CWPFormer: Towards High-performance Visual Place Recognition for Robot with Cross-weight Attention Learning

***Abstract*—As an important component of robot localization and navigation, visual place recognition (VPR) has made significant improvements in the past few decades. Most existing state-of-the-art VPR methods face the following challenges: 1) Aiming only for performance with ideal conditions while overlooking real-world conditions; 2) The existing VPR paradigm, struggling to reconcile the transfer gap between upstream pre-training and downstream fine-tuning; 3) Deeper networks produce high-dimensional parameters in the model training that results in lower efficient of models. To address those problems, we propose a high-performance visual place recognition framework for robot navigation tasks. Concretely, our framework is composed of three major modules: 1) based on vision transformer (ViT), we design a siamese Cross-weight Pyramid Transformer (CWPFormer) backbone for image feature extraction. First, we integrate feature reconstruction and content cognition by inserting a feature pyramid phase into pre-training. Second, we establish weight correlation and sharing between feature downsampling and upsampling that offers multi-stage supervision to fine-tuning. 2) We found that the attention map has high similarity between heads, and the high-dimensional data processing based on the ViT leads to computational redundancy. To cope with this problem, we present a cascaded hash attention (CHA) module to feed the hash attention head with different complete feature splits, which not only saves the computational cost but also improves the attention diversity. 3) Besides, we adopt a Bayesian learning scheme with a dynamically constructed similarity matrix to learn one-dimensional compact hash codes to improve recognition accuracy. Exhaustive experiments demonstrate the superiority of our proposed VPR approach on datasets and real-world environments. Our code is available: https://github.com/CV4RA/CWPFormer.**

***Impact Statement*-This paper is inspired by the loop-closure detection problem in mobile robot localization and navigation, but it applies to all perception-based unmanned systems. Typical examples are autonomous vehicles, unmanned aerial vehicles, etc. Existing state-of-the-art approaches only focus on improving the model's performance and ignore the computational complexity. As a result, these algorithms can only stay in the offline verification stage and are difficult to effectively apply in real-world environments. In addition, most existing methods obtain redundant information by stacking the network deep to improve performance, which imposes a heavy computational burden on hardware. This paper presents a cascaded hash attention module that feeds the hash attention head with different complete feature segmentations, which not only saves computational costs but also improves attention diversity. This paper also proposes to establish a weight correlation and sharing mechanism between feature downsampling and upsampling, which provides multi-stage supervision for fine-tuning, further reduces network parameters, and improves efficiency. Preliminary real-world experiments show that this approach is feasible, but it does not yet operate in fully lowlight environments. In future research, we will solve the problem of place recognition in completely low-light environments and try to use knowledge distillation to improve the generalization ability of the model.**

***Index Terms*—Visual place recognition, Cross-weight pyramid transformer, Hash code, Bayesian learning network, Multi-head attention.**

I. Introduction

As one of the core components of robot navigation and the fundamental tasks of computer vision, VPR has attracted much attention and has yielded many applications in recent years [1]. VPR attempts to help a robot or autonomous unmanned system determine whether a location is one it has visited before, which is an essential and challenging problem in the field of autonomous robots. Over the past decade, these fields have witnessed a surge in the use of VPR in applications such as simultaneous localization and mapping (SLAM), autonomous drone patrols, and autonomous vehicle driving. For example, the main function of VPR is to perform closed-loop detection in robot localization and navigation, which can not only reduce localization errors caused by the visual odometer (VO) but also avoid establishing blurry maps of unknown environments.

Deep learning-based VPR aims to find similar places that robots have visited from a large-scale dataset against a query scene. Generally, the similarity between the salient features of the query scene and the reference images is used to rank the images in the dataset. The highly ranked image is considered to be in the same place as the current query image. Traditionally, various handcrafted methods have been used for VPR mainly by using simple visual cues such as color, texture, shape, etc [2]. However, traditional handcrafted features often fail to represent the scenes due to the lack of semantic representation. In recent years, with the rapid development of deep learning technology, visual features based on deep learning are regarded as a feasible alternative to manual features in a variety of visual detection tasks such as place recognition, object detection [3], face recognition [4], etc. To further boost the place recognition performance, some works have been devoted to exploiting the ViT backbone for visual feature extraction. Transformer is a network architecture of feature extraction based on a self-attention mechanism, which can perform attention computing for each position in the input sequence, so as to extract the global context information in the image.

Although the ViT pipeline has made breakthroughs in many aspects, there is also a key problem, namely: the transfer gap between upstream pre-training and downstream fine-tuning in model training. From this perspective, we believe that downstream visual recognition, especially fine-scale recognition, requires hierarchical visual features. However, most

existing pre-training tasks are built on the general ViT. Even though hierarchical visual Transformers are already in use, e.g., feature pyramid Transformers, the pre-training task only affects the backbone but leaves the neck untrained, which introduces additional risk to downstream fine-tuning as an optimization starts with a randomly initialized neck that does not guarantee to match the pre-trained backbone [5]. To solve this problem, we construct the CWPFormer model, which combines feature reconstruction and content recognition by inserting feature pyramids in pre-training and establishes weight correlation and sharing between feature downsampling and upsampling to provide multi-stage supervision for refined sampling. In addition, We discover that attention maps have high similarity across different heads, and deeper transformer training produces a large number of parameters, resulting in computational redundancy. To solve this problem, we design a CHA module to feed the attention head with different complete feature segmentations and embed hash encoding into the attention module, which not only saves computational cost but also improves attention diversity.

The main contributions of this paper can be summarized as follows.

1) To handle the coordination and matching of upstream pre-training and downstream fine-tuning in model training, we propose a siamese CWPFormer, which combines feature reconstruction and content recognition by inserting feature pyramids in pre-training and establishes weight correlation and sharing between feature downsampling and upsampling to provide multi-stage supervision for refine sampling.

2) To improve efficiency, we integrate hash coding into the attention model and propose a CHA module to feed the attention head with different complete feature segmentations and embed hash encoding into the attention module, which not only saves computational costs but also improves attention diversity.

3) To improve recognition accuracy, we adopt a Bayesian learning scheme with a dynamically constructed similarity matrix to learn one-dimensional compact hash codes for similarity score and ranking.

II. Relate work

A. Feature representation methods for visual place recognition

Feature representation plays a key role in VPR and can generally be divided into two categories: local representation and global representation. Local descriptors such as traditional hand-crafted features and more recent deep learning-based features, can alternatively be thought of as keypoint or region descriptors. Local descriptors can be used to obtain global descriptors through aggregation operations, and can also be used for cross-matching between image pairs. Chen et al. [6] delved deeper into the internal structure of CNNs and proposed new CNN-based image features for VPR. To better learn prior knowledge for VPR, patch-NetVLAD [7] adopts optimization aggregation technology (NetVLAD [8]) that directly extracts multi-scale patch features from the global descriptor for VPR. Global descriptors are used to condense images into compact representations that are robust to changes in appearance, illumination, and viewpoint for large-scale VPR. Currently, almost

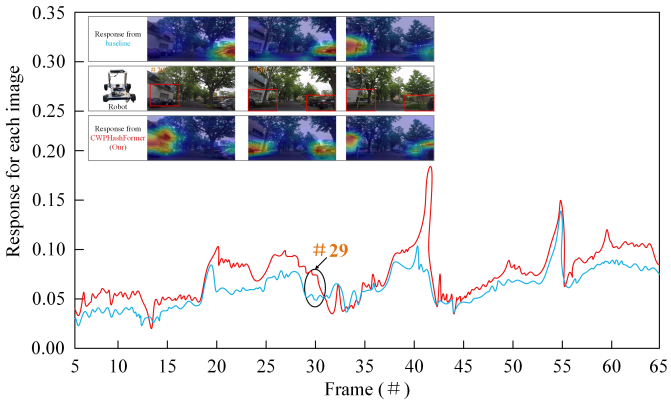


Fig. 1: Qualitative evaluation of the proposed CWPFormer (in red) and its baseline (in blue) on kitti_06 with similar salient features around. The red boxes denote salient regions for ground truth. The red and blue curves represent the changes in responses based on the proposed CWPFormer and its baseline from frame 5 to frame 65.

all global descriptor-based visual tasks are based on CNN or ViT structures. For example, Lin et al. [9] proposed a nested invariant pooling method for video matching, localization, and retrieval. Ma et al. [10] proposed a transformer-based network that exploits the spatiotemporal information provided by the sequence of range images to generate global descriptors for each sequence in an end-to-end manner.

Existing pyramid structures have been widely explored in the computer vision field to enhance multi-scale feature representation. However, traditional pyramid structures often lack effective control of feature flow between downsampling and upsampling stages, especially between pre-training and fine-tuning phases, which can lead to instability due to random initialization. We propose a pyramid structure with a cross-weight sharing mechanism, which ensures consistency in feature flow between the upsampling and downsampling stages by sharing weights. This approach effectively reduces the transfer gap between pre-training and fine-tuning, mitigating instability caused by random initialization. This design is not extensively explored in prior research.

B. Feature pyramid in vision tasks

The use of a single high-level feature representation has an obvious drawback, that is, small objects themselves have less pixel information, which is easily lost in the process of downsampling. The network structure of the feature pyramid can deal with the multi-scale change problem in object detection without increasing the amount of computation. Wang et al. [11] proposed a versatile backbone for dense prediction without convolutions, which overcomes the difficulties in porting transformers to various dense prediction scenarios. Zang et al. [12] proposed a multi-scale pyramid Transformer network for video-based pedestrian retrieval. To overcome the inherent locality of CNN, Wang et al. [13] proposed a hybrid CNN-Transformer feature extraction network to model image context information for VPR. The presence of distracting

elements in complex scenes often leads to biases in visual place perception. To solve this problem, Wang et al. [14] proposed a global VPR model to integrate information from task-relevant regions into image representations.

Existing hash attention mechanisms have been introduced in some studies, primarily to reduce the computational complexity of high-dimensional features. However, these methods often encounter redundancy between attention heads in deeper Transformer networks, which can affect computational efficiency and overall performance. Our CHA module enhances attention diversity by feeding different complete feature splits into each attention head, reducing redundancy between the heads. This design improves computational efficiency and increases the distinction between different attention heads. This advancement is particularly important in deep Transformer networks and represents a significant improvement in our work.

C. Deep hashing network

By mapping data into binary code through a machine learning mechanism, hash coding can significantly reduce data storage and communication overhead, thereby effectively improving the efficiency of the learning system. Therefore, recent years have witnessed more large-scale models using hash coding to encode high-dimensional image data into one-dimensional binary codes, which greatly improves model efficiency. Chen et al. [15] presented a pure transformer-based framework for deep hashing learning for efficient image Retrieval. Li et al. [16] took the ViT as the backbone and binary code as the intermediate representation and proposed a transformer-based image retrieval method for image Retrieval.

Existing contrastive loss is widely used in similarity learning tasks to improve the model's ability to distinguish between similar features, but it lacks flexibility when dealing with diverse input data. We introduce a dynamically constructed similarity matrix within the Bayesian learning framework, enabling the model to adaptively score the similarity between input data. Compared to traditional contrastive loss, this method offers greater flexibility and better adaptation to variations in input data, thus improving recognition accuracy in challenging scenarios.

III. The Proposed Approach

In this section, we will introduce the overall structure and workflow of the proposed method in detail, as shown in Fig. 2.

A. Cross-weight pyramid Transformer (CWPFormer)

We adopt a local coarse-to-fine feature encoding method (coarse encoding: Up and down sampling; fine encoding: feature reconstruction) to train the model so that it can fully perceive every pixel of the image and achieve accurate extraction and encoding of salient features. Downsampling of the model enables coarse perception of images and model pre-training while upsampling enables fine-grained perception and fine-tuning. Our CWPFormer consists of a set of sub-blocks, which are stacked by multi-channel attention layers (MCAL) (see Fig. 2(b)) that form up-and-down sampling. The MCAL block is composed of a backbone, neck, and self-attention head, and is presented separately through the formulation: $f(x;\alpha)$, $g(x;\beta)$, and $h(x;\gamma)$, where x represents the input of model, α, β, and γ represent learnable parameters. Therefore, The function can be denoted as: $h(g(f(x;\alpha);\beta);\gamma)$. We pre-train the model at the downsample stage, and fine-tuning the model at the upsample stage. To reduce computational complexity, weights are shared between the two stages. Also, the two stages share the same skeleton but have different necks and heads. Pretraining and fine-tuning can be formulated as:

$$\min \mathbb{E}_{\mathcal{D}^{\mathrm{p}}} \| x_n^p - h^p(g^p(f^p(x_n^p;\alpha);\beta^p);\gamma^p) \tag{1}$$

$$\min \mathbb{E}_{\mathcal{D}^{\mathrm{f}}} \| x_m^f - h^f(g^f(f^f(x_m^f;\alpha);\beta^f);\gamma^f) \tag{2}$$

We argue that such a pipeline results in a significant transfer gap between pre-training and fine-tuning, bringing double drawbacks when the parameters between β^p, β^f and γ^p, γ^f are not shared. Firstly, the extraction of the backbone parameters for multi-level features is not optimal. Secondly, the training process may be slowed down and unsatisfactory recognition results may result from the fine-tuning stage's optimization of a randomly initialized β^f and γ^f. To bridge the gap, according to [17], we construct a general framework that combines $g^p(x)$ and $g^f(x)$, which allows for the easy reuse of pre-trained β^p to initialize fine-tuned β^f, resulting in the random initialization of only γ^f. The overall pre-training to fine-tuning framework is illustrated in Fig. 2(b).

Our CWPFormer contains n stages and each stage has a CWPFormer subblock. During the whole training process, the backbone first downsamples the input image and will produce $n+1$ feature maps:

$$f(x;\alpha) = E^0, E^1, ..., E^n \tag{3}$$

Where E^0 represents a direct embedding of the input, and the superscript corresponds to the stage layer of the input. Each feature map includes a set of tokens, $E^n = e_1^n, e_2^n, ..., e_k^n$, where k represents the number of tokens in the s^{th} stage.

In our work, $g^p(x)$ and $g^f(x)$ share the same structure and training parameters as they all start with E^n and gradually aggregate it with lower-level features. The process can be written as follows:

$$\begin{aligned} &\mathbf{U}^k = \mathbf{E}^k, k = 1 \\ &\mathbf{U}^k = \mathbf{E}^k + g^k\left(\mathbf{E}^{k+1};\beta^k\right), \quad 1 \leqslant k < n \end{aligned} \tag{4}$$

Where g^k upsamples U^k to keep the resolution consistent with downsampling E^k. We define the learnable parameters as consisting of a layer-wise set $\{\beta^n\}$. As these parameters learned from pre-training are reused in fine-tuning, the transmission gap is reduced to a large extent such that the only module that remains independent between pre-training and fine-tuning is the heads.

We reconstruct the feature maps from pre-training and fine-tuning by constructing a reconstruction loss from MCAL: $\|x^0 - h^{p,0}(U^0;\gamma^{p,0})\|$, where $h^{p,0}$ involves reconstructing several MCAL blocks of the original image from $U = \sum_{k=1}^{k=n} U^k$. Each reconstruction process relies on a reconstruction head

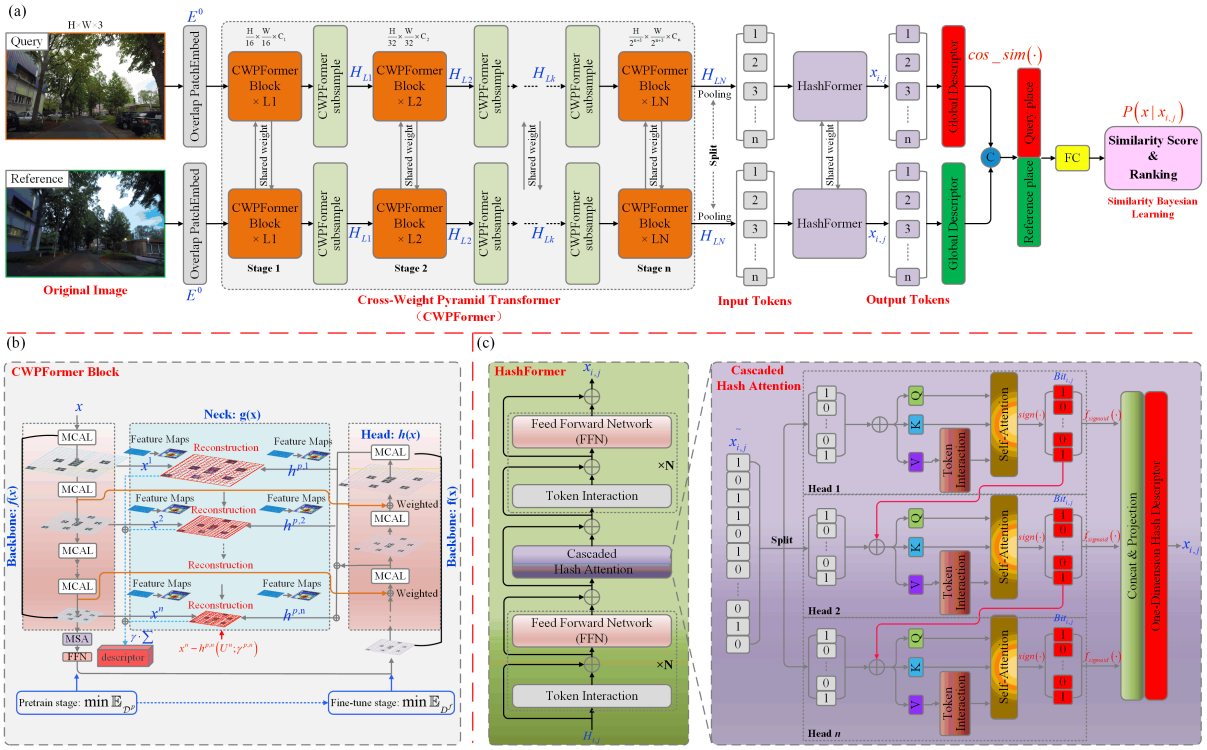


Fig. 2: Overview of the our end-to-end VPR framework. It consists of three modules, i.e., CWPFormer module, HashFormer module, and similarity Bayesian learning module. The three modules are seamlessly connected in an end-to-end trainable manner.

implementation to obtain multi-scale features. The multi-scale loss function can be constructed as follows:

$$L = \|x^0 - h^{p,0}(U^0;\gamma^{p,0})\| + \gamma \cdot \sum_{k=1}^{k=n} \|x^k - h^{p,k}(U^k;\gamma^{p,k})\| \quad (5)$$

Where $\|x^0 - h^{p,0}(U^0;\gamma^{p,0})\|$ is the image reconstruction, and $\sum_{k=1}^{k=n} \|x^k - h^{p,k}(U^k;\gamma^{p,k})\|$ is the feature reconstruction. x^k represents the output at the k^{th} decoder stage and γ is related to the validation set and is set to 0.25, in particular, x^0 represents the input to the decoder, also can be taken as the output of encoder.

At the stages of pre-training and fine-tuning, we use MCAL as the basic backbone for feature encoding, as shown in Fig. 3. Each encoder layer starts with a self-attention layer that exploits each channel to learn contextual relationships within a single channel. We use multi-head scale dot product attention (MH-SDPA) [20] as the scoring function to calculate attention weights across time. Given the i^{th} channel embedding x_i, we can obtain the query, key, and value for channel-wise self-attention by a linear transition, and then the activation function can be written as:

$$\begin{aligned} Q_i^{cw} &= \sigma\left(\tilde{x}_i W^{cw,q} + \mathbf{1}\left(\mathbf{b}_i^{cw,q}\right)^T\right) \\ K_i^{cw} &= \sigma\left(\tilde{x}_i W^{cw,k} + \mathbf{1}\left(\mathbf{b}_i^{cw,k}\right)^T\right) \\ V_i^{cw} &= \sigma\left(\tilde{x}_i W^{cw,v} + \mathbf{1}\left(\mathbf{b}_i^{cw,v}\right)^T\right) \end{aligned} \quad (6)$$

Where $\tilde{x}_i$ is the i^{th} channel feature matrix that contains magnitude features x_i^m and phase features x_i^p, W_i^{cw} and b_i^{cw} are learnable weight and bias parameters for the channel-wise self-attention, σ is the action function. Then, we adopt three linear projection layers (w_i^m, w_i^p, w_i^j) to embed the magnitude, phase, and their concated embeddings. The whole embedding process is as follows:

$$\tilde{x}_i = [x_i^m \cdot w_i^m, x_i^p \cdot w_i^p, x_i^j \cdot w_i^j] + PE_{embedding} \quad (7)$$

Then, the output of channel-wise self-attention is computed by the following [18]:

$$H_i = Softmax\left(\frac{Q_i^{cw}(K_i^{cw})^T}{\sqrt{d_m}}\right)V_i^{cw} \quad (8)$$

Where d_m is scale factor. We then add residual connections and layer norm to obtain the final output of channel-wise attention $\tilde{H}_i$ through feed-forward layers.

The cross-channel attention layer not only learns the temporal interrelationships between time frames but also learns the interrelationships between self-attention channel representations. we can obtain the query, key, and value for cross-channel self-attention by a linear transition, and then the activation function can be written as:

$$\begin{aligned} Q_i^{cc} &= \sigma\left(\tilde{H}_i^{cw} W^{cc,q} + \mathbf{1}\left(\mathbf{b}_i^{cc,q}\right)^T\right) \\ K_i^{cc} &= \sigma\left(\tilde{H}_j^{cc} W^{cc,k} + \mathbf{1}\left(\mathbf{b}_i^{cc,k}\right)^T\right) \\ V_i^{cc} &= \sigma\left(\tilde{H}_j^{cc} W^{cc,v} + \mathbf{1}\left(\mathbf{b}_i^{cc,v}\right)^T\right) \\ H_j^{cc} &= \sum_{j,j\neq i} A_j \odot \tilde{H}_j^{cw} \end{aligned} \quad (9)$$

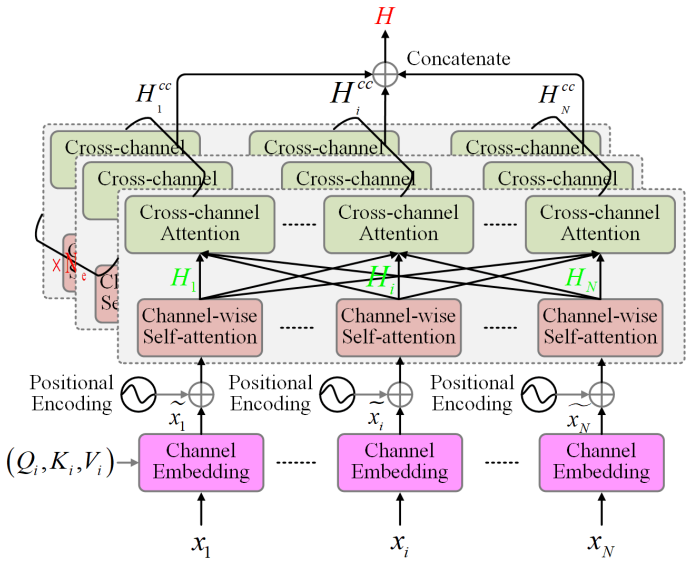


Fig. 3: Overview of our used multi-channel attention layer. Where N and N_e are the number of attention channels and the number of encoder layers, x_i represents the i^{th} input image sequence.

Where $\tilde{H}_i^{cw}$ represents the input, and W_i^{cc} and b_i^{cc} are learnable weight and bias parameters for the cross-channel self-attention. Then, the output of cross-channel self-attention is computed by the following:

$$H_i^{cc} = Softmax(\frac{Q_i^{cc}(K_i^{cc})^T}{\sqrt{d_m}})V_i^{cc} \tag{10}$$

Last, we can obtain the final multi-channel attention output by the following concatenation operation:

$$H_{Lk} = H_1^{cc,k} \oplus H_2^{cc,k} \oplus, ..., \oplus H_N^{cc,k} \tag{11}$$

Where "$\oplus$" is the concatenation of multiple features of the same dimension, L_k is the k^{th} encoding stage of CWPFormer.

B. Cascaded hash attention (CHA)

Inspired by cascaded group attention (CGA) [19], we propose a highly efficient attention module named cascaded hash attention (CHA) for feature encoding. CHA encodes high-dimensional visual descriptions into one-dimensional binary hash codes and then processes them through MHA to achieve efficient feature encoding. CHA also provides a different split of the complete features for each head, thus explicitly decomposing the attention computation across heads. Formally, this attention can be written as:

$$\begin{aligned} \tilde{x}_{i,j} &= \text{Attn}\left(x_{i,j}W_{i,j}^{\text{q}}, x_{i,j}W_{i,j}^{\text{k}}, x_{i,j}W_{i,j}^{\text{v}}\right), \\ \widetilde{x}_{i+1} &= \text{Concat}\left[\widetilde{x}_{i,j}\right]_{j=1:h} W_i^{\text{P}} \end{aligned} \tag{12}$$

Where $\tilde{x}_{i,j}$ is the i^{th} split input feature by j^{th} head computing, $W_{i,j}^{\text{q}}$, $W_{i,j}^{\text{k}}$, and $W_{i,j}^{\text{v}}$ represent projection layers that map the split input into different subspaces, W_i^{P} is a linear layer that projects the output features into same dimensions with the input, h is the number of attention head.

Then, we generate the hash codes through the following steps:

- Linear Transformation:

$$H_{i,j} = w_{i,j}\tilde{x}_{i,j} + b_{i,j} \tag{13}$$

  where $w_{i,j}$ and $b_{i,j}$ are the weights and biases of the linear transformation.
- Sign Function:

$$Bit_{i,j} = sign(H_{i,j}) \tag{14}$$

  Where the sign function maps positive values of $H_{i,j}$ to "+1" and negative values to "-1", thereby generating the discrete hash codes $Bit_{i,j}$.

Assuming that the hash code length of each place image in the matching stage is b bits, then for a feature vector with a token embedding size of d, a b-bit global hash vector can be encoded as:

$$x_{i,j} = f_{sigmoid}(z_{global}(Bit_{i,j})U_{i,j} + B_{i,j}) \tag{15}$$

Where $f_{sigmoid}$ is the activation function, z_{global} is the hash encoding function, $U_{i,j}$ and $B_{i,j}$ are the weight matrix and the set of bias parameter.

C. Place matching with similarity Bayesian learning

We use cosine similarity to measure the similarity between images. We project the image representation into a vector space and then determine the degree of similarity between the two images based on the distance between them. The cosine similarity can be computed by the following:

$$cos\_sim(x_{que}^{'}, x_{ref}^{'}) = \frac{x_{que}^{'} \cdot x_{ref}^{'}}{\|x_{que}^{'}\| \cdot \|x_{ref}^{'}\|} \tag{16}$$

Where $x_{que}^{'}$ and $x_{que}^{'}$ are feature vectors respectively from the query dataset and the reference dataset.

Generally, the smaller the distance between two place vectors, the more similar they are. Therefore, we can transform the process of solving the minimum distance probability into the cosine similarity probability problem through the softmax function: Given a training dataset $\{x(x_{que}, x_{ref}), x_{i,j}(x_{i,j}^{que}, x_{i,j}^{ref})\}$, we first obtain the output $\tilde{x}(x_{que}, x_{ref})$ through the attention mechanism, and encode the input features into hash codes $x_{i,j}$, then compute the probabilities for each sample pair:

$$P(x(x_{que}^{'}, x_{ref}^{'})|x_{i,j}) = \frac{exp(cos\_sim(x_{que}^{'}, x_{ref}^{'}))}{\sum exp(cos\_sim(x_{que}^{'}, x_{ref}^{'}))} \tag{17}$$

The likelihood function for the entire dataset is the joint probability of all sample pairs:

$$L_{likelihood} = \prod(P(x(x_{que}^{'}, x_{ref}^{'})|x_{i,j})) \tag{18}$$

Typically, we use the negative log-likelihood as the loss function for optimization:

$$L_{nll} = -\sum(log\{P(x(x_{que}^{'}, x_{ref}^{'})|x_{i,j})\}) \tag{19}$$

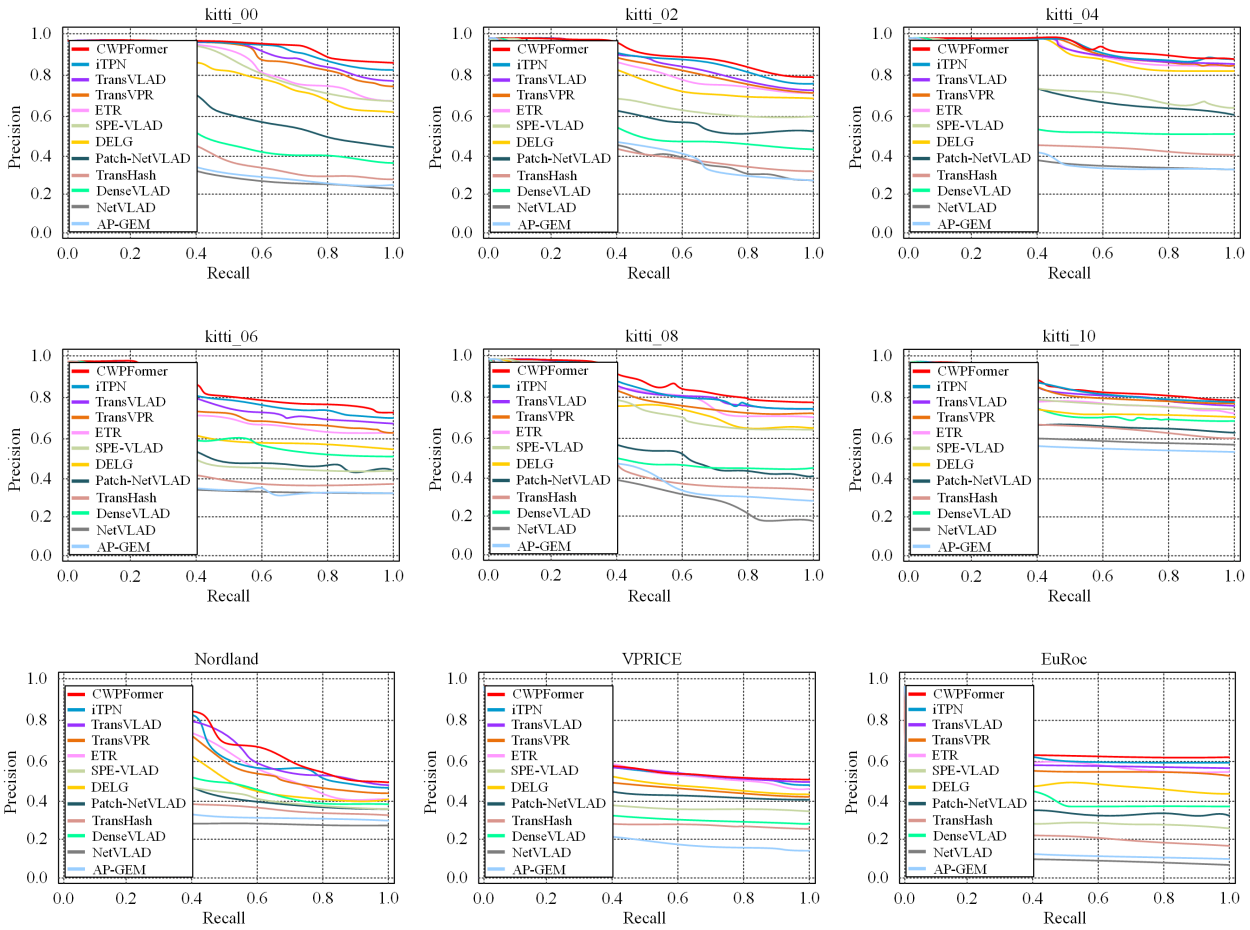


Fig. 4: Comparison of our CWPFormer versus state-of-the-art methods.

IV. Experiments

In this section, the proposed CWPFormer is evaluated on four public VPR benchmarks, i.e., KITTI (00-08) [21], Nordland [22], VPRICE [23], and EuRoC [24], and also tested on challenging real-world VPR environments.

A. Implementation Details

The CWPFormer is pre-trained based on the ImageNet-1K dataset that includes 1.28M images for training. Class labels are not used in the pre-training stage. Before training, each image is resized as 224×224 and divided into 14 patches, each patch size is 16 pixels. The AdamW optimizer with an initial learning rate of 1.2^{-4}, weight decay of 0.05, and batch size of 128 is used for the model optimization. We unify local and global feature extraction into a CWPFomer and extract global descriptors at three scales with a dimensionality of 1024. We extract local descriptors at seven scales (from 0.2 to 1.4) with a dimensionality of 64. We set the number of MHA is set as 8 for comparison experiments with state-of-the-arts. To guarantee the objectivity of the results, we run all models on the same hardware platform (NVIDIA 12G-3060 GPUs).

B. Evaluation Criteria

In our work, we mainly adopt the precision-recall curve to evaluate the performance. The former indicates the degree of accuracy with which a positive sample is predicted, while the latter indicates the proportion of the total positive sample that is predicted to be positive. Note that we use the Top-N as the metric to rank images for image retrieval. "Top-N" refers to the top N classes with the highest probability of predicted output. As long as the correct class appears, the sample is regarded as the correct prediction.

C. Offline evaluation

1) Comparison with state-of-the-art methods: For a more comprehensive comparison of offline evaluation, we also divide these VPR methods into two groups: CNN-based (NetVLAD [8], AP-GEM [25], DenseVLAD [26], Patch-NetVLAD [7], DELG [27], and SPE-VLAD [28]) and Transformer-based VPR methods (TranHash [15], ETR [29], TransVPR [14], TransVLAD [30], iTPN [31], and our CWPFormer). We report the results in Fig. 4 and Table I which shows that our CWPFormer achieves the best performance on almost all four datasets compared to state-of-the-art methods. It is worth noting that ViT-based methods are generally better

TABLE I: Offline evaluations compared to state-of-the-art methods on four datasets (KITTI contains nine sequences). All methods involved in the comparison rank topn (n = 1 and n = 5) places through the global feature methods. The best performances in CNN methods and ViT methods are respectively highlighted with green and red colors.

	Methods	KITTI_00		KITTI_02		KITTI_04		KITTI_06		KITTI_08		KITTI_10		Nordland		VPRICE		EuRoc	
		Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
CNN	AP-GEM	48.6	52.4	47.9	51.5	50.2	54.3	53.3	57.4	54.4	58.1	47.3	50.5	11.6	13.4	48.7	55.3	45.5	52.4
	DenseVLAD	50.2	54.8	51.6	55.3	51.5	57.8	55.6	59.8	56.6	59.9	49.8	56.1	12.8	21.2	48.8	56.9	56.2	52.4
	NetVLAD	55.4	62.8	55.9	63.3	57.4	64.1	56.9	63.1	58.7	64.6	58.2	62.9	13.4	21.7	50.2	60.7	58.6	53.5
	Patch-NetVLAD	73.2	79.8	66.5	71.8	67.9	73.0	67.8	72.1	68.4	73.2	67.7	76.8	50.6	58.9	79.5	83.4	77.3	81.0
	DELG	**75.4**	**81.4**	**69.6**	**73.5**	**70.4**	**74.9**	**68.4**	**75.0**	**70.8**	**75.3**	**67.9**	**77.8**	**58.7**	**64.5**	**81.6**	**85.5**	**77.6**	**82.1**
	SPE-VLAD	64.2	69.1	66.1	71.4	66.9	72.4	65.3	71.6	65.7	72.3	60.2	64.1	18.6	23.2	55.6	64.5	60.9	66.5
	TransHash	72.5	77.9	65.4	70.8	67.0	71.6	67.2	71.5	68.2	72.5	67.1	75.9	48.8	59.1	77.5	81.7	75.9	80.8
	ETR	78.5	82.4	70.8	74.2	72.6	76.9	72.1	75.6	72.4	76.5	73.8	78.4	51.4	60.0	75.3	80.4	76.2	82.8
ViT	TransVPR	81.5	86.8	80.4	85.2	80.6	86.4	79.8	85.8	80.1	85.4	75.2	80.1	72.6	78.7	84.9	88.7	81.5	85.4
	TransVLAD	83.4	87.5	81.7	86.5	82.5	87.3	81.2	86.9	81.5	87.4	76.5	81.8	73.4	80.4	84.3	89.0	81.7	89.4
	iTPN	84.7	87.8	82.3	87.2	82.9	88.1	**81.7**	87.4	82.3	87.9	78.4	83.9	75.8	83.5	85.5	89.6	83.3	88.2
	CWPFormer (our)	**85.2**	**88.7**	**82.9**	**87.8**	**83.4**	**88.4**	**81.7**	**88.5**	**82.9**	**88.7**	**79.1**	**85.2**	**76.2**	**84.8**	**85.8**	**90.3**	**84.0**	**89.3**

TABLE II: The training parameters are evaluated based on an NVIDIA 12G-GeForce RTX 3060 GPU on the Nordland dataset.

Method	Architecture Type	Training Parameters (M)	Training Time (h)	Extraction latency (ms)	Ranking latency (ms)	Compressed with Vector
AP-GEM	CNN + Aggregation	25.8	21.3	147	46	19.8
DenseVLAD	CNN + Aggregation	33.7	34.4	241	37	32.5
NetVLAD	CNN + Aggregation	25.5	26.3	232	22	24.3
Patch-NetVLAD	CNN + Local Aggregation	60.2	45.4	485	98	43.5
DELG	CNN + Global Features	72.3	54.8	148	74	1.4
SPE-VLAD	CNN + Weighted Aggregation	35.7	33.9	267	42	16.8
TransHash	Transformer + Hashing	65.2	35.2	13	8	0.9
ETR	Transformer	109.4	53.5	145	11	1.6
TransVPR	Transformer + Multi-layer Attention	120.8	59.8	38	6	37
TransVLAD	Transformer + Aggregation	100.6	56.7	19	86	34
iTPN	Transformer + Pyramid Structure	110.4	65.5	16	25	68.4
CWPFormer (our)	Transformer + Pyramid Structure + Hashing	101.6	46.7	17	8	1.4

than CNN-based methods, but some CNN models can significantly improve performance by combining some advanced aggregation technology such as embedded VLAD layer or feature patch operation, and can even surpass ViT on some datasets. For example, In addition to losing to TransHash when comparing top-1 performance on KITTI_00, Patch-NetVLAD has taken the lead regardless of comparing top-1 or top-5 on other datasets or sequences. DELG achieves the best results among CNN methods, mainly due to its unification of global and local features into a deep model, accurate retrieval through efficient feature extraction, and the use of reranking in the backend. iTPN achieves results close to our method in the ViT methods mainly because it also uses a cross-weight pyramid Transformer model and integrates masked features in model pre-training and fine-tuning.

The significant advantage of CWPFormer lies in its efficient feature compression through hash encoding, reducing storage requirements to just 1.4MB, which is much lower than other methods like TransVPR (37MB) and iTPN (68.4MB), making it highly suitable for memory-constrained devices, which is shown in Table II. Hash encoding not only minimizes storage usage but also optimizes computational efficiency, with a feature extraction latency of 17ms and ranking latency of 8ms, delivering excellent performance in real-time applications. Although the initial parameter count is 101.6M, CWPFormer shortens training time to 46.7 hours compared to other complex Transformer models due to reduced computational redundancy. Overall, CWPFormer strikes a good balance between efficient storage and fast computation, making it an ideal choice for VPR tasks that require high performance and resource optimization.

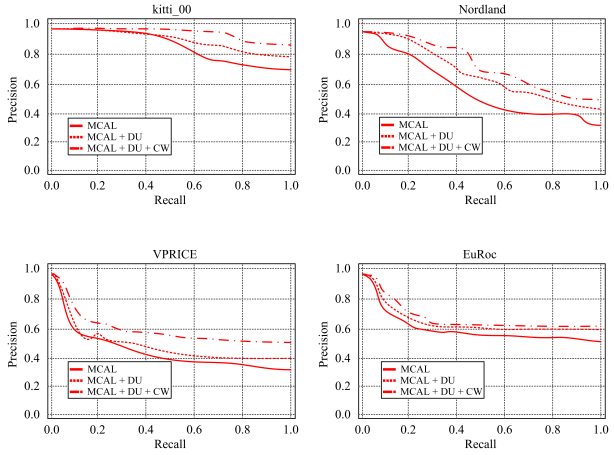


Fig. 5: Ablation studies with different encoding methods in the pyramid Transformer module.

2) Ablation study: We perform ablation studies with different encoding methods in the pyramid Transformer module. We use our model with the pyramid Transformer and tested on the KITTI_00, Nordland, VPRICE, and EuRoc datasets. Fig. 5 shows our model with downsampling, upsampling, and cross-weight pre-training (MCAL + DU + CW) achieves the highest level of precision against the models with the simple pyramid Transformer (MCAL), and also be superior to the DU (downsampling and upsampling) pyramid Transformer.

We also perform ablation studies with different ranking methods on the KITTI_00, Nordland, VPRICE, and EuRoc datasets. Table III shows our model with a similar Bayesian

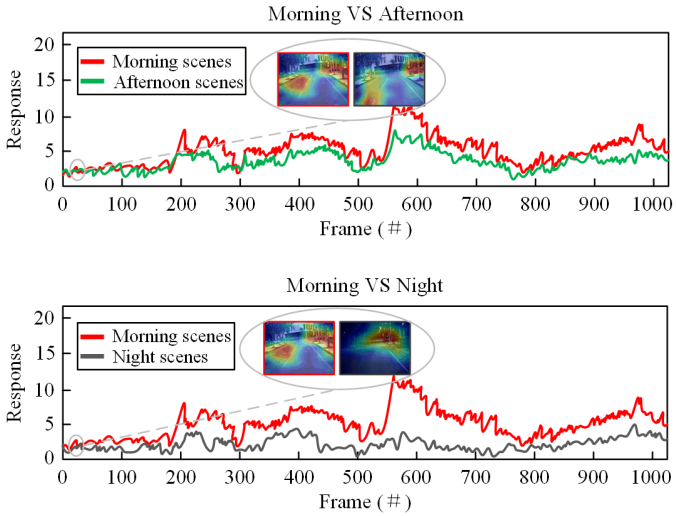


Fig. 6: Results of real-world tests on the robot platform embedded Jetson Xavier NX. The response values at morning, afternoon, and night scenes are marked with red, green, and grey colors.

TABLE III: Ablation studies with different ranking methods in the CWPFormer.

Ranking methods	KITTI_00		Nordland		VPRICE		EuRoc	
	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
Softmax + Cosine	83.8	87.3	75.9	83.2	85.1	89.4	83.6	88.5
Softmax + + Cosine + Bayesian	85.2	88.7	76.2	84.8	85.8	90.3	84.0	89.3

learning network is superior to the one without Bayesian learning.

D. Online evaluation

As discussed before, the robot place recognition performance in real-world scenes is influenced by illumination, Therefore, to comprehensively and realistically evaluate the place recognition performance of robots in real scenarios, We conduct robot place recognition experiments with three different lighting conditions (the morning, evening, and night), and record the distribution of the sum of salient feature response values in each frame by the proposed CWPFormer, which is shown in Fig. 6. In Fig. 6, we can intuitively see that the salience responses between the scene of the same place in the morning and evening are closer than that in the morning and night, indicating that the robot is more likely to successfully identify the place when it visits the same place again in the afternoon with better lighting conditions.

Fig. 7 shows an example of successful place recognition in real-world environments of robot tests. Based on our ranking method, we list the top five images that are similar to the reference image, and place images are considered to have been successfully retrieved and matched with the reference place image.

We also conducted experimental evaluations of loop-closure detection on the proposed method in two scenarios (afternoon and night), which is shown in Fig. 8. Results demonstrate that under the same scene conditions, the success rate of loop-closure detection of our method is significantly higher than that of iPTN. Comparing the success rate of loop-closure detection of the same method under different scene conditions, it can be seen that the afternoon with better lighting conditions is better than the evening with poor night conditions.

In Fig. 9, we compare the precision-recall curves of our approach with iTPN in two scenes, which further proves the above loop-closure detection results in a quantitative evaluation manner.

V. Conclusions

In this work, we propose a novel robot place recognition framework, called CWPFormer, which uses the cross-weight pyramid Transformer to downsample and upsample pre-training and fine-tuning visual features. To address the computational redundancy due to the attention map having high similarity between heads, We present a CHA module to feed the hash attention head with different complete feature splits. To learn one-dimensional compact hash codes, we adopt a Bayesian learning scheme with a dynamically constructed similarity matrix for accuracy improvement. We conducted experiments based on datasets and real-world environments, and the results not only demonstrated that our method exceeded the existing state-of-the-art methods but also its effectiveness in robot place recognition real-world testing.

References

[1] X. Zhang, L. Wang, Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition,* 2021, vol. 113, pp. 1-21.

[2] Z. Yuan, X. Song, L. Bai, Z. Wang and W. Ouyang, "Temporal-Channel Transformer for 3D Lidar-Based Video Object Detection for Autonomous Driving,' *IEEE Transactions on Circuits and Systems for Video Technology,* 2022, vol. 32, no. 4, pp. 2068-2078.

[3] M. Wieczorek, J. Siłka, M. Woźniak, S. Garg, M. M. Hassan, "Lightweight Convolutional Neural Network Model for Human Face Detection in Risk Situations," *IEEE Transactions on Industrial Informatics,* 2022, vol. 18, no. 7, pp. 4820-4829.

[4] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2022, vol. 45, no. 1, pp. 87-110.

[5] Y. Tian, L. Xie, Z. Wang, L. Wei, X. Zhang, J. Jiao, Q. Ye, "Integrally Pre-Trained Transformer Pyramid Networks," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* Vancouver, Canada, 19-22 Jun. 2023, pp. 18610-18620.

[6] Z. Chen, F. Maffra, I. Sa and M. Chli, "Only look once, mining distinctive landmarks from ConvNet for visual place recognition," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2017, pp. 9-16.

[7] S. Hausler, S. Garg, M. Xu, M. Milford, T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* Online, 19-25 Jun. 2021, pp. 14141-14152.

[8] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* Nevada, USA, 27-30 Jun. 2016, pp. 5297-5307.

[9] J. Lin, L. Y. Duan, S. Wang, Y. Bai, Y. Lou, V. Chandrasekhar, W. Gao, "Hnip: Compact deep invariant representations for video matching, localization, and retrieval," *IEEE Transactions on Multimedia,* vol. 19, no. 9, pp. 1968-1983.

[10] J. Ma, X. Chen, J. Xu, G. Xiong, "SeqOT: A Spatial–Temporal Transformer Network for Place Recognition Using Sequential LiDAR Data," *IEEE Transactions on Industrial Electronics,* 2023, vol. 70, no. 8, pp. 8225-8234.

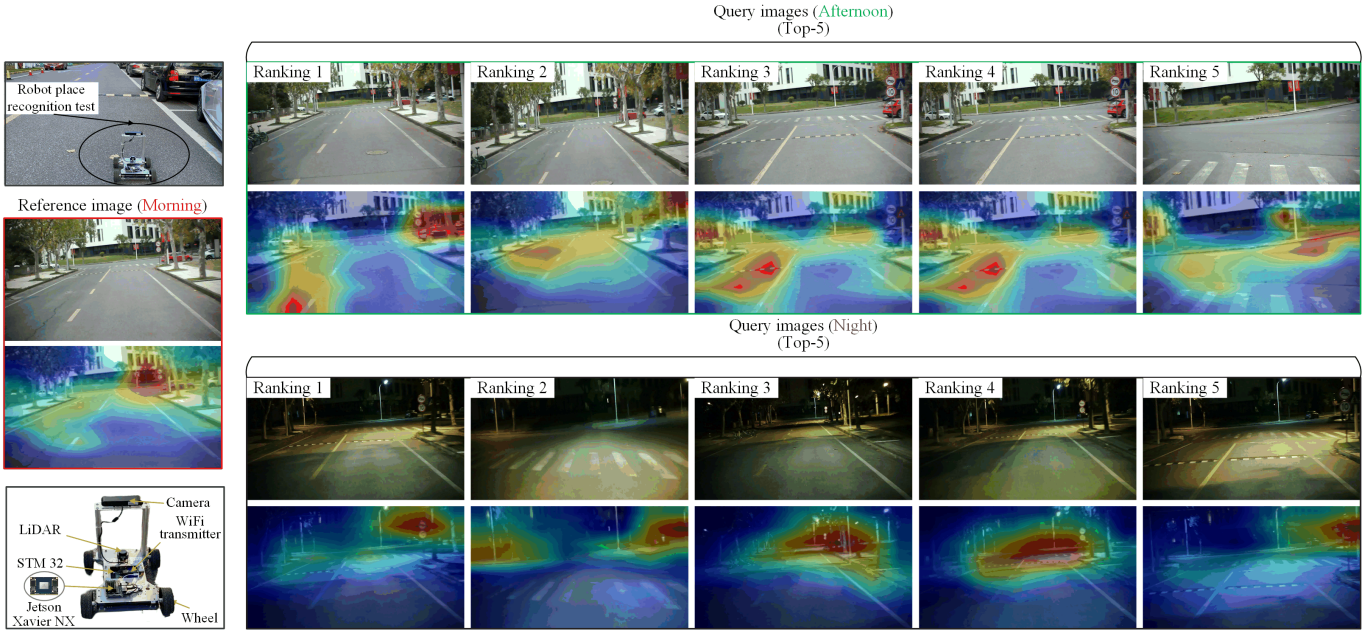
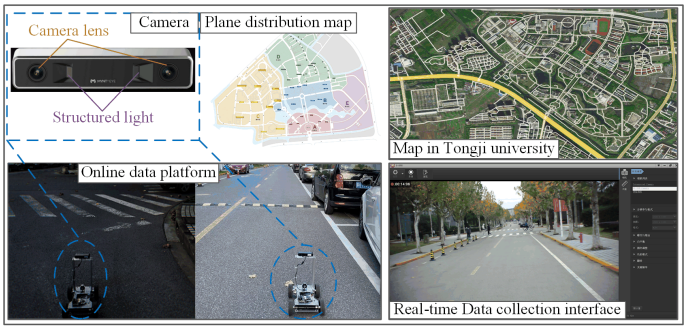
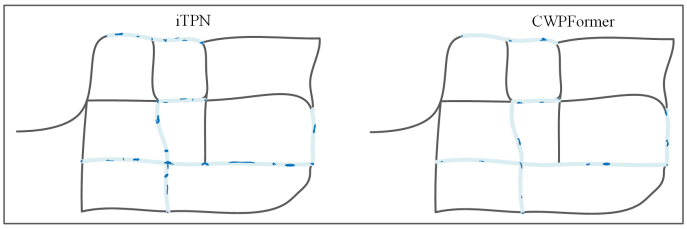


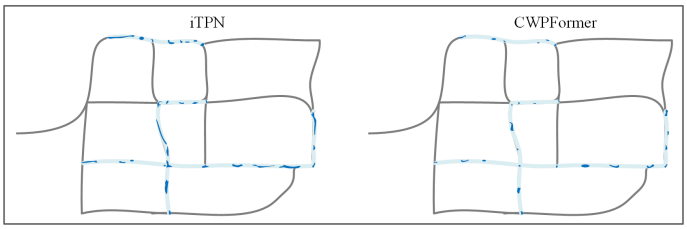
Fig. 7: Real-world robot place recognition experiments.



(a) Robot platform and data collection



(b) Loop-closure detection in afternoon



(c) Loop-closure detection in night

Fig. 8: Qualitative loop-closure detection results of the proposed CWPFormer and iPTN in two scenarios ((a) afternoon and (b) night). The light gray line, Aqua line, and dark blue line represent driving trajectory, successful loop-closure detection and failed loop-closure detection, respectively.

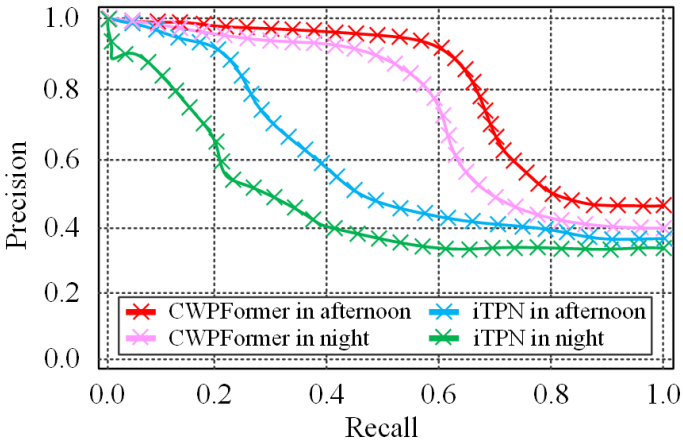


Fig. 9: Comparison of precision-recall curves evaluated using the proposed CWPFormer and iTPN in the afternoon scenes and night scenes.

[11] W. Wang, E. Xie, X. Li, D. P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," *In Proceedings of the IEEE/CVF International Conference on Computer Vision,* Venice, Italy, 11-17 Oct. 2021, pp. 568-578.

[12] X. Zang, G. Li, W. Gao, "Multidirection and Multiscale Pyramid in Transformer for Video-Based Pedestrian Retrieval," *IEEE Transactions on Industrial Informatics,* 2022, vol. 18, no. 12, pp. 8776-8785.

[13] Y. Wang, Y. Qiu, P. Cheng and J. Zhang, "Hybrid CNN-Transformer Features for Visual Place Recognition," *IEEE Transactions on Circuits and Systems for Video Technology,* 2023, vol. 33, no. 3, pp. 1109-1122.

[14] R. Wang, Y. Shen, W. Zuo, S. Zhou, N. Zheng, "Transvpr: Transformer-based place recognition with multi-level attention aggregation," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* Louisiana, USA, 19-24 Jun. 2022, pp. 13648-13657.

[15] Y. Chen, S. Zhang, F. Liu, Z. Chang, M. Ye, Z. Qi, "Transhash: Transformer-based hamming hashing for efficient image retrieval," *In Proceedings of International Conference on Multimedia Retrieval,* Taipei, Taiwan, 21-24 Jun. 2022, pp. 127-136.

[16] T. Li, Z. Zhang, L. Pei, Y. Gan, "HashFormer: Vision Transformer Based

Deep Hashing for Image Retrieval," *IEEE Signal Processing Letters,* 2022, vol. 29, pp. 827-831.

[17] Y. Tian, L. Xie, Z. Wang, L. Wei, X. Zhang, J. Jiao, Q. Ye, "Integrally Pre-Trained Transformer Pyramid Networks," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* Vancouver, Canada, 19-22 Jun. 2023, pp. 18610-18620.

[18] F. J. Chang, M. Radfar,A. Mouchtaris, B. King, S. Kunzmann, "End-to-end multi-channel transformer for speech recognition," *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Online, 6-11 Jun. 2021, pp. 5884-5888.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. "Attention is all you need." *Advances in Neural Information Processing Systems*, Online, 6-14 Dec. 2021, pp. 1-11.

[20] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, Y. Yuan, "EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* Vancouver, Canada, 19-22 Jun. 2023, pp. 14420-14430.

[21] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research,* 2013, vol. 32, no. 11, pp. 1231-1237.

[22] N. Sünderhauf, P. Neubert, P. Protzel, "Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons," *In Processing of IEEE International Conference on Robotics and Automation,* 2013, pp. 1-3.

[23] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello and W. Burgard, "Robust visual SLAM across seasons," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* Hamburg, Germany, 28 Sep-2 Oct. 2015, pp. 2529-2535.

[24] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research,* 2016, vol. 35, no. 10, pp. 1157-1163.

[25] J. Revaud, J. Almazán, R. S. Rezende, C. R. D. Souza, "Learning with average precision: Training image retrieval with a listwise loss," *In Proceedings of the IEEE/CVF International Conference on Computer Vision,* Seoul, South Korea, 27 Oct.-2 Nov.2019, pp. 5107-5116.

[26] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, T. Pajdla, "24/7 place recognition by view synthesis," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* Massachusetts, USA, 7-12 Jun. 2015, pp. 1808-1817.

[27] B. Cao, A. Araujo, J. Sim, "Unifying deep local and global features for image search," *In Proceedings of the European Conference on Computer Vision,* Online, 23-28 Aug. 2020, pp. 726-743.

[28] J. Yu, C. Zhu, J. Zhang, Q. Huang, D. Tao, "Spatial Pyramid-Enhanced NetVLAD With Weighted Triplet Loss for Place Recognition," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 31, no. 2, pp. 661-674.

[29] H. Zhang, X. Chen, H. Jing, Y. Zheng, Y. Wu, C. Jin, "ETR: An Efficient Transformer for Re-ranking in Visual Place Recognition," *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision,* Utah, USA, 3-7 Jan. 2023, pp. 5665-5674.

[30] Y. Xu, P. Shamsolmoali, E. Granger, C. Nicodeme, L. Gardes, J. Yang, "TransVLAD: Multi-scale attention-based global descriptors for visual geo-localization," *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision,* Utah, USA, 3-7 Jan. 2023, pp. 2840-2849.

[31] Y. Tian, L. Xie, Z. Wang, L. Wei, X. Zhang, J. Jiao, Q. Ye, "Integrally Pre-Trained Transformer Pyramid Networks," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* Vancouver, Canada, 19-22 Jun. 2023, pp. 18610-18620.