
Activating Self-Attention for Multi-Scene Absolute Pose Regression

Miso Lee
Sungkyunkwan University
dlal1th557@skku.edu

Jihwan Kim
Sungkyunkwan University
damien@skku.edu

Jae-Pil Heo[†]
Sungkyunkwan University
jaepilheo@skku.edu

Abstract

Multi-scene absolute pose regression addresses the demand for fast and memory-efficient camera pose estimation across various real-world environments. Nowadays, transformer-based model has been devised to regress the camera pose directly in multi-scenes. Despite its potential, transformer encoders are underutilized due to the collapsed self-attention map, having low representation capacity. This work highlights the problem and investigates it from a new perspective: distortion of query-key embedding space. Based on the statistical analysis, we reveal that queries and keys are mapped in completely different spaces while only a few keys are blended into the query region. This leads to the collapse of the self-attention map as all queries are considered similar to those few keys. Therefore, we propose simple but effective solutions to activate self-attention. Concretely, we present an auxiliary loss that aligns queries and keys, preventing the distortion of query-key space and encouraging the model to find global relations by self-attention. In addition, the fixed sinusoidal positional encoding is adopted instead of undertrained learnable one to reflect appropriate positional clues into the inputs of self-attention. As a result, our approach resolves the aforementioned problem effectively, thus outperforming existing methods in both outdoor and indoor scenes.

1 Introduction

Camera pose estimation is a fundamental and essential computer vision task, adopted in numerous applications such as augmented reality and autonomous driving. Geometric pipelines [1–6] with 2D and 3D data have been a mainstream with high accuracy. After extracting features and matching 2D-3D correspondences, camera pose is approximated via Perspective-n-Points (PnP) algorithm and RANSAC [7]. However, there still remains several challenges for real-world applications, including high computational cost and a huge amount of 3D point cloud.

Absolute Pose Regression (APR) tackles these issues by directly estimating the 6-DoF pose from a single RGB image in an end-to-end manner. First introduced by Kendall *et al.* [8], subsequent APR methods [9–18] have been devised based on convolutional neural networks (CNN). However, they still demand multiple models and individual optimization to be applied in real-world multi-scene scenarios. In this regard, Multi-Scene Absolute Pose Regression (MS-APR) has emerged to satisfy the needs of speed and memory efficiency across multiple scenes [19]. MStTransformer [20] pioneers a streamlined one-stage MS-APR approach with transformer architecture. It leverages the transformer decoder not only to improve memory efficiency but also to enhance accuracy significantly.

However, we point out that the learning capacity of transformer encoders is underutilized. As shown in Tab. 1, MStTransformer’s encoder self-attention modules do not significantly improve or even degrade performance. Although we discovered low-rank, collapsed attention maps, which are known

[†]Corresponding author

Table 1: Ablation on encoder self-attention modules in MSTransformer [20]. We report the average of median position and orientation errors for each experiment.

	Outdoor [8]	Indoor [27]
MST [20]	1.28m/2.73°	0.18m/7.28°
MST w/o encoder SA	1.21m/2.84°	0.18m/7.49°

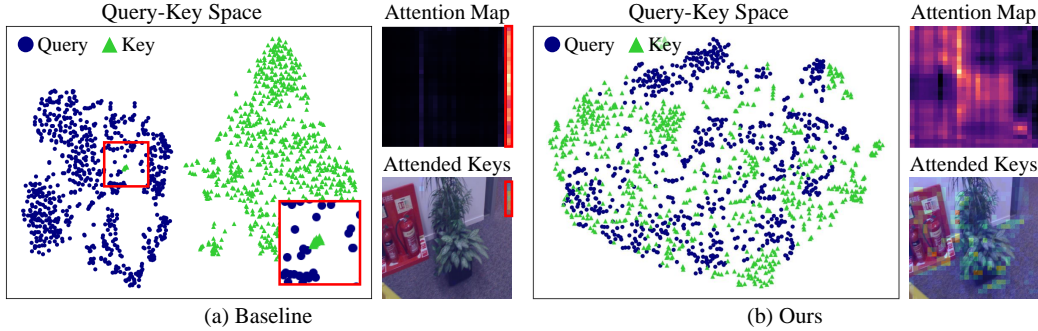


Figure 1: The figure shows query-key spaces, self-attention maps, and attended keys from the orientation transformer encoder of the baseline and ours, respectively. (a) In the case of the baseline, queries and keys are mapped in separate regions, while only a few keys are blended into the query region. Consequently, the self-attention map collapses and whole image features are represented by meaningless few keys, indicating waste of learning capacity of transformer encoder. (b) However, our solution makes queries and keys interact with each other, activating self-attention. This allows the model to obtain crucial global relations within image features, capturing salient global features.

to cause gradient vanishing [21] or training instability [22], general solutions [22–24] were not correct the problem in the case of MSTransformer. This challenge motivates us to elucidate the phenomenon from another perspective in the context of MS-APR.

Therefore, this work proposes brand new analysis: distortion of query-key embedding space, where the attention score originates. We statistically substantiate that queries and keys are mapped in completely separated regions in the embedding space while only a few keys are blended into the query region. In this situation, attention maps inevitably collapse, as all queries are represented similarly to those few keys. Intuitively, it means that all image patches are just represented by only a few image patches, as illustrated in Fig. 1. We conjecture that the task difficulty contributes to the issue; the model should extrapolate the camera pose from a single RGB image across multiple scenes. To support this, we empirically reveal that the model tends to avoid exploring self-relationships at the beginning of the training. These findings align with our additional observations that learnable positional embeddings used in self-attention are also undertrained.

Nevertheless, we expect that image features incorporating global self-relations will be valuable to estimate the camera pose by capturing salient features such as long edges and corners. This work thus introduces simple but effective solutions that activate encoder self-attention modules for MS-APR. Concretely, we design an auxiliary loss which aligns the query region and the key region. By forcing all queries and keys to be mapped into the close and dense space, they are highly encouraged to interact with each other by self-attention—like putting boxers in the ring and blowing the whistle! Furthermore, we explore various positional encoding methods based on empirical evidence since current undertrained one confuses the model to estimate the camera pose with incorrect positions. Finally, fixed 2D sinusoidal encoding is adopted instead of learnable parameter-based methods [25, 26]. This enables the model to consider appropriate positional clues of each image feature during exploring self-relationships from the beginning of the training.

As such, the model can obtain rich global relations from activated self-attention as shown in Fig. 1. These relations are incorporated into image features, acquiring more informative encoder output. Extensive experiments demonstrate that our solution recovers the self-attention successfully by preventing the distortion of query-key space and keeping high capacity of self-attention map [22]. As a result, our model outperforms existing MS-APR methods in both outdoor [8] and indoor [27] scenes without additional memory during inference, upholding the original purpose of MS-APR.

2 Related Work

Absolute Pose Regression (APR) is to estimate 6-DoF camera pose directly using only an RGB image as input. First proposed by Kendall *et al.* [8], subsequent works [9, 10, 13, 14, 17] have introduced advanced architectures and training methods. However, Sattler *et al.* [28] demonstrated that these models have a tendency to memorize training data, resulting in poor generalization when presented with novel views. In response, IRPNet [16] attempted fast APR by adopting pre-trained image retrieval model as feature extractor. On the other hand, E-PoseNet [15] substituted vanilla CNN with Group-Equivalent CNN to extract geometric-aware image features. Recently, NeRF-based models [11, 12, 29] have improved the APR performance significantly by combining existing APR models with NeRF-W [30] to capture geometric information or synthesize additional images. However, these models have the limitation of being scene-specific, making them impractical for real-world applications [20]. NeRF-based models [11, 12, 29] are particularly vulnerable since NeRF-W is not just scene-specific, but also requires lots of time for taking structure-from-motion, training, and generating additional data.

Multi-Scene Absolute Pose Regression (MS-APR) aims to estimate a camera pose from multiple scenes with a single model for scalability. Initially, MSPN [19] made a feature weight database for each scene using CNN, and then fed the indexed features into scene-specific regressors for estimating camera pose in multi-scenes. However, it still had drawbacks of training time and memory inefficiency. Against the backdrop, MStTransformer [20] enabled sound one-stage MS-APR with transformer-based architecture. They proposed to use decoder queries as scene queries to make both scene classification and camera pose regression possible at once. The whole decoder outputs are firstly used to predict the scene index, and then the decoder query corresponding to the scene is used to regress the camera pose. This allows it to become more general through various scenes. Recently, additional studies [31, 32] have been proposed fine-tuning methods, but reverting scene-specific again and using additional parameters. In contrast, we propose simple but effective training methods for training MStTransformer from scratch, maintaining the original purpose of MS-APR.

Self-Attention is a attention mechanism which represents input sequences by inter-relationships among all elements. Following the remarkable performance of transformer [33–36], it became widely adopted in various fields. MStTransformer [20] also used self-attention both the transformer encoder and decoder. However, recent studies [21–24, 37–40] have pointed out the failure of self-attention. Firstly, several studies on natural language processing [37, 38] have analyzed that self-attention map collapses to a low-rank matrix, by focusing attention scores to meaningless tokens such as CLS and SEP. In the case of vision, several works [23, 40] have addressed the issue of collapsed self-attention in transformer encoder. There are also studies [21, 22, 24, 39] which fundamentally demonstrated that self-attention maps collapse into low-rank matrices in certain conditions. Similar to these findings, self-attention modules in MStTransformer’s encoders do not significantly improve or even impair performance due to the collapse. However, we found that pioneers are not very effective in the case of MS-APR, which is shown in our experiments. Therefore, we analyse the problem from a new viewpoint, distortion of query-key space and undertrained learnable positional embedding, and propose solutions that settle down the problem successfully in MS-APR.

3 Preliminary

Camera Pose. The location of a captured image \mathcal{I} can be calculated from camera intrinsic and extrinsic parameters. The former is specific to the camera itself, and the latter corresponds to the pose of the camera with respect to a fixed world coordinate system. As such, the camera pose \mathbf{p} can be represented as (\mathbf{t}, \mathbf{r}) . Here, $\mathbf{t} \in \mathbb{R}^3$ the camera position (translation) and $\mathbf{r} \in \mathbb{R}^4$ the camera orientation (rotation). Note that \mathbf{r} is an unit quaternion vector so \mathbf{p} has 6-DoF.

Model Architecture. MStTransformer [20] suggests the model architecture inspired by DETR [35]. It is composed of a CNN, transformer encoder-decoders, a scene classifier, and regressors. Firstly, given an image \mathcal{I} , CNN features for the orientation \mathbf{f}_r is extracted from middle level while those for the position \mathbf{f}_t is computed from middle-high level. \mathbf{f}_t and \mathbf{f}_r are projected by 1×1 convolution and reshaped as $\tilde{\mathbf{f}}_t \in \mathbb{R}^{N_t \times D}$ and $\tilde{\mathbf{f}}_r \in \mathbb{R}^{N_r \times D}$ to be transformer-compatible inputs, where N_t and N_r are the number of tokens for the position transformer and the orientation transformer, respectively.

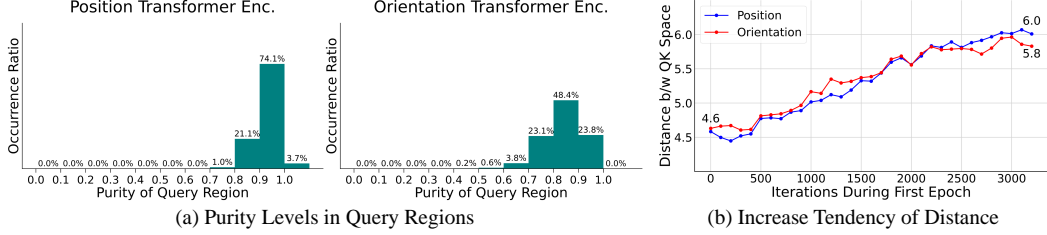


Figure 2: (a) shows the purity levels in query regions with the baseline on 7Scenes, referring the Eq. 2. Note that the purity is 1.0 when the query region is composed only with queries, but slightly lower than 1.0 when a small subset of keys resides in the query region. According to (a), statistical evidence supports the prevalent occurrence of the blending of a few keys into the query region across the entire dataset, both in the position and orientation transformer encoders. (b) illustrates the increasing tendency of distance between the query region and the key region in the encoder. They lean away each other even at the beginning of the training. Here, the distance between query region and key region is an average value across layers and heads.

The architecture of transformer encoder-decoder follows the general transformer [33, 35]. It utilizes decoder query as scene query to make it possible for both scene classification and camera pose regression at once. There are the position and the orientation transformers, each of which has scene queries $\mathbf{z}_t \in \mathbb{R}^{M \times D}$ and $\mathbf{z}_r \in \mathbb{R}^{M \times D}$ where M is the number of scenes, respectively. In the encoder, there are several layers to reinforce the CNN features. Every layer of the encoder consists of a multi-head self-attention module and multi-layer perceptron (MLP) with layer normalization and residual connection.

Meanwhile, the decoder is designed to not only enhance the decoder queries but also learn the relationship between encoder features and decoder queries. It has multiple layers like the encoder but additionally includes a multi-head cross-attention module in each layer. Afterwards, the scene classifier predicts the scene index m from concatenated decoder outputs $\mathbf{z} = [\mathbf{z}_t; \mathbf{z}_r] \in \mathbb{R}^{M \times 2D}$. In the end, $\hat{\mathbf{t}}$ and $\hat{\mathbf{r}}$ are predicted by MLP regressors from the m -th decoder outputs $\mathbf{z}_t^m \in \mathbb{R}^D$ and $\mathbf{z}_r^m \in \mathbb{R}^D$, respectively.

Self-Attention. There are three main components in attention mechanism: query \mathbf{Q} , key \mathbf{K} , and value \mathbf{V} . The final output is computed as a weighted sum of \mathbf{V} , with weights derived from the similarity score between \mathbf{Q} and \mathbf{K} . Formally, given the input $\mathbf{X} \in \mathbb{R}^{N \times D}$ with N tokens of dimension D , the h -th attention head is defined as follows:

$$\text{Attn}_h(\mathbf{X}) := \mathbf{A}_h \mathbf{V}_h, \quad (1)$$

$$\text{where } \mathbf{A}_h = \text{softmax} \left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{D/H}} \right).$$

Here, \mathbf{A}_h is h -th attention map and H is the number of heads. When inputs to query, key, and value are all the same, it is usually called as self-attention.

4 Problem Analysis

4.1 Distortion of Query-Key Embedding Space

We introduce our viewpoint why encoder self-attention modules in MStTransformer are not utilized as shown in Tab. 1. Our main intuition is that the model tends to avoid the self-attention mechanism [38, 24] by distorting the query-key embedding space due to the learning difficulty. On the face of it, collapsed self-attention maps may seem to be the root cause. However, we further investigate the issue, focusing on the query-key embedding space. The attention map is determined by computed similarity scores between queries and keys, as expressed in Eq. 1. In other words, the projection matrices and bias are what the model learns, and the attention map is merely a calculated consequence. As such, the collapsed attention map which induces the model’s bypassing comes from the distorted query-key space, therefore we highlight it to address the problem.

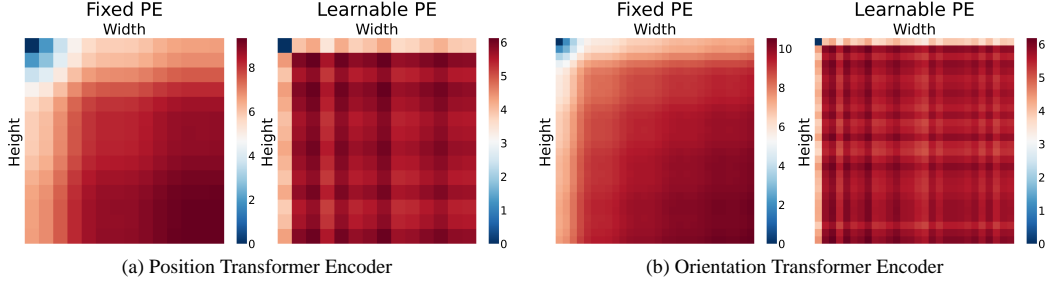


Figure 3: The figure shows L2 distances between the top-left token and other tokens based on the fixed 2D sinusoidal positional encoding and learnable positional embedding, respectively. Here, the learnable positional embedding is the result of training with the baseline. The fixed positional encoding preserves the order of input sequences, but in the case of the learnable positional embedding, tokens not aligned at the same height or width were all treated randomly further away.

When collapsing, queries and keys are completely separated while a small subset of keys are blended into the query region across all layers and heads, both in the position and the orientation encoders of the baseline [20]. To gauge the statistical incidence of such cases, we first clarify the query region and the key region in the query-key embedding space. By clustering $Q \cup K$ through k-means using a mean vector of queries \bar{q} and a mean vector of keys \bar{k} as initial centers, query-dominant region \hat{Q} is obtained from the cluster with center \bar{q} . Then we define the purity of query region \mathcal{P} as follows:

$$\mathcal{P} = \frac{1}{|\hat{Q}|} |\hat{Q} \cap Q|. \quad (2)$$

Intuitively, \mathcal{P} is 1.0 when the query region is composed only with queries, slightly lower than 1.0 when small subset of keys are blended into the query region, and about 0.5 when queries and keys are mixed together. Fig. 2 (a) illustrates the statistical results on 7Scenes [27] dataset. It shows that the phenomenon we point out is predominant throughout the whole dataset both for the position and orientation transformer encoders. Under this condition, the attention map inevitably collapses as all queries are computed to be similar to those keys.

We also verify that queries and keys lean away each other from the beginning of the training as shown in Fig. 2 (b). The model is trained to locate only a few keys close to the query region while keeping the long distance between the query region and the key region. Here, the distance is calculated using the L2 metric between their centers. These investigations imply that the model avoids the process of identifying self-relations due to the task difficulty only with the original task loss.

4.2 Undertrained Positional Embedding

Moreover, we discuss the importance of appropriate positional encoding for the task and the negative effects of the current positional embeddings in this section. As self-attention is a permutation-invariant mechanism, positional encoding is widely adopted to the queries and keys to account for the order of the input sequence [34–36]. It plays a vital role to utilize the encoder self-attention modules, particularly in MS-APR. To be specific, it is not only important to detect features like roofs and windows in the image, but also crucial to know where they are located in the image for accurate pose estimation. In other words, the model has to estimate the camera pose with shuffled image patches if it uses inappropriate positional embedding.

Despite its importance, learnable positional embeddings in the baseline remain undertrained with deactivated self-attention modules. Fig. 3 shows the visualization results of the distances between the top-left token and other tokens, based on the fixed 2D sinusoidal positional encoding and the learnable positional embedding trained in the baseline. One can observe that distances between tokens are drastically larger if the token has different vertical or horizontal position with the top-left token, in the case of learnable positional embeddings. This makes it even harder for the model to learn geometric relationships between image features by self-attention mechanism.

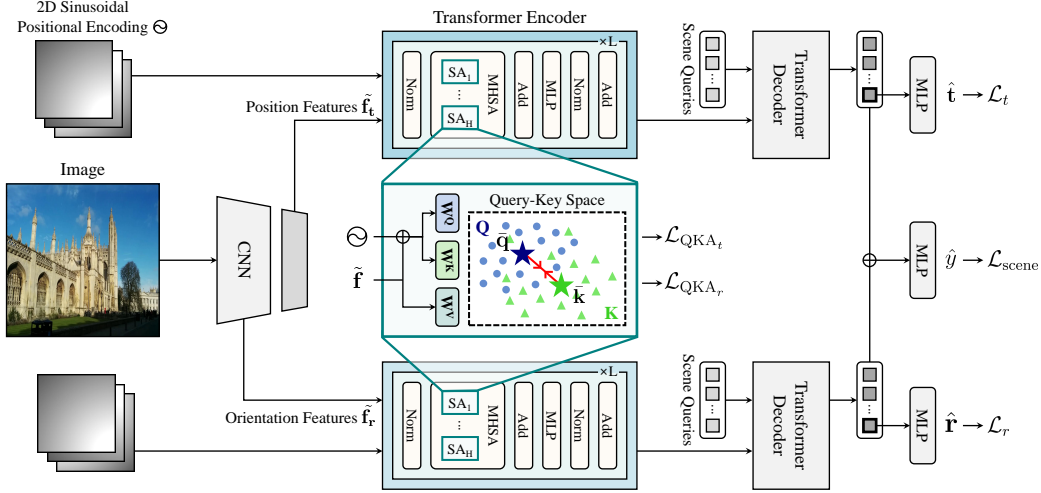


Figure 4: Fig. 4 illustrates the training pipeline with our solutions. We apply additional objectives \mathcal{L}_{QKA_t} and \mathcal{L}_{QKA_r} to the model to activate the self-attention modules. Specifically, queries Q and keys K interact with each other by forcing the centroid of query region \bar{q} and the centroid of key region \bar{k} to become closer. Here, we encode all input queries and keys with fixed 2D sinusoidal positional encoding to ensure active interaction between Q and K with reliable positional clues.

5 Activating Self-Attention for MS-APR

In this section, we introduce our solutions with full objectives for training our transformer-based MS-APR model successfully by mitigating self-attention map collapse. Firstly, we employ the L2 loss to regress the camera position and orientation. Let us denote the ground-truth camera pose as (\mathbf{t}, \mathbf{r}) and the estimated camera pose as $(\hat{\mathbf{t}}, \hat{\mathbf{r}})$. Then, the position loss and the orientation loss are defined as $\mathcal{L}_t = \|\mathbf{t} - \hat{\mathbf{t}}\|_2$ and $\mathcal{L}_r = \|\mathbf{r} - \frac{\hat{\mathbf{r}}}{\|\hat{\mathbf{r}}\|}\|_2$, respectively. Here, it is required to balance two losses to make the model learn both camera position and orientation since the former is a scalar value but the latter is a unit quaternion [8, 14]. According to [14], the final pose loss is given by:

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_t \exp(-s_t) + s_t + \mathcal{L}_r \exp(-s_r) + s_r, \quad (3)$$

where s_t and s_r are the learnable parameters to adjust the uncertainty.

Secondly, the model should be capable of classifying the scene from the input image to work with multi-scene dataset. Let us denote $y \in \mathbb{R}^M$ as one-hot vector encoding the ground-truth scene index of the input image, and $\hat{y} \in \mathbb{R}^M$ as the predicted probability distribution of scene classes from concatenated decoder outputs \mathbf{z} . Then, the scene classification loss proposed in [20] is given by:

$$\mathcal{L}_{\text{scene}} = -\sum_{j=1}^M y_j \log \hat{y}_j. \quad (4)$$

On top of it, we introduce an auxiliary loss to prevent the model from avoiding the self-attention as depicted in Fig. 4. As we discussed in Sec. 4.1, it is essential to ensure that the queries and the keys do not become alienated from each other while encouraging their interaction by embedding in close proximity. To achieve this, we propose the Query-Key Alignment (QKA) loss as an additional objective. It is defined as follows:

$$\mathcal{L}_{\text{QKA}} = \frac{1}{L} \frac{1}{H} \sum_{l=1}^L \sum_{h=1}^H \|\bar{\mathbf{q}}_h^l - \bar{\mathbf{k}}_h^l\|_2, \quad (5)$$

where L is the number of encoder layers, H is the number of heads, and $\bar{\mathbf{q}}_h^l$ and $\bar{\mathbf{k}}_h^l$ are the mean vectors of the queries and the keys per each encoder layer and head, respectively. By inducing queries and keys to be placed in a shared compact area, we ensure that they interact with each other in the

Table 2: Comparative analysis of MS-APRs on the Cambridge Landmarks dataset (outdoor localization) [8]. We report the median position/orientation errors in meters/degrees.

Method	King’s College	Old Hospital	Shop Facade	St. Mary	Average
MSPN [19]	1.73/3.65	2.55/4.05	2.92/7.49	2.67/6.18	2.47/5.34
MST [20]	0.83 /1.47	1.81/2.39	0.86/3.07	1.62/3.99	1.28/2.73
+Ours	0.88/ 1.29	1.55 / 1.87	0.79 / 2.51	1.57 / 3.50	1.19 / 2.29

Table 3: Comparative analysis of MS-APRs on the 7Scenes dataset (indoor localization) [27]. We report the median position/orientation errors in meters/degrees.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
MSPN [19]	0.09 /4.76	0.29/10.50	0.16/13.10	0.16 /6.80	0.19/5.50	0.21/6.61	0.31/11.63	0.20/8.41
MST [20]	0.11/4.66	0.24 /9.60	0.14 /12.19	0.17/5.66	0.18/4.44	0.17 /5.94	0.26/8.45	0.18/7.28
+Ours	0.10/4. 15	0.24 /8. 79	0.14 /11. 59	0.17/5. 28	0.17 /3. 48	0.17 /5. 62	0.22 /7. 58	0.17 /6. 64

self-attention module. Here, we encode all input queries and keys with fixed 2D sinusoidal positional encoding instead of undertrained learnable positional embedding. This guides the model to stably learn self-relationships from the beginning of the training. As a result, the accurate interaction can be accomplished by reflecting the verified correct positional information to input queries and keys.

We apply our QKA loss to both the position and orientation transformer encoders, while adjusting the weight of the loss to integrate with the original task loss. Accordingly, the final auxiliary loss $\mathcal{L}_{\text{aux}} = \lambda_{\text{aux}} (\mathcal{L}_{\text{QKA}_t} + \mathcal{L}_{\text{QKA}_r})$, where λ_{aux} is the weight of the auxiliary loss. $\mathcal{L}_{\text{QKA}_t}$ and $\mathcal{L}_{\text{QKA}_r}$ are QKA losses for the position and orientation transformer encoders, respectively. Putting all together, our full objective $\mathcal{L} = \mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{scene}} + \mathcal{L}_{\text{aux}}$.

6 Experimental Results

6.1 Experimental Setup

Datasets. We train and evaluate the model on outdoor and indoor datasets [8, 27], which include RGB images labeled with 6-DoF camera poses. Firstly, we use the Cambridge Landmarks which consist of six outdoor scenes scaled from $875m^2$ to $5600m^2$. Each scene contains 200 to 1500 training data. We conduct the experiment on four scenes in Cambridge Landmarks, which are typically adopted for evaluating absolute pose regression [12, 15, 20]. On the other hand, we use the 7Scenes dataset which consists of seven indoor scenes scaled from $1m^2$ to $18m^2$. Each scene includes from 1000 to 7000 images.

Training Details. We entirely follow the configuration of the baseline [20]. We train the model with a single RTX3090 GPU, Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-10}$, and the batch size of 8. For the 7Scenes dataset, we train the model for 30 epochs with the initial learning rate of 1×10^{-4} , reducing the learning rate by 1/10 every 10 epochs. In the case of Cambridge Landmarks dataset, we train the model for 500 epochs with the initial learning rate of 1×10^{-4} , reducing the learning rate by 1/10 every 200 epochs. Afterwards, we freeze the CNN and the orientation branch, then fine-tune the position branch as same as the baseline [20]. The model is trained for 60 epochs with the initial learning rate 1×10^{-4} , reducing the learning rate by 1/10 every 20 epochs.

The learnable parameters s_t and s_r for the original task loss are initialized as in [14]. Both for the position and orientation transformer encoder-decoder, the number of layers L is 6 and the number of heads H is 8. Lastly, we set the λ_{aux} for our query-key alignment loss as 0.1.

6.2 Comparative Analysis

MS-APR Methods. We compare the model trained our solutions with other single-frame multi-scene APR models for fair comparison, *i.e.*, excluding scene-specific methods. Tab. 2, Tab. 3, and Tab. 4 show the experimental results on four outdoor scenes in Cambridge Landmarks [8] and seven indoor scenes in 7Scenes [27], respectively. With our solutions, the model shows better performance through all scenes, compared to other MS-APR methods. This results verify that activating self-attention has

Table 4: Comparative analysis of the baseline and ours. We report the localization recall at several thresholds on Cambridge Landmarks and 7Scenes datasets.

Method	Cambridge Landmarks [8]				7Scenes [27]			
	(1m, 5°)	(1m, 10°)	(2m, 5°)	(2m, 10°)	(0.2m, 5°)	(0.2m, 10°)	(0.3m, 5°)	(0.3m, 10°)
MST [20]	32.6	35.8	60.4	68.2	28.8	50.2	34.4	63.5
+Ours	35.8	38.5	65.5	72.7	32.6	52.6	39.5	67.1

Table 5: Comparison with alternative methods for collapsed SA and positional encoding methods. We report the average of the median position/orientation errors on 7scenes dataset.

(a) Alternative Methods for Collapsed SA		(b) Alternative Positional Encoding Methods	
Method	7Scenes [27]	Method	7Scenes [27]
Improved SN [23]	0.18m/7.04°	T5 PE [25]	0.18m/6.97°
$1/\sqrt{L}$ -scaling [24]	0.18m/6.87°	Rotary PE [26]	0.18m/6.94°
σ Reparam [22]	0.19m/6.81°	Fixed PE	0.17m/6.64°
QK Alignment	0.17m/6.64°		

the potential to become a general module, generating task-relevant features across multiple scenes, regardless of the specific features of each scene.

Alternative Methods for Collapsed Self-Attention. We apply general-purpose techniques [22–24], proved to alleviate collapsed self-attention, to the baseline to compare the effectiveness of each method and ours in MS-APR. Here, we conduct experiments with fixed positional encoding, as incorrect inputs may suppress the potential of the solutions. Tab. 5a shows that existing techniques have little or no effect in terms of recovering the self-attention for the task. Meanwhile, ours significantly improves the baseline in both conditions. We conjecture that it is required to guide the model in a more direct manner to utilize self-attention for MS-APR.

Alternative Positional Encoding Methods. We conduct experiments with recent advanced learnable positional encoding methods [25, 26] with our QKA loss. As shown in Tab. 5b, even advanced methods are not suitable to the task as they mainly depend on learnable parameters. We assume that purified positional clues, *i.e.* not randomly initialized ones, is deemed important to stabilize the training of the embedding space in the case of MS-APR.

6.3 Quantitative Analysis

Attention Entropy of Self-Attention Maps. To provide quantitative evidence that our solutions enhance performance of model by activating the encoder self-attention, we measure attention entropy [22] defined as follows:

$$\text{Ent}(\text{Attn}(\mathbf{X})) = \frac{1}{H} \sum_{h=1}^H \text{Ent}(\mathbf{A}_h), \quad (6)$$

$$\text{where } \text{Ent}(\mathbf{A}_h) = \frac{1}{N} \sum_{i=1}^N \left(- \sum_{j=1}^N a_{ij} \log(a_{ij}) \right). \quad (7)$$

Note that high attention entropy indicates training stability and high representation capacity of self-attention map [22].

Fig. 5 shows the attention entropy for each encoder layer in both the position and orientation transformers. As illustrated, the model employing our solutions has significantly higher attention entropy than the original baseline model across all encoder layers. These results validate that our solutions successfully induce the transformer-based model to activate the encoder self-attention, thus improving the representation quality of self-attention map.

Purity Levels in Query Regions. To verify the effectiveness of our auxiliary loss on rectifying the distortion of query-key space, we measure the purity levels in query regions defined in Eq. 2. The statistical results for both the baseline and the model employing our solutions are presented in Fig. 6.

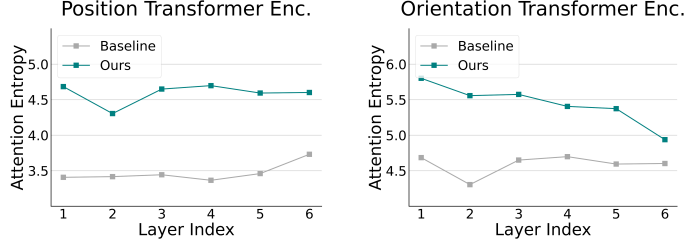


Figure 5: The figure shows the attention entropy of the baseline and ours for each encoder layer. It demonstrates that our solutions significantly improve the utilization of encoder’s learning capacity.

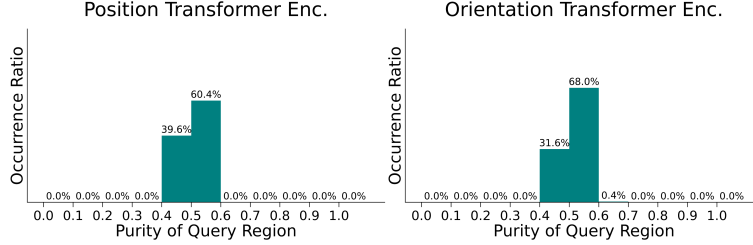


Figure 6: The results show the change of purity of query region \mathcal{P} , defined in Eq. 2, across 7Scenes dataset. Note that the purity is 1.0 when the query region is composed only with queries, but slightly lower than 1.0 when few keys resides in the query region. Compared to the baseline, there are no cases where only a few keys are blended into the query region with our solution.

Here, the purity levels represent the proportion of the queries allocated to the query-dominant cluster. Our observations reveal that the nearly half keys belong to the query region in almost all cases, indicating that the query and key regions are highly interleaved. This demonstrates that our solutions restrain all queries from being treated as similar to only a few keys, thus preventing self-attention collapse and activating self-attention.

6.4 Qualitative Analysis

Fig. 7 reports visualization of query-key space by t-SNE and attention results of the baseline and the model employing our solution. Note that the attention results depict the scores of the keys, averaged over the queries in the self-attention map. One can observe that the query regions and key regions of the baseline are separate. In contrast, the regions are highly aligned by our solutions, reducing the occurrence of only a few keys blended into the query region. This condition necessitates that the model reflects global interactions between queries and keys, therefore, global relations are incorporated into image features as depicted in Fig. 7. We can see that geometric relations such as long edges are usually obtained, which are the critical points for the task. More examples are in the supplementary material.

6.5 Ablation Study

We conduct an ablation study on our solutions to verify the effectiveness of each component. As shown in Tab. 6, our QKA loss enhances the performance in both the outdoor and indoor datasets. Additionally, overall performances are further improved with fixed positional encoding, rectifying the incorrect inputs in the self-attention. This ablation study indicates that several useful self-relation can be extracted from noisy content features by self-attention, but reliable positional clues are particularly important to learn various crucial relationship for estimating more accurate camera pose.

7 Discussion

Limitation and Future Work. The proposed auxiliary loss is designed based on the assumption that an image has many key features which are useful for estimating the camera pose. However, there

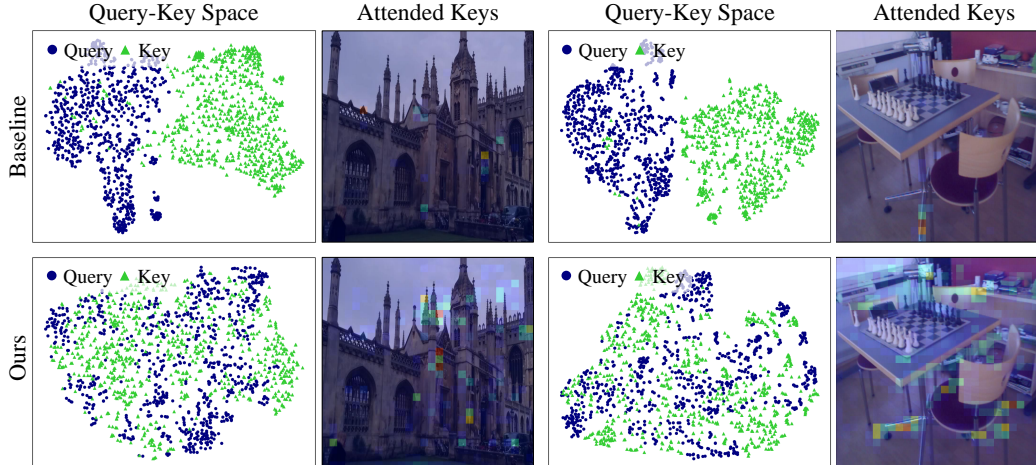


Figure 7: The figure shows the examples of t-SNE results of query-key space and attention results of the baseline and the model employing our solution. The query and key regions are separate in the baseline while our solutions align the regions, enabling the model to focus on salient global features and incorporate self-relations into image features.

Table 6: Ablation on our solutions. We report the average of the median position/orientation errors.

Method	Outdoor [8]	Indoor [27]
MST [20]	1.28m/2.73°	0.18m/7.28°
MST+Fixed PE	1.29m/2.46°	0.18m/6.80°
MST+QKA	1.22m/2.68°	0.18m/7.09°
MST+Fixed PE+QKA	1.19m/2.29°	0.17m/6.64°

could be a side effect if only a few key features are in the image; *i.e.*, if a dynamic moving object mainly occupies the image. Developing an algorithm to decide whether global self-attention is useful for the image or not could be an interesting future work.

Broader Impacts. The camera pose estimation technology with this work can be expected to have several positive social impacts, such as being utilized in programs to locate missing persons. On the other hand, it could potentially be misused for purposes such as illegal location tracking programs.

8 Conclusion

This work has exposed that self-attention modules in transformer encoders are not activated in MS-APR. Looking beyond the collapsed self-attention maps, we have discovered the distortion of query-key embedding space; queries and keys are entirely separate while a small subset of keys resides in the query region. We have examined the predominance of the phenomenon, adducing statistical and empirical evidence. Additionally, the importance of accounting for the order of the input sequence has been addressed in the context of the task. Based on these analyses, this work has proposed simple but effective solutions to activate self-attention by rectifying the distorted query-key space with alignment loss and appropriate positional clues. As a result, the model could learn salient global features through self-attention, thus achieving state-of-the-art performance both in outdoor and indoor scenes without additional memory in inference. Our extensive experiments have demonstrated the effectiveness of proposed solutions in terms of recovering self-attention for MS-APR.

Acknowledgement

This work was supported in part by MSIT&KNPA/KIPoT (Police Lab 2.0, No. 210121M06), MSIT/IITP (No. 2022-0-00680, 2020-0-01821, 2019-0-00421, RS-2024-00459618, RS-2024-00360227, RS-2024-00437102, RS-2024-00437633), and MSIT/NRF (No. RS-2024-00357729).

References

- [1] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9): 1744–1756, 2016.
- [2] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4654–4662, 2018.
- [3] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019.
- [4] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11983–11992, 2020.
- [5] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3247–3257, 2021.
- [6] Eric Brachmann and Carsten Rother. Visual camera re-localization from rgb and rgb-d images using dsac. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5847–5865, 2021.
- [7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [8] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.
- [9] Samarth Brahmabhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2616–2625, 2018.
- [10] Ming Cai, Chunhua Shen, and Ian Reid. A hybrid probabilistic model for camera relocalization. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 238. BMVA Press, 2018. URL <http://bmvc2018.org/contents/papers/0799.pdf>.
- [11] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *2021 International Conference on 3D Vision (3DV)*, pages 1175–1185. IEEE, 2021.
- [12] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022.
- [13] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016.
- [14] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5974–5983, 2017.
- [15] Mohamed Adel Musallam, Vincent Gaudilliere, Miguel Ortiz Del Castillo, Kassem Al Ismaeil, and Djamila Aouada. Leveraging equivariant features for absolute pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6876–6886, 2022.
- [16] Yoli Shavit and Ron Ferens. Do we really need scene-specific pose encoders? In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3186–3192. IEEE, 2021.
- [17] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE international conference on computer vision*, pages 627–637, 2017.
- [18] Xin Wu, Hao Zhao, Shunkai Li, Yingdian Cao, and Hongbin Zha. Sc-wls: Towards interpretable feed-forward camera re-localization. In *European Conference on Computer Vision*, pages 585–601. Springer, 2022.

- [19] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–39, 2020.
- [20] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021.
- [21] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pages 2793–2803. PMLR, 2021.
- [22] Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, pages 40770–40803. PMLR, 2023.
- [23] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. In *International Conference on Learning Representations*, 2021.
- [24] Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems*, 35:27198–27211, 2022.
- [25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.
- [26] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [27] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013.
- [28] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3302–3312, 2019.
- [29] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanculescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *5th Annual Conference on Robot Learning*, 2021.
- [30] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.
- [31] Yoli Shavit and Yosi Keller. Camera pose auto-encoders for improving pose regression. In *European Conference on Computer Vision*, pages 140–157. Springer, 2022.
- [32] Yoli Shavit, Ron Ferens, and Yosi Keller. Coarse-to-fine multi-scene pose regression with transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [34] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.
- [35] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.
- [36] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.

- [37] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, 2019.
- [38] Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, 2020.
- [39] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, pages 5562–5571. PMLR, 2021.
- [40] Jihwan Kim, Miso Lee, and Jae-Pil Heo. Self-feedback detr for temporal action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10286–10296, 2023.
- [41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

A Appendix

A.1 Training Details

We entirely follow the setting of the baseline [20]. The settings are as below.

Data Augmentation. Data augmentation is performed in the same way as [8], which is also adopted in the baseline. Afterwards, brightness, contrast, and saturation are randomly jittered for photometric robustness. Test images are also resized to 256×256 , but they are center-cropped to 224×224 , without any photometric augmentation. Additionally, following the baseline, smaller scenes in the Cambridge Landmarks dataset [8] are oversampled during training to deal with data imbalance problem.

Backbone. As did in the baseline [20], we utilize a CNN model, the EfficientNet-B0 [41], to extract local image features from input images. The image features f_t for the position and f_r for the orientation are extracted from the fourth block and the third block, respectively. Accordingly, the size of f_t is $14 \times 14 \times 112$ and the size of f_r is $28 \times 28 \times 40$, *i.e.*, the number of transformer encoder tokens N_t and N_r are 196 and 784, respectively.

A.2 Visualization

Query-Key Spaces. In addition to the statistical and quantitative results introduced in the main paper, we visualize the query-key spaces across heads and layers. Fig. A1 reports the visualization results of query-key spaces from the baseline [20] across all even layers and heads. It qualitatively shows that the distortion of query-key space occurs in most heads in transformer encoder of the baseline. In contrast, one can observe that queries and keys properly intertwine each other with our solutions as shown in Fig. A2.

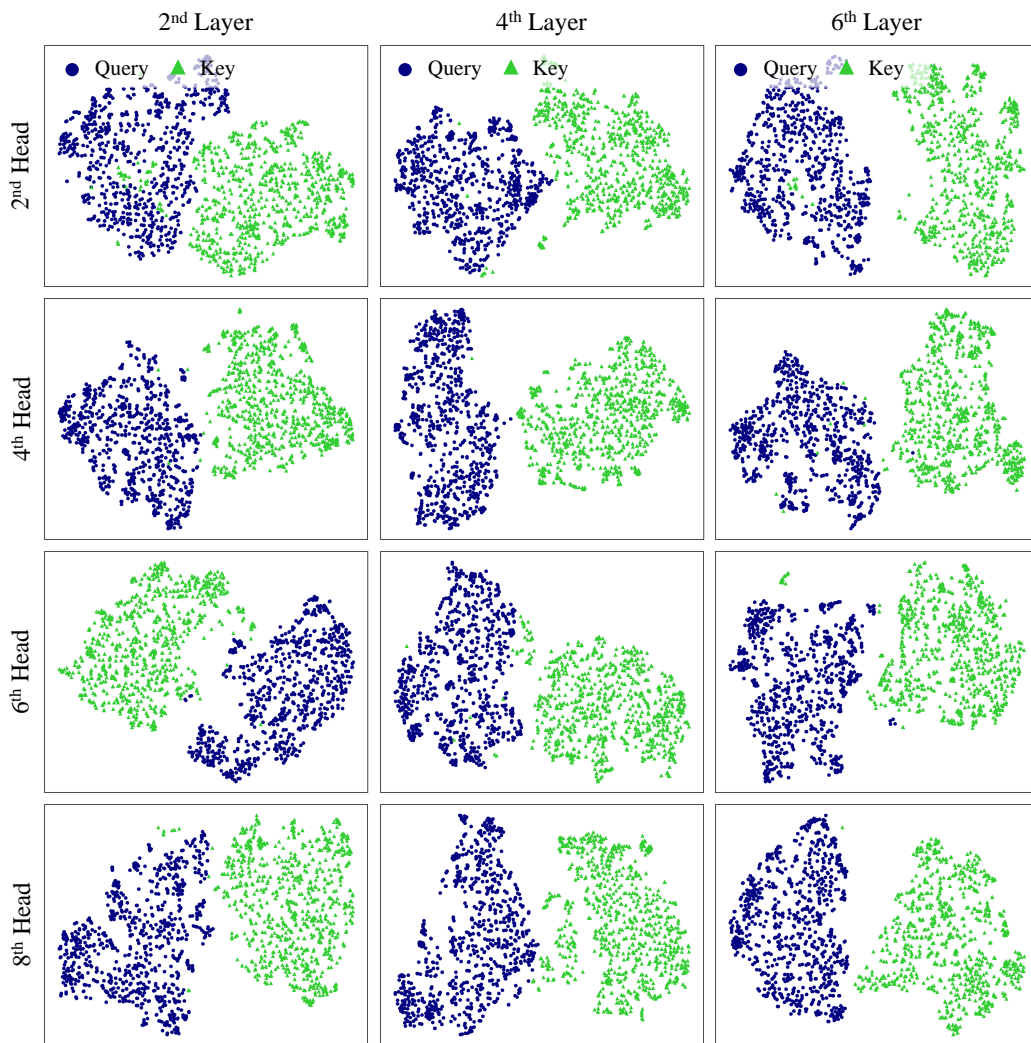


Figure A1: The figure shows the t-SNE results of query-key space from the baseline across even layers and heads. Overall, query and key regions are separate while a small subset of keys are blended into the query region.

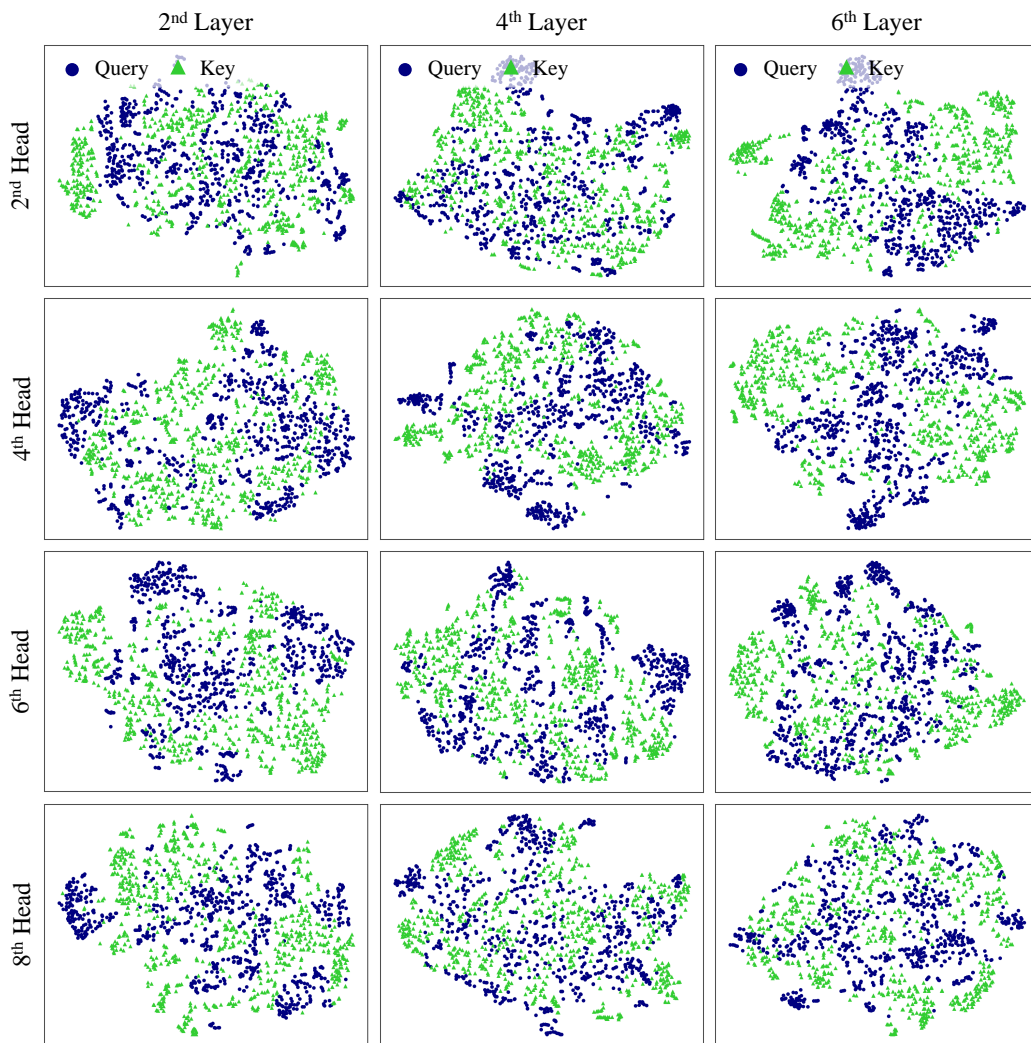


Figure A2: The figure shows the t-SNE results of query-key space from the model with our solutions across even layers and heads. The problem we point out is resolved with our solutions; similar query subsets and key subsets are grouped together.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We explain the paper's contributions and scope accurately in the abstract and Sec. 1.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitation in Sec. 8.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We explain all the formulas in Sec. 4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We explain the exact method in Sec. 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include the code in the supplemental materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify all details in Sec. 6 and supplemental materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Since the train dataset and test dataset used for evaluation are completely separated, there are no issues arising from data splitting. For the fair comparison, our experiments follow the exact same configuration including random seeds as the baseline. The configuration and the code is included in the supplemental materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We specify the compute resources in Sec. 6.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: We conducted the research with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the broader impacts in Sec. 8.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This work uses public datasets and poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We checked the license of our baseline code and datasets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This work does not provide new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This work does not contain crowdsourcing experiments and research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This work does not need study participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.