$

Where w_k, b_k are also trainable parameters. Both SeqGeM and SeqVLAD are sequence-length-agnostic, ensuring our method remains flexible to variable input sizes.

B. Loss Function

We utilise the same triplet loss in seqnet [9] and in most baseline methods.

EXPERIMENTS

TABLE I: Datasets detail. This table specifies the number of images in the dataset used.

dataset		database / queries
NordLand	train set	15000 / 15000
	test set	3000 / 3000
Oxford-easy	train set	3619 / 3962
	test set	3632 / 3921
Oxford-hard	train set	2322 / 2585
	test set	2970 / 2920
Nuscenes	train set	- / -
	test set	4500 / 4000

A. Dataset

In our experiments, we use three datasets: NordLand, Oxford RobotCar, and Nuscenes. The number of images in train/test is listed in Table I, while the specific filenames of the images used are detailed in the struct files of the codebase.

1) NordLand: The Nordland dataset [21] comprises a collection of images captured during rail journeys across four seasons, covering various weather and lighting conditions. We use the Summer-Winter pair for training, and the Spring-Fall pair for testing.

2) Oxford RobotCar: The Oxford RobotCar [22] [23] dataset is a large-scale dataset for autonomous driving research. It encompasses road scenes captured during different periods. We design two experimental sub-datasets: Oxford-easy and Oxford-hard. For Oxford-easy, we use a database (2014-12-16-09-14- 09, day) and query (2014-12-17-18-18-43, night) for train and database (2014-11-18-13-20-12, day) and query (2014-12-16-18-44-24, night) for test. In Oxford-easy, all 6 sections of the path were used, and 2 meters of frame separation was used. For Oxford-hard, we use a database (2014-11-18-13-20-12, day) and query (2014-12-16 18-44-24, night) for train and database (2014-12-16-09-14 09, day) and query (2014-12-17-18-18-43, night) for test. In Oxford-hard sections 1,2 of the path were used in training, and sections 3,4 were used in the test set, and a 1-meter frame separation is used. We call it Oxford-hard because in the train and test sets, we see different sections, and the test conditions are more challenging than the train conditions.

3) Nuscenes: The Nuscenes dataset [13] comprises 1,000 scenes, each 20s long, captured at 2 Hz from a full autonomous vehicle sensor suite. In our experiments, we exclusively utilize the Singapore scenes to focus on urban day–night traversals representative of extreme lighting transitions. We form day and night test splits based on the provided scene-level tags, and define positive sample pairs by a maximum spatial separation of 10 m and a maximum heading difference of 30 ° similar to the setup in [12]. We use Nuscenes as the only test set, using an Oxford-Easy trained model.

B. Model Implementation Details

Model hyperparameters: We use 8 heads, 8 points for the deformable spatial attention, and 2 levels, although we only feed in one feature level.

Training: In the training phase, we use the same setup as seqvlad [7]. We initialize the model with pre-trained parameters from CCT [17] and adopt the Adam optimizer [24]. All images are resized to 384 ×384, and the learning rate is 0.00001. We set the batch size = 4, with each batch consisting of a query sequence, a best positive sequence, and 5 hardest negative sequences. The length of each sequence is set to L = 5. The margin in triplet loss is specified as a = 0.1. The mining method selects training samples by first using GNSS-based labels to identify candidate positives and negatives between query and database images. For each query, the best positive is chosen as the nearest neighbor in descriptor space among the non-trivial positives—those within a predefined distance threshold but outside a trivial matching radius—using either cosine or Euclidean distance [9]. To efficiently mine hard negatives, instead of searching the entire database, a subset of potential negatives is randomly sampled from those filtered by GNSS distance (i.e., outside the positive threshold), and a cache is used to store and retrieve their descriptors. The hardest negatives are then selected from this subset based on their proximity to the query in descriptor space, ensuring efficient and focused negative mining. We implement early stopping by halting the training if the Recall@5 does not improve for 5 consecutive epochs.

Evaluation: In the evaluation phase, we use Recall@K as the performance metric. Recall@K is defined as the ratio of the number of correct queries retrieved to the total number of queries. A correct retrieval is defined as at least one of the top K retrievals being within the given radius from the ground truth position of the query. We use radii of 10 meters for Oxford, Nuscene datasets, and 1 frame for the Nordland dataset.

RESULTS

A. Recall comparison with Baseline Methods

The chosen baseline methods include state-of-the-art sequence descriptors: SeqNet [9], SeqVLAD [7], STformer [5], and JIST [11]. Their comparative results are summarized in Table II.

TABLE II: Recall performance of models and resource usage

Method	Backbone	Embed dim	Nordland			Oxford-Easy			Oxford-Hard			NuScenes			Resource Usage			
			R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	Mem (MB)	Params	Time (s)	GFLOPs
Our method	CCT384	24576	**0.9763**	0.9943	0.9947	0.8854	0.9528	0.97	**0.694**	**0.7706**	**0.7969**	**0.45**	**0.6524**	**0.7421**	180.42	13,974,722	0.0206	102
Seqvlad	CCT384	24576	0.9603	**0.9947**	0.9947	0.8275	0.9249	0.9541	0.4709	0.5601	0.6151	0.3236	0.4327	0.5142	177.66	13,309,249	0.0222	99.56
STformer	CCT384	24576	0.9697	**0.9947**	**0.995**	0.8488	0.9323	0.9633	-	-	-	0.4461	0.5319	0.5945	276.55	13,539,522	0.0338	127.74
Seqnet	NetVLAD	4096	0.7943	0.9013	0.9317	0.5363	0.7467	0.8241	0.3064	0.4505	0.5228	0.0827	0.1575	0.2031	676.09	83,890,176	0.0042	0.17
Our method+PCA	CCT384	4096	0.961	0.9923	0.9937	**0.8882**	**0.9544**	**0.971**	0.6874	0.7659	0.7918	0.45	0.6563	0.7413	-	-	-	-
Seqvlad + PCA	CCT384	4096	0.9567	0.9947	0.9947	0.828	0.9251	0.9557	0.4603	0.555	0.6151	0.3236	0.4327	0.5146	-	-	-	-
STformer + PCA	CCT384	4096	0.97	0.9947	0.995	0.8534	0.9372	0.9667	-	-	-	0.4461	0.5319	0.5949	-	-	-	-
JIST	ResNet	512	0.821	0.92	0.945	0.5868	0.7416	0.8134	0.4423	0.5405	0.6139	0.2047	0.3146	0.3713	159.65	11,439,170	0.0077	53.63
Our method+PCA	CCT384	512	0.949	0.99	0.9943	0.887	0.9554	0.9731	0.6516	0.7459	0.7765	0.439	0.6386	0.7382	-	-	-	-
Seqvlad + PCA	CCT384	512	0.9567	0.9947	0.9947	0.8236	0.9244	0.9564	0.4383	0.533	0.6049	0.3146	0.4236	0.5024	-	-	-	-
STformer + PCA	CCT384	512	0.9637	0.9933	0.995	0.8513	0.9369	0.9667	-	-	-	0.4445	0.537	0.5969	-	-	-	-

TABLE III: Ablation study to validate the effectiveness of sequential frames, temporal recurrent mechanism, and spatial deformable attention

Method	Oxford-Easy			Oxford-Hard			NuScenes		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Our Method (seqL=1)	0.8813	**0.9557**	0.9708	0.6658	0.7431	0.7702	0.4508	0.7004	0.7720
Our Method (seqL=2)	0.8800	0.9546	**0.9731**	0.6807	0.7549	0.7816	**0.5638**	**0.7453**	**0.8091**
Our Method (seqL=2, no recurrent)	0.8688	0.9477	0.9669	0.5024	0.6347	0.6834	0.3744	0.5370	0.6059
Our Method (seqL=5)	**0.8854**	0.9528	0.9700	**0.6940**	**0.7706**	**0.7969**	0.4500	0.6524	0.7421
Our Method (seqL=5, no recurrent)	0.8567	0.9423	0.9621	0.5440	0.6661	0.7078	0.4161	0.5602	0.6287

Our method outperforms baselines on the Oxford and NuScenes datasets, while SeqVLAD, STformer, and our method achieve near-perfect recall on Nordland. This can be attributed to Nordland's relatively minor lighting variations, which diminish the visibility of improvements from our lighting-robust design. On the Oxford-Easy dataset, we achieve a +3% recall@1 gain over STformer. While Oxford-Easy presents more lighting challenges than Nordland, its train and test sets share overlapping locations, simplifying the task.

Substantial gains are observed on harder benchmarks: Oxford-Hard (poor lighting, rain, and unseen sections) and NuScenes (poor lighting and cross-domain train-test splits). On Oxford-Hard, we achieve approximately +20% gains in recall@1,5,10 over SeqVLAD. For NuScenes, we observe +12% and +15% improvements in recall@5 and @10, respectively, over STformer, while the recall@1 results are similar. To enable fair comparison with baselines using lower-dimensional descriptors, we apply PCA [25] to our model's outputs.

On Appendix A, we demonstrate the results of utilizing our model's sequence length-agnostic function.

B. Resource usage

Efficent computational resource usage is critical in real-world VPR systems and is a primary focus of our work. Table II summarizes sequence-extraction latency, GFLOPs, and parameter counts across the evaluated methods. Compared to STformer—our main attention-based spatio-temporal baseline—our approach achieves 39% faster descriptor extraction per sequence, requires 20% fewer GFLOPs, and consumes 35% less peak memory; we also outperform SeqVLAD in raw extraction speed.

Figure 3 provides a module-wise breakdown of extraction time for our method, STformer, and SeqVLAD. Thanks to

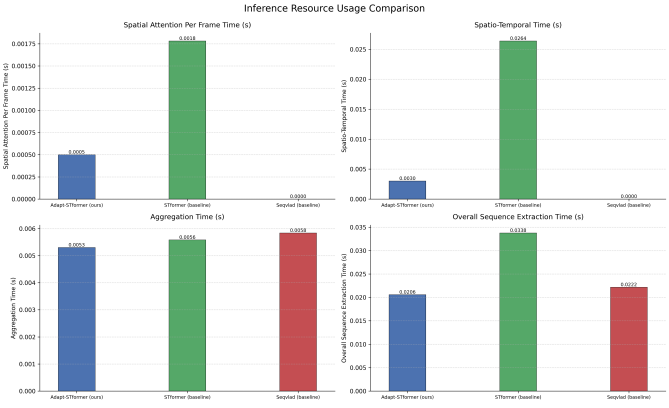


Fig. 3: Detailed extraction time visualisation

deformable attention, our spatial-attention stage runs 3.6× faster per frame than STformer's dense attention. Although STformer's spatial attention can be more efficient on longer sequences since our method processes each frame iteratively, our integrated spatio-temporal module—by omitting a separate temporal-attention block—is 8.8× faster overall. In the aggregation stage (identical across all three methods), we observe a slight additional speedup, which we attribute to our use of SeqGEM pooling multiple frame features before entering the SeqVLAD layer.

All benchmarks were conducted on an NVIDIA RTX 8000 GPU using the PyTorch [26] framework.

C. Ablation Study

We conduct an ablation study on the Oxford and Nuscenes datasets to evaluate (1) the impact of temporal sequence length (seqL), (2) the necessity of our recurrent mechanism. and (3)

isolate the impact of the spatial deformable attention module. Ablation results are on table III.

Temporal Sequence Length: Table III compares models trained and tested on seqL=1 versus seqL>1. On Oxford-Hard, using sequences longer than 1 (with recurrence) yields 3% gains across recall@1,5,10. For Oxford-Easy, seqL=1 and seqL>1 perform similarly, likely due to the dataset's simplicity. In NuScenes, seqL = 2 achieves optimal performance, outperforming seqL=1 by 11%, 4%, and 3% in recall @ 1,5,10, and surpassing seqL = 5 by 11%, 9%, 9% and 6%. This aligns with prior work showing dataset-dependent optimal sequence lengths [27].

Recurrent Mechanism: Removing the recurrence mechanism (not feeding in t-1 output features back as the current query) across all datasets see a drop in recall, demonstrating the necessity of having a recurrent mechanism when doing a video VPR task. causes a recall drops in the ranges of 2-20 %, underscoring its importance.

Spatial deformable attention module: By examining the results of the train/test on seqL=1, we can attribute most of the difference in recall performance to our spatial attention module, as the recurrent temporal mechanism is not utilized. As shown in Table III, we continue to outperform our baseline results on Table III on the Oxford and NuScenes datasets.

D. Qualititative Results

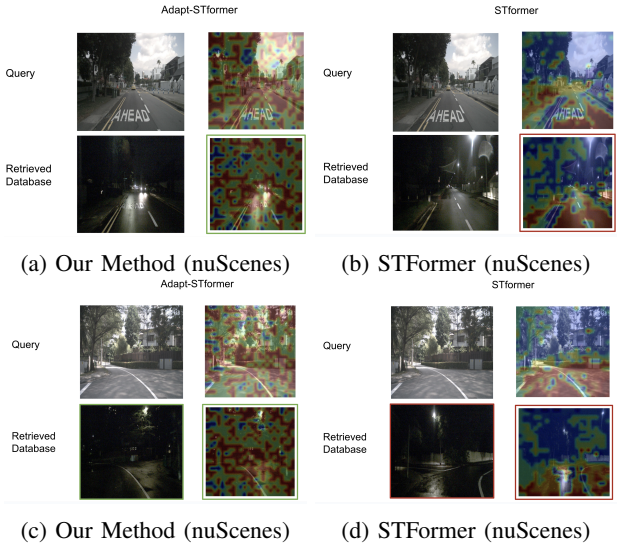
Based on the ablation above, our recall gains stem primarily from the spatial attention module. To illustrate this, we visualize attention maps from our method versus baselines. In Figure 4 (a)(b) and (c)(d), two Nuscenes examples show our model retrieving correctly while STFormer fails. In both day and night scenes, STFormer produces smooth, concentrated activations over a few regions, whereas our model displays many scattered high-activation patches across the entire frame. This difference arises because global dense attention computes similarity over every pixel pair—resulting in blob-like activations on the most informative areas—while deformable attention learns a small set of sparse offsets that jump directly to features relative to the token, yielding scattered activations. Under poor lighting, when many features are weak, capturing any available geometric cue is crucial for successful recall, much like a human navigating in the dark by noticing any visible structural clue rather than focusing on regions that had informative features under good lighting. We observe a similar pattern in Figures 5 (a)(b) and (c)(d) when comparing our method to SeqVLAD activation maps on the Oxford Hard dataset.

E. Limitations

Our model is unable to yet fully take advantage of the most powerful backbones like the DINOv2 variants [28], thus we are not confident in claiming global SOTA recall results. However, CCT384 [17] retains its advantages of computational efficiency over DINOv2.

III. Conclusion

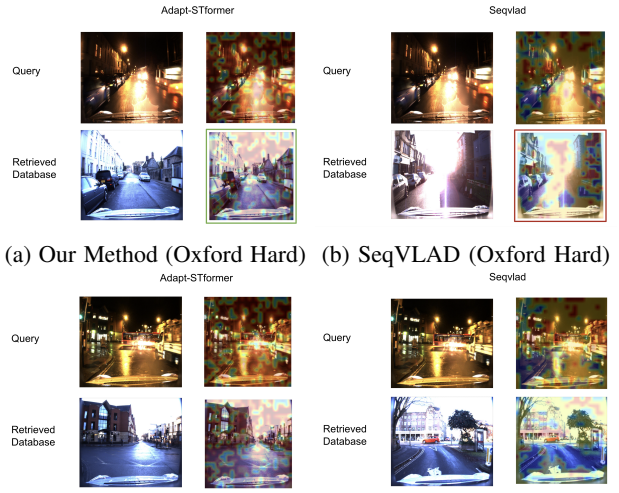
We explore how attention mechanisms can more effectively capture spatial and temporal cues in VPR by introducing a novel approach that leverages ideas from related domains. Our method shows promising improvements in recall, flexibility, and computational efficiency. We hope this work inspires future VPR research to tackle even more challenging environments while also improving its real-world practicality.



(a) Our Method (nuScenes) (b) STFormer (nuScenes)

(c) Our Method (nuScenes) (d) STFormer (nuScenes)

Fig. 4: Qualitative comparisons across datasets.



(a) Our Method (Oxford Hard) (b) SeqVLAD (Oxford Hard)

(c) Our Method (Oxford Hard) (d) SeqVLAD (Oxford Hard)

Fig. 5: Qualitative comparisons across datasets.

References

[1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017, pp. 5998–6008.
[3] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 813–824.
[4] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4769–4778.
[5] J. Zhao, F. Zhang, Y. Cai, G. Tian, W. Mu, C. Ye, and T. Feng, "Learning sequence descriptor based on spatio-temporal attention for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2351–2358, 2024.
[6] W. Li, H. Luo, Z. Lin, C. Zhang, Z. Lu, and D. Ye, "A survey on transformers in reinforcement learning," *arXiv preprint arXiv:2301.03044*, 2023.
[7] R. Mereu, G. Trivigno, G. Berton, C. Masone, and B. Caputo, "Learning sequential descriptors for sequence-based visual place recognition," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 383–10 390, 2022.
[8] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1643–1649.
[9] S. Garg and M. Milford, "Seqnet: Learning descriptors for sequence-based hierarchical place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4305–4312, 2021.
[10] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5297–5307.
[11] G. Berton, G. Trivigno, B. Caputo, and C. Masone, "Jist: Joint image and sequence training for sequential visual place recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1310–1317, 2024.
[12] F. Ge, Y. Zhang, S. Shen, Y. Wang, W. Hu, and J. Gao, "Bev^{2}pr: Bev-enhanced visual place recognition with structural cues," in *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 13 274–13 281.
[13] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.
[14] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations (ICLR)*, 2021.
[15] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Computer Vision – ECCV 2022*, ser. Lecture Notes in Computer Science, vol. 13669. Springer, 2022, pp. 1–18.
[16] X. Xu, Y. Jiao, S. Lu, X. Ding, R. Xiong, and Y. Wang, "Leveraging bev representation for 360-degree visual place recognition," *arXiv preprint arXiv:2305.13814*, 2023.
[17] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, "Escaping the big data paradigm with compact transformers," *arXiv preprint arXiv:2104.05704*, 2021.
[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.
[19] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4784–4793.
[20] R. M. Schmidt, "Recurrent neural networks (rnns): A gentle introduction and overview," *arXiv preprint arXiv:1912.05911*, 2019.
[21] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," in *Proceedings of the IEEE ICRA Workshop on Long-Term Autonomy*, 2013.
[22] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000km: The oxford robotcar dataset," *International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.
[23] W. Maddern, G. Pascoe, M. Gadd, D. Barnes, B. Yeomans, and P. Newman, "Real-time kinematic ground truth for the oxford robotcar dataset," *arXiv preprint arXiv:2002.10152*, 2020.
[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[25] H. Jégou and O. Chum, "Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.
[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019, pp. 8024–8035.
[27] C. Malone, A. Vora, T. Peynot, and M. Milford, "Dynamically modulating visual place recognition sequence length for minimum acceptable performance scenarios," in *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 3340–3347.
[28] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jégou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

IV. Appendix A

By taking the same model trained on seqL=5, we use it to infer on seqL of 1 and 3, results detailed on table IV and V. This could be a useful feature in the real world, where sometimes, due to computational constraints, shorter sequences might be needed for inference compared to training. Although we do see a drop in recalls in the range of 1-6 percent across the results, the results remain competitive. We hypothesize that this performance drop stems from the model's optimization for its training sequence length.

TABLE IV: Performance of our method trained on seqL=5 across different sequence lengths (Part 1/2)

Method	Embed Dim	Nordland			Oxford-Easy		
		R@1	R@5	R@10	R@1	R@5	R@10
Our method (seqL=5)	24576	0.9763	0.9943	0.9947	0.8854	0.9528	0.9700
Our method (seqL=3)	24576	0.9653	0.9927	0.9943	0.8662	0.9408	0.9615
Our method (seqL=1)	24576	0.9740	0.9930	0.9943	0.8344	0.9280	0.9510

TABLE V: Performance of our method trained on seqL=5 across different sequence lengths (Part 2/2)

Method	Embed Dim	Oxford-Hard			NuScenes		
		R@1	R@5	R@10	R@1	R@5	R@10
Our method (seqL=5)	24576	0.6940	0.7706	0.7969	0.4500	0.6524	0.7421
Our method (seqL=3)	24576	0.6779	0.7549	0.7844	0.4543	0.6051	0.6618
Our method (seqL=1)	24576	0.6575	0.7341	0.7690	0.4465	0.6224	0.7130