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

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

***Abstract*—Modern sequential Visual Place Recognition (seq-VPR) uses transformers to better understand spatio-temporal features available in sequential frames. However, existing transformer-based seq-VPR research prioritizes performance over real-world deployability. A deployable seq-VPR needs to handle a variable number of frames per sequence (sequence length) and be efficient in both inference speed and memory usage. To bridge this gap, we propose Adapt-STformer, which combines two key ideas. First, a recurrent mechanism that computes temporal attention between adjacent frames iteratively, naturally being sequence length agnostic. Although iterative attention is inefficient, we counterbalance it with a deformable transformer encoder. By attending to only a few offsets per query, deformable transformer encoder computes spatial attention much faster than transformer encoders used in existing methods. We call this combined innovation the Recurrent Deformable Transformer Encoder module. On Nordland, Oxford, and NuScenes, Adapt-STformer cuts sequence extraction time by 40 %, reduces memory by 35 %, and improves recall by up to 15 % over our strongest baseline. This demonstrates that our transformer-based seq-VPR method balances between flexibility, efficiency, and performance.**

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

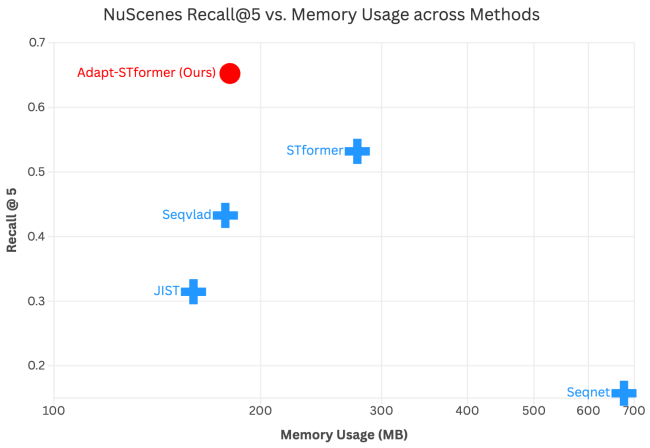


Fig. 1: Adapt-STformer (Ours) utilizes a reasonable amount of memory while achieving higher recall compared to baseline methods on the NuScenes dataset

I. Introduction

Sequential Visual Place Recognition (seq-VPR) aggregates temporally adjacent frame features into a compact descriptor. By aggregating features from multiple frames, seq-VPR dilutes the influence of distortions, such as lighting noise and occlusions, that occur in only a subset of frames [1], [2]. This temporal fusion also reinforces discriminative features, improving performance under difficult conditions compared to single-frame VPR methods. [3].

Earlier seq-VPR methods only aggregate descriptors or local features of multiple frames, failing to capture temporal interactions across frames [1], [4], [5]. Inspired by advances in video understanding [6], [7], recent seq-VPR methods integrate transformer modules to model both inter (temporal) and intra (spatial) frame interactions. These methods have yielded impressive SOTA performance [8], [9], especially under challenging VPR conditions.

However, existing transformer-based seq-VPR methods suffer from two key drawbacks. First, approaches like STformer require the same number of frames per sequence (sequence length) during training and inference. This requirement limits flexibility, since real-world VPR streams do not guarantee consistent sequence lengths. Second, methods that accept arbitrary sequence lengths (e.g. [8]) use a separate transformer encoder to learn cross-frame temporal interactions. Since each frame S generates P tokens, transformer computation scales $\mathcal{O}((S \cdot P)^2)$ in both compute and memory, making real-time deployment impractical.

To address this, we introduce the Recurrent Deformable Transformer Encoder module that combines two key innovations. First, a recurrent mechanism computes temporal attention between adjacent frames iteratively and is naturally sequence-length agnostic. Secondly, to offset the lower efficiency of iterative attention, we pair it with a Deformable Transformer Encoder, proposed by [10], for learning spatial features. The encoder use deformable attention (deform-attn), where each query attends only to a sparse set of learned offsets, to compute attention. Thus, computation scales roughly linearly rather than quadratically with token count. This contrasts with dense attention used in existing seq-VPR methods, where each query attends to every other token. Our experiments show that deform-attn focuses on structural cues that are crucial in challenging conditions, while the recurrent mechanism effectively leverages temporal information present in seq-VPR.

We summarise our contributions in two points:

- We design, Adapt-STformer, a seq-VPR method well balanced between efficiency, flexibility, and performance

(see fig. 1). Achieving a 40 % reduction in sequence extraction time, 35 % memory savings, and up to 15 % recall improvement over our strongest baseline.

- We propose the Recurrent Deformable Transformer Encoder module which combines two innovations. First, a recurrent mechanism that computes temporal attention between adjacent frames iteratively, making it sequence length agnostic. Second, a deformable transformer encoder that sparsely attends to informative features, reducing spatial attention time by 4.5× per frame, while maintaining high performance.

II. Related Works

A. Classic seq-VPR

Early work matched frame descriptors over time [11], but was costly and motion-assumption sensitive. SeqNet [1] used a lightweight 1D temporal convolution to encode a sequence into one descriptor. SeqVLAD [4] extended NetVLAD [12] to aggregate variable-length sequences without retraining, and JIST [5] employed SeqGeM to compress sequences efficiently while outperforming SeqVLAD. These non-attention methods deliver the flexibility and efficiency we value, but they lack the ability to selectively focus on salient spatio-temporal cues, limiting robustness in difficult conditions [9].

B. Attention-based seq-VPR

STformer [9] was the first fully transformer-based seq-VPR method. It uses spatial attention for frame-level features and a sliding-window temporal transformer with relative positional encoding to model dynamics. It outperforms earlier non-attention methods and showcases transformers' potential in seq-VPR. However, STformer is tied to a fixed sequence length and relies on dense attention. CaseVPR [8] removes the fixed-length constraint (it is sequence-length invariant) but still models temporal relationships with a full transformer, making it computationally expensive. In short, [8], [9] trade flexibility and/or efficiency for performance. Our method aims to restore efficiency and flexibility while preserving high recall.

C. Deformable Transformer

Deformable Transformer in Deformable DETR [10] replaces dense attention with adaptive sparse sampling. Each query predicts a small set of offsets $\Delta\mathbf{p}$ and weights A via a lightweight projection, easing quadratic cost and slow convergence on high-res features. We use this module to learn spatial features from images.

BEVFormer [2] fuses past BEV features recurrently and uses a deformable transformer encoder for spatial lookup across multi-camera views. We draw on this design, adapting its recurrent fusion and deformable encoding to our seq-VPR method. Xu et al. [13] showed this encoder's value for multi-camera VPR but ignored temporal cues. Both methods assume synchronized multi-camera inputs and known intrinsics. We instead target a practical monocular, front-view setting with unknown camera parameters.

D. Challenging VPR

BEV2PR (BEV-Enhanced image VPR) [14] 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 [15] in our experiments for seq-VPR, as described in their work.

Methodology

A. Overall Architecture

Adapt-STformer takes as input a frame sequence $S = \{s_i\}_{i=1}^{L}$, with each $s_i \in \mathbb{R}^{H\times W\times 3}$, and outputs a NetVLAD descriptor $V \in \mathbb{R}^{C\times D}$, where C is the number of VLAD clusters and D is the embedding dimension.

Adapt–STformer comprises three main stages. First, a CCT384 encoder maps the input frame sequence S to a feature tensor $F \in \mathbb{R}^{L\times n\times D}$, where n is the number of tokens per frame and D is the feature dimension. Next, our recurrent deformable transformer refines F in place—preserving its $L \times n \times D$ shape. Finally, a NetVLAD aggregation layer pools the $L \times n$ token embeddings into an output descriptor $V \in \mathbb{R}^{C\times D}$

Recurrent Deformable Transformer Encoder: Inspired by [10], we employ their Deformable Transformer Encoder to learn spatial features. Deformable attention samples a small set of K key-points near each query's reference point—rather than computing full pairwise attention—significantly reducing computational cost. We overlay a uniform grid of 2D reference points across each feature map as in [2], [10]. To capture temporal dependencies, we introduce a recurrent update: given the frame sequence indices $t = 0, \ldots, L-1$, at each step t the query Q_t is the encoder output from step $t-1$ and the value V_t is the CCT384 embedding of frame t, i.e.,

$$\text{Output}_t = \text{DeformableTransformerEncoder}(Q_{t-1}, V_t),$$

with Q_0 initialized as the first frame's token embeddings plus a learnable offset. This design promotes inter-frame attention and remains agnostic to sequence length.

Aggregation Module: We save the output from each timestep to form a tensor of shape $F \in \mathbb{R}^{L\times n\times D}$. We permute F to $F' \in \mathbb{R}^{n\times L\times D}$ (treating tokens as the batch dimension) and apply SeqGeM [5], a learnable mean-pooling over the temporal axis, collapsing the L dimension to 1:

$$\tilde{F} \;=\; \text{SeqGeM}(F') \;\in\; \mathbb{R}^{1\times n\times D}$$

We then feed $\tilde{F}$ into SeqVLAD [4], which aggregates the n D-dimensional embeddings into a descriptor $V \in \mathbb{R}^{C\times D}$,where C is the number of VLAD clusters. Both SeqGeM and SeqVLAD remain agnostic to the input sequence length L.

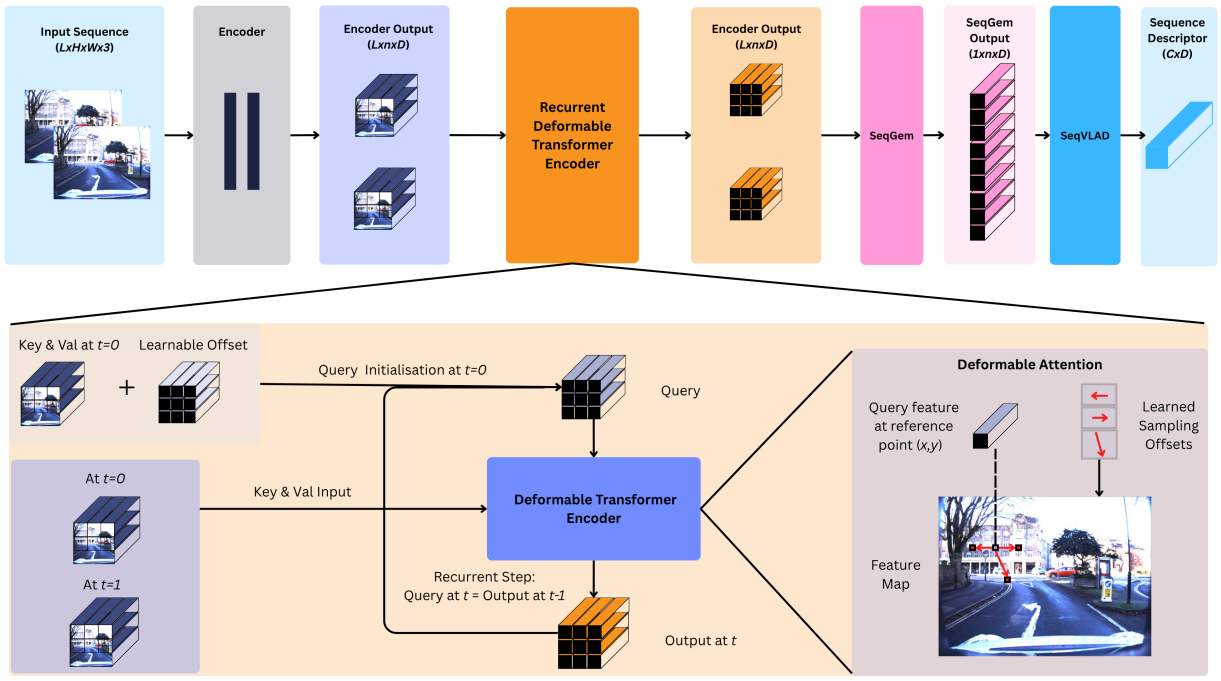


Fig. 2: **Proposed architecture of Adapt-STformer, example setup where $L = 2$:** An input sequence composed of L frames is processed by an image encoder for feature extraction. These features serve as keys and vals for our Recurrent Deformable Transformer Encoder module. For $t > 0$, the Query for Deformable Transformer Encoder is its output at $t - 1$, while keys and vals correspond to encoder features at t. Outputs from all timesteps are concatenated, then aggregated through SeqGem or SeqVLAD to generate a sequential descriptor.

B. Loss Function

We utilise the same triplet loss detailed in seqnet [1] 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.

NordLand: The Nordland dataset [16] 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.

Oxford RobotCar: The Oxford RobotCar [17] [18] 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. The Oxford Easy dataset corresponds to the Oxford2 split used in [9]. 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.

NuScenes: The NuScenes dataset [15] comprises 1,000 scenes, each 20s long, captured at 2 Hz from a full autonomous vehicle sensor suite. We select the Singapore scenes to capture

urban day–night traversals under extreme lighting transitions. We split scenes into day and night using scene tags and treat pairs with under a 30° angular difference as positives, following [14]. 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 training, we follow SeqVLAD [4] by initializing from CCT [19] and using Adam (LR $1e$-5). Images are resized to 384×384, batch size is 4 (one query, one best positive, five hardest negatives), sequence length $L = 5$, and triplet margin $a = 0.1$. Positives and negatives are mined via GNSS labels: the best positive is the nearest non-trivial neighbor in descriptor space, and negatives are drawn from GNSS-filtered candidates—cached for efficiency—and the hardest are selected. Training stops early if Recall@5 does not improve for five straight 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

The chosen baseline methods include state-of-the-art sequence descriptors: SeqNet [1], SeqVLAD [4], STformer [9], and JIST [5]. Their comparative results are summarized in Table II.

Results on easier VPR datasets: 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.

Results on harder VPR datasets: 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.

Other Experiments: To enable fair comparison with baselines using lower-dimensional descriptors, we apply PCA [20] to model outputs. On Appendix A, we also demonstrate the results of utilizing our model's sequence length-agnostic function.

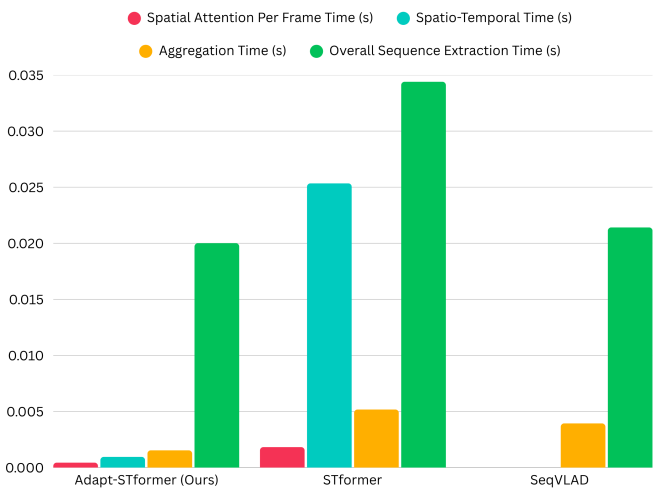


Fig. 3: Per module extraction speed compared between Adapt-STformer (Ours), STformer and Seqvlad

B. Resource usage

Computational efficiency is crucial for real-world deployability and the primary focus of our work. Table II summarizes sequence-extraction latency, GFLOPs, and parameter counts across the evaluated methods. Compared to STformer, the main transformer-based seq-VPR baseline, our approach achieves 40% faster descriptor extraction and consumes 35% less memory per sequence.

Figure 3 shows a module-wise breakdown of extraction time for our method, STformer, and SeqVLAD. Thanks to deformable attention, our spatial-attention stage runs 4.5 $\times$ faster per frame than STformer's. Although STformer's parallel spatial attention scales well with longer sequences, our Deformable Recurrent Transformer Encoder does not rely on a separate temporal transformer encoder, making it 25.3 $\times$ faster overall. In the aggregation stage (identical across all three methods), we see further speed-ups of 2.6 $\times$ over SeqVLAD and 3.5 $\times$ over STformer. This is because Adapt-STformer uses SeqGeM to pool the $L \times n \times D$ into a single $1 \times n \times D$ tensor, thus reducing the aggregator's workload. By combining our efficient spatio-temporal and aggregation layers, our extraction time is even slightly faster than SeqVLAD (a non-transformer-based method). All benchmarks were conducted on an NVIDIA RTX 8000 GPU using the PyTorch [21] framework.

C. Ablation Study

We conduct an ablation study on the Oxford and Nuscenes datasets to evaluate the impact of sequence length (L), compare temporal fusion techniques (Transformer vs SeqGEM only vs Recurrent + SeqGEM (ours)), and isolate the impact of the Deformable Transformer Encoder module. Ablation results are on table III and IV. All models are trained on sequence length $L=5$ but evaluated at $L=3$ and $L=1$, demonstrating their ability to handle shorter sequences when inference resources are limited (sequence length agnostic). We

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)	Parameters	Time (s)
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	13.3 M	0.0222
STformer	CCT384	24576	0.9697	0.9947	0.9950	0.8488	0.9323	0.9633	–	–	–	0.4461	0.5319	0.5945	276.55	13.5 M	0.0338
Our method	CCT384	24576	0.9763	0.9943	0.9947	0.8854	0.9528	0.9700	0.6940	0.7706	0.7969	0.4500	0.6524	0.7421	180.39	13.9 M	0.0206
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	83.9M	0.0042
Seqvlad + PCA	CCT384	4096	0.9567	0.9947	0.9947	0.8280	0.9251	0.9557	0.4603	0.5550	0.6151	0.3236	0.4327	0.5146	–	–	–
STformer + PCA	CCT384	4096	0.9700	0.9947	0.9950	0.8534	0.9372	0.9667	–	–	–	0.4461	0.5319	0.5949	–	–	–
Our method + PCA	CCT384	4096	0.9610	0.9923	0.9937	0.8882	0.9544	0.9710	0.6874	0.7659	0.7918	0.4500	0.6563	0.7413	–	–	–
JIST	ResNet	512	0.8210	0.9200	0.9450	0.5868	0.7416	0.8134	0.4423	0.5405	0.6139	0.2047	0.3146	0.3713	159.65	11.4 M	0.0077
Seqvlad + PCA	CCT384	512	0.9567	0.9947	0.9947	0.8236	0.9244	0.9564	0.4383	0.5330	0.6049	0.3146	0.4236	0.5024	–	–	–
STformer + PCA	CCT384	512	0.9637	0.9933	0.9950	0.8513	0.9369	0.9667	–	–	–	0.4445	0.5370	0.5969	–	–	–
Our method + PCA	CCT384	512	0.9490	0.9900	0.9943	0.8870	0.9554	0.9731	0.6516	0.7459	0.7765	0.4390	0.6386	0.7382	–	–	–

TABLE III: Ablation study to evaluate sequential frames, temporal recurrent mechanism, and combination of recurrent and seqGeM

Temporal Method	Oxford-Hard R@5	NuScenes R@5	Mem (MB)	Time (s)
Transformer ($L = 1$)	0.7368	0.4827	106.02	0.0129
seqGeM only ($L = 1$)	0.7306	0.5154	88.71	0.0113
Ours ($L = 1$)	0.7341	0.6224	88.71	0.0117
Transformer ($L = 3$)	0.7384	0.452	152.03	0.0202
seqGeM only ($L = 3$)	0.7255	0.4622	134.39	0.0134
Ours ($L = 3$)	0.7549	0.6051	134.39	0.0137
Transformer ($L = 5$)	0.7341	0.4480	198.16	0.0319
seqGeM only ($L = 5$)	0.7247	0.4201	180.39	0.0197
Ours ($L = 5$)	0.7706	0.6559	180.39	0.0206

TABLE IV: Ablation to evaluate the number of points k used in deformable transformer encoder

K points	Oxford-Hard R@5	NuScenes R@5	Mem (MB)	Time (s)
K=576 ($L = 1$)	0.7474	0.6972	262.32	0.0134
K=8, Ours ($L = 1$) Ours	0.7341	0.6224	88.71	0.0117
K=576 ($L = 3$)	0.7486	0.6988	269.07	0.0197
K=8, Ours ($L = 3$)	0.7549	0.6051	134.39	0.0137
K=576 ($L = 5$)	0.7467	0.7059	275.53	0.02948
K=8, Ours ($L = 5$)	0.7706	0.6559	180.39	0.0206

also provide the results (see Appendix **??**) where we train/test models on the same sequence length.

Sequence Length: As shown in Table III, our recurrent+SeqGeM temporal method's R@5 on Oxford-Hard rises from 0.7341 ($L = 1$) to 0.7549 ($L = 3$) and 0.7706 ($L = 5$). On NuScenes, it likewise increases from 0.6224 to 0.6051 and 0.6559. This trend, improved recall with longer sequences, demonstrates that leveraging more frames boosts VPR performance. This proves that our temporal fusion is effective at utilising additional information present in longer frame sequences.

Temporal Fusion Techniques: Our Recurrent+SeqGeM fusion consistently boosts retrieval performance over SeqGeM alone. For $L = 1$, we achieve an R@5 of 0.7341 on Oxford-Hard (versus 0.7306) and 0.6224 on NuScenes (versus 0.5154). At $L = 3$, our method's R5 rises to 0.7549 and 0.6051—substantial gains over seqGeM's 0.7255 and 0.4622. At $L = 5$, we reach 0.7706 and 0.6559 compared to only seqGeM temporal fusion's 0.7247 and 0.4201. Across all L, our temporal method vs only seqGeM uses the same amount of memory and a neglible increase in extraction time while having substantial improvement in performance.

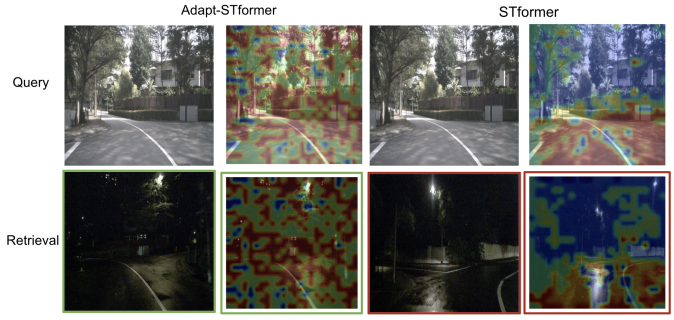
Against full Transformer fusion, our approach matches or outperforms its recall while using far fewer resources. For example, at $L = 3$ we record 0.7549 vs. Transformer's 0.7384 on Oxford-Hard and 0.6051 vs. 0.452 on NuScenes, yet we consume 29MB less memory (134.39MB vs. 152.03MB) and run 32 % faster (0.0137s vs. 0.0202s). At $L = 5$, the gap widens: we achieve 0.7706 vs. 0.7341 on Oxford-Hard and 0.6559 vs. 0.4480 on NuScenes, all while using 18MB less memory (180.39MB vs. 198.16MB) and running roughly 35% faster (0.0206s vs. 0.0319s). These results highlight the critical advantage of our iterative recurrence: it delivers substantial accuracy improvements over SeqGeM, and equal or better results than a full Transformer encoder, all with minimal additional cost and significantly greater inference efficiency.

Deformable Transformer Encoder module: Cutting the sampling points from dense attention ($K = 576$) to our sparse regime ($K = 8$) slashes memory by 66% at $L = 1$ (262,MB→88.7,MB), 50% at $L = 3$ (269,MB→134,MB) and 35% at $L = 5$ (275,MB→180,MB), while latency drops by 13%–30%. On *Oxford-Hard*, R5 is virtually unchanged at $L = 1$ (0.747→0.734) and improves at longer sequences (0.755 at $L = 3$, 0.771 at $L = 5$). The trade-off is a drop of roughly 6%–12% R5 on *NuScenes*. Overall, with just eight learned sampling points, our deformable encoder delivers comparable—or better—accuracy on Oxford while yielding large efficiency gains, albeit at some cost to NuScenes recall.

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 query 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.

(a) Nuscenes Example 1



(b) Nuscenes Example 2

Fig. 4: Qualitative comparisons across NuScenes

E. Limitations

Our model is unable to yet fully take advantage of the most powerful VPR backbones such as the DINOv2 variants [22] used in [8], [23], thus we are not confident in claiming global SOTA recall results. We currently employ CCT384 [19]—a hybrid convolutional-transformer encoder used in [4], [9]—which offers greater computational efficiency than pure ViT backbones such as DINOv2.

(a) Our Method (Oxford Hard)

(b) SeqVLAD (Oxford Hard)

Fig. 5: Qualitative comparisons across oxford-hard

III. Conclusion

We demonstrate that complex attention-based seq-VPR architectures can be streamlined for efficiency and adaptability without compromising performance in adverse conditions. As seq-VPR models continue to advance, it is crucial to prioritize designs that balance algorithmic innovation with deployment feasibility. We hope our work encourages the community to develop solutions that not only push the boundaries of performance but also address the practical challenges of real-world visual localization.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] H. Li, G. Peng, J. Zhang, M. Wen, Y. Ma, and D. Wang, "Casevpr: Correlation-aware sequential embedding for sequence-to-frame visual place recognition," *IEEE Robotics and Automation Letters*, vol. 10, no. 4, pp. 3430–3437, 2025.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] F. Lu, X. Lan, L. Zhang, D. Jiang, Y. Wang, and C. Yuan, "Cricavpr: Cross-image correlation-aware representation learning for visual place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.