Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **10**, Number **3**, August **2025**

Attention Heads Prediction for Self-Supervised Learning with Internal Pretext Tasks

Bing-Bing Chen

School of Computer Science and Mathematics
Fujian University of Technology, Fuzhou 350118, China
qalaxice@gmail.com

Wen-Wu He

School of Computer Science and Mathematics
Fujian University of Technology, Fuzhou 350118, China
Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fuzhou 350118, China
hwwhbb@163.com

Ya-Ting Li

School of Computer Science and Mathematics
Fujian University of Technology, Fuzhou 350118, China
liyatingo@163.com

*Corresponding author: Wen-Wu He

Received January 25, 2024, revised July 8, 2024, accepted October 19, 2024.

Abstract. *This paper proposes two innovative self-supervised internal pretext tasks, namely HPAH (The Highest Scoring Patch In The Attention Head) and MPAH (The Masked Patches In The Attention Head), aimed at enhancing the performance of Transformer-like models in image classification. HPAH improves sensitivity to key features and recognition accuracy by predict the index of attention head which the highest scoring patch belongs to. Alternatively, MPAH achieves performance gain by predict the index of attention head which masked patches belong to. Experiments are conducted on four public datasets to validate the applicability and effectiveness of these methods in various settings. The results demonstrate that HPAH and MPAH significantly enhance the performance of CRATE and Vit. This study shows the potential of self-supervised internal pretext task in enhancing the key features learning.*
Keywords: image classification, self-supervised learning, internal pretext task, HPAH, MPAH

1. **Introduction.** In recent years, with the widespread application of various neural network variants in the field of computer vision, we have witnessed a significant leap in the technology of this domain. The training of most neural networks usually adopts a supervised learning paradigm, which, although effective, relies heavily on a large amount of manually annotated data. It is the time-consuming and labor-intensive nature of this process that has prompted researchers to seek new solutions. The emergence of self-supervised learning is a key to addressing this challenge. This kind of method reduces the reliance on manually annotated data, enabling effective training of neural networks even in the absence of labeled data. More importantly, the feature representations obtained through self-supervised learning have shown excellent performance in a variety of visual tasks, demonstrating outstanding generalization ability.

The break of self-supervised learning can be traced back to [1], which proposed the Momentum Contrast (MoCo), marking a significant breakthrough in this field. MoCo focuses on extracting useful feature representations from unlabeled data through contrastive learning. The core of this method is to optimize the loss function by minimizing the distance within positive samples while maximizing the distance between positive and negative samples. After data augmentation, two different cropped regions of the same image are considered as positive samples, while cropped regions from different images are considered as negative samples. To effectively address the memory demand issue during the feature extraction of negative samples, He et al. introduced the so-called momentum encoder. This structure is similar to a regular encoder but views the dictionary composed of all samples as a dynamic queue for participation in loss function computation. The momentum encoder slowly updated by copying parameters from the encoder with a smaller step size, ensuring a sufficient number of negative samples with relatively consistent features. Additionally, Chen et al. [2] proposed SimCLR, which focused on the construction of positive and negative samples, exploring more effective data augmentation preprocessing methods to further enhance model performance. SimCLR treats different regions from the same image as positive samples, while other images as negative samples. This approach added an MLP structure to the encoder, further improving the model's performance. Subsequently, similar self-supervised learning algorithms like BYOL [3] and SwAV [4] were proposed, showing a diverse and active development trend in the field of self-supervised learning. The emergence of MAE [5], which randomly masked input image patches and directly reconstructed the masked image patches, further pushed self-supervised learning to a new peak, demonstrating the continuous development and innovative potential of this field.

The training process of involved model is driven by specific tasks, among which the core of self-supervised learning lies in creatively constructing tasks that facilitate feature representation learning. Currently, the methods for constructing these tasks are mainly divided into three schools: pretext tasks constructing, contrastive learning, and image masking. As the methods based on contrastive learning and image masking require a high volume of samples during the training process, they are more suitable for large-scale pre-training. For finetuning, we tend to adopt self-supervised learning based on pretext tasks, which often involve operations on external images [6], such as changes in color, orientation, sequence, and resolution of the images. Essentially, this approach equates to expand the samples multiple times, thereby incurring higher training costs. To address this problem, in this paper, we focus on constructing pretext tasks based on internal features, generating self-supervised labels by delving into the intrinsic structural features of original image. We integrate these internally oriented pretext tasks into models where attention heads are applicable, particularly, CRATE [7] and Vit [8], to enhance its ability in semantic segmentation.

Specifically, we develope two innovative and efficient self-supervised internal pretext tasks, namely HPAH (The Highest Scoring Patch In The Attention Head) and MPAH (The Masked Patches In The Attention Head). These tasks are distinguished by their plug-in characteristics and minimal demands on time and storage resources, allowing them to be easily integrated into models where attention head is applicable.The main idea of HPAH is to randomly select an attention head from the model and further identify the highest-scoring patch within that attention head. By predicting which attention head this high-scoring patch belongs to, the sensitivity of the model to key features within its attention heads is enhanced. Similar to HPAH, MPAH is also based on a randomly selected attention head, but it masks top-k scoring patches. It utilizes features of unmasked patches to predict the index of the head which masked ones belong to, for

better understanding on the entire feature map. The design of the two internal pretext tasks enhance model performance without any extral samples.

Our contributions can be summarized as follows:

- We introduced self-supervised internal pretext tasks into models where attention heads are applicable, obtaining self-supervised labels without sample expansion. The resultant model understands the entire feature map well and identify key features more precisely.
- We designed two specific self-supervised internal pretext tasks, namely HPAH and MPAH. In HPAH, the model focuses on predicting the attention head where the highest-scoring patch is located; in MPAH, the model aims to predict the attention head which masked patches belong to. Compared to external pretext tasks, our internal pretext tasks cost negligible extra time and memory.
- To validate the effectiveness and universality of our method, we conducted experiments on four general datasets. The experimental results demonstrate that the proposed method is effective for CRATE and Vit, showing notable improvements in classification performance.

2. **Related Work.**

2.1. **Self-Supervised Learning Based on Pretext Tasks Constructing.**

The core of self-supervised learning based on pretext tasks is to construct well-defined tasks.

Doersch et al. [9] believed that context contains abundant supervisory information, and based on context prediction they proposed a pretext task. Specifically, two patches are randomly selected from an image and the task is to predict their relative positions. The model must deeply understand various scenes, objects, and relationships between different parts within the image to accurately predict the relative positions between patches. Zhang et al. [10] took image colorization as the pretext task that feeds channel L of a grayscale image in LAB format into the model to infer channel A and B, and then combining the original L with the inferred A and B to generate a colored image. In this method, the model needs to deeply comprehend the independent semantic information of various scenes and their interconnections to effectively complete the image colorization task. Similar to image colorization, Context Encoders [11] also aimed to learn the semantic information of images through image reconstruction. However, unlike image colorization which reconstructs image from the channel dimension, Context Encoders reconstruct images from the spatial dimension. This is done by masking random blocks or regions of the image, and then feeding the masked image into the model for image reconstruction. Gidaris et al. [12] proposed rotation prediction that rotates images at various angles (such as 90, 180, 270, etc.) and then let the model predict the rotation angle. Compared to other methods, image rotation does not leave any low-level clues to make rotation prediction easy and it forces the model to understand the semantic information of the image well. Moreover, the recognition of rotation angles is a well-defined task. In most cases, objects in images are presented in an upright position, so ideally, the model can clearly identify the degree to which the image has been rotated.

2.2. **Self-Supervised Learning Based on Contrastive Learning.**

The idea behind self-supervised learning based on contrastive learning is that, after various data augmentations, features extracted from the same image by a model should be highly consistent.

MoCo v1 [1] introduced Momentum Contrast that created a sufficiently large and consistent dictionary, maintained via a queue, for the contrastive loss function. Differently, SimCLR [2] used a larger batch size (such as 4096) to construct negative samples and added an MLP nonlinear layer to the encoder. It also explored and used more powerful

data augmentation techniques, providing a simple yet effective framework. MoCo v2 [13] referenced two designs from SimCLR: the MLP layer and stronger data augmentation. Compared to MoCo v1, MoCo v2 significantly improved the performance, even surpassed SimCLR. In both MoCo and SimCLR models, negative samples are essential, and large number of negative samples have to be used to avoid model collapses and shortcut solutions. Conversely, BYOL [3], within the contrastive learning framework, discarded negative samples and relied solely on positive samples for training, using an asymmetric structure and a momentum update mechanism to prevent model collapse. Its loss function is a simple MSE loss, requiring distance calculation only for positive samples. A contemporaneous work with BYOL, SwAV [4], proposed an online clustering loss that can be effectively optimized regardless of batch size, without the need for large queues or momentum encoders. SwAV adopted the dot product between an output of encoder and the clustering center to get a similarity matrix which was used to predict the output of another network, achieving swap prediction. Additionally, SwAV introduced a plug-in multi-crop data augmentation strategy, that was effective for other contrastive learning frameworks as well. All the aforementioned models used CNN-based backbones, while MoCo v3 [14] combinesd contrastive learning with ViT. Rather than using a queue for key storage, MoCo v3 adopted a larger batch size (such as 4096) and freezed the patch projection layer in ViT during the training process.

2.3. **Self-Supervised Learning Based on Image Masking.** The idea behind self-supervised learning based on image masking is to predict the masked areas with unmasked ones, forcing the model to learn intrinsic features and structures of the image.

Masked Autoencoders (MAE) [5] masked 75% blocks of the image and employed an asymmetric encoder-decoder architecture. In MAE, only unmasked image blocks were used and a lightweight decoder was adopted, to reduce computation and memory costs. By reconstructing pixel values of masked blocks, MAE outperformed MoCo v3 [14] in most partial and full finetuning tasks. Concurrent with MAE, another notable work was SimMIM [15], which used random masking and a linear prediction head, achieving impressive results. Unlike MAE, MaskFeat [16] used HOG [17] feature descriptors as reconstruction target. Moreover, MaskFeat can be directly trained on unlabeled video datasets and had shown excellent transfer learning performances.

3. **Approach.** Based on attention heads prediction, we propose two internal pretext tasks for self-supervised learning to avoid multiple expansion of samples, named HPAH and MPAH respectively. Specifically, HPAH randomly selects an attention head and then identifies the highest-scoring patch within it, with the aim to predict which attention head this high-scoring patch belongs to. Unlike HPAH, MPAH masks top-k scoring patches within a randomly selected attention head and then uses the features of the remaining patches to predict which attention head these masked ones belong to. The two innovative self-supervised internal pretext tasks are adaptable to Transformer-like models where attention heads are used.

As for the backbone, we adopt CRATE [7], which demonstrates exceptional performance in semantic segmentation, and Vit [8], the details of which we omit due to its similarity in principle. The architectural diagram of the CRATE model is shown in Figure 1 (it is modified from [7]). We believe that the strengths of CRATE in semantic segmentation, combined with our self-supervised internal pretext tasks, can effectively enhance the sensitivity of attention heads to key features, thereby deepen its understanding of the entire feature map. To conserve memory usage and reduce training costs, we

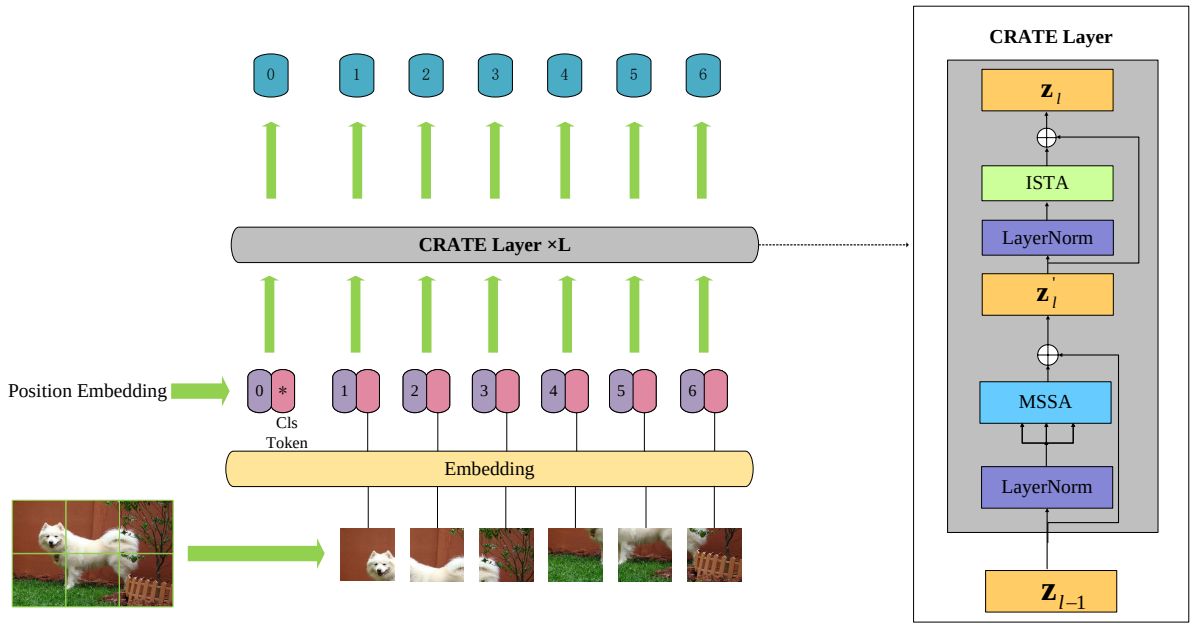

FIGURE 1. The overall architecture diagram of CRATE (modified from [7]). The right half shows the specific structure of the CRATE Layer.

opt to implement our self-supervised internal pretext tasks in the last CRATE layer of the model.

3.1. **Notation and Preliminaries.** In this paper, we use $\mathbf{z} = [\mathbf{z}^0; \mathbf{z}^1, \mathbf{z}^2, \cdots, \mathbf{z}^N]$ to represent input features, where $\mathbf{z}^0$ represents the cls token, and N denotes the total number of image tokens. The cls token and all the image tokens are fed together into a CRATE layer composed of MSSA (Multi-Head Subspace Self-Attention) and ISTA (Iterative Shrinkage-Thresholding Algorithm). In CRATE, the roles of MSSA and ISTA are analogous to MHSA (Multi-Head Self-Attention) and MLP (Multi-Layer Perceptron) components in ViT. We assume that the model has a total of L layers, and the features of the l-th layer can be described as:

$$\mathbf{z}_l^{'} = MSSA(LN(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1} \tag{1}$$

$$\mathbf{z}_l = ISTA(LN(\mathbf{z}_l^{'})) + \mathbf{z}_l^{'} \tag{2}$$

Assuming that there are H attention heads in MSSA, the output of MSSA is obtained by aggregating all the heads:

$$MSSA(LN(\mathbf{z})) = Aggregate(\mathbf{SSA_1}, \ldots, \mathbf{SSA_h}, \ldots, \mathbf{SSA_H}) \tag{3}$$

3.2. **HPAH.** Figure 2 shows the implementation details of HPAH. In the last CRATE layer, we first randomly select an attention head $\mathbf{A_h}$. By using the *torch.topk*() function, we sort the scores in the attention map of this attention head and select top-1 ($k = 1$) attention score along with its corresponding position information *max_patch_pos*.

$$max\_score, max\_patch\_pos = topk(\mathbf{A_h}, k = 1, dim = -1, largest = True) \tag{4}$$

Utilizing *max_patch_pos*, we can precisely select the corresponding patch block $\mathbf{z}_{max}$ from $\mathbf{z}_l$. In this model, $\mathbf{z}_{max}$ represents an important region within that attention head:

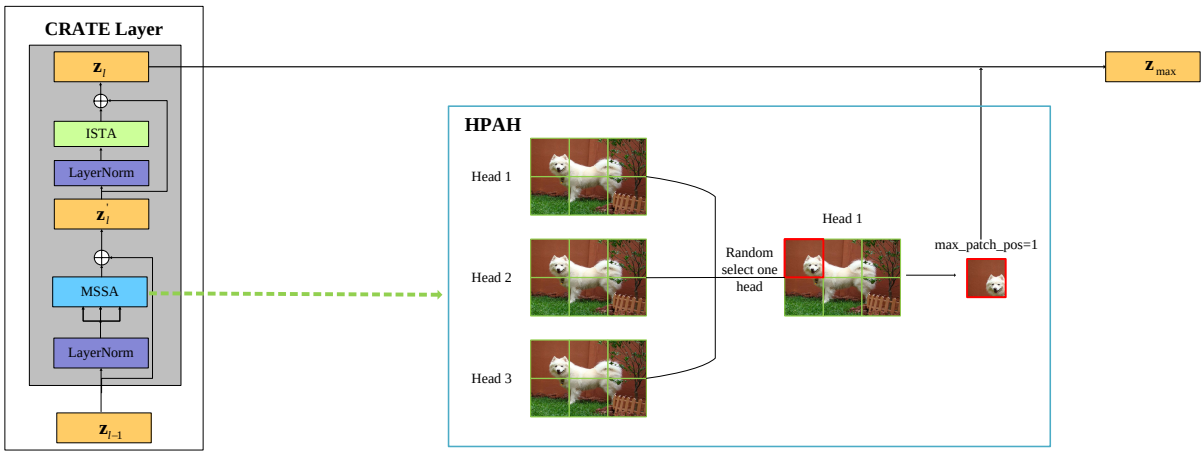

FIGURE 2. Overview of HPAH. The red box represents the highest-scoring patch, whose feature representation $\mathbf{z}_{max}$ is used as the input for self-supervised learning and the index of attention head which the red box belongs to is used as the label.

$$\mathbf{z}_{max} = \mathbf{z}_l[:, max\_patch\_pos + 1, :] \tag{5}$$

The size of $\mathbf{z}_l$ is $[B, N+1, C]$, where B represents the batch size, N denotes the total number of patches, and C refers to the number of channels. Since array indices start from 0, and index 0 is precisely where the cls token is located, we need to add 1 to the original index. Subsequently, we use $\mathbf{z}_{max}$ to predict which head the highest-scoring patch comes from, to realize self-supervised learning based on internal pretext tasks. As for supervised learning, we adopt standard learning of CRATE. Ultimately, the total loss of the model is composed of the supervised and the self-supervised cross-entropy loss:

$$L = L_{SUP}(\mathbf{z}_L) + \lambda \cdot L_{HPAH}(\mathbf{z}_{max}) \tag{6}$$

where λ is a trade-off hyperparameter.

3.3. **MPAH.** Figure 3 shows the implementation details of MPAH. Similar to HPAH, in the last CRATE layer of the model, we randomly select an attention head $\mathbf{A_h}$. While, we use *torch.topk*() to obtain a ranking of the scores in the attention map of this attention head. Then, we select a subset of high scoring attention scores and their corresponding position information, known as *topk_patch_pos*:

$$topk\_score, topk\_patch\_pos = topk(\mathbf{A_h}, k = 3, dim = -1, largest = True) \tag{7}$$

In this process, we select top-3 ($k = 3$) attention scores from the attention head $\mathbf{A_h}$. Assuming the original multi-head attention weights matrix are $\mathbf{A}$, we create a binary mask $\mathbf{M}$ of the same size as $\mathbf{A}$. Within this mask $\mathbf{M}$, we set the weight values at the *topk_patch_pos* positions of the h-th attention head $\mathbf{A_h}$ to 0, while the others are set to 1. By multiplying the original attention weights $\mathbf{A}$ with the mask $\mathbf{M}$, we obtain a new attention map, $\tilde{\mathbf{A}}_h$. Therefore, the output of the MSSA can be represented as follows:

$$\begin{aligned} \hat{\mathbf{z}}_{l-1} &= MSSA(LN(\mathbf{z}_{l-1})) \\ &= Aggregate(\mathbf{SSA_1}, \ldots, \mathbf{SS\tilde{A}_h}, \ldots, \mathbf{SSA_H}) \end{aligned} \tag{8}$$

Then the features of the l-th layer can be represented as:

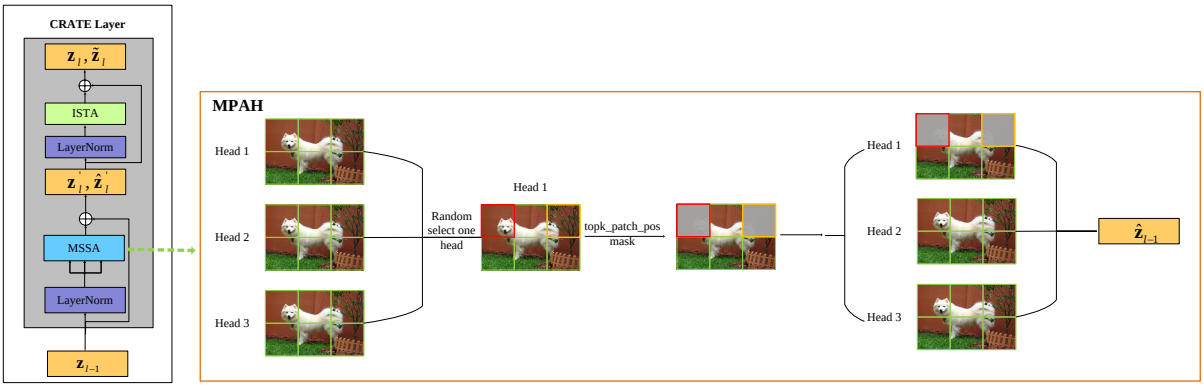

FIGURE 3. Overview of MPAH. The gray boxes indicate top-k scoring patches that have been masked, and the resultant feature representation $\tilde{\mathbf{z}}_l$ is used as the input for self-supervised learning and the index of attention head which masked patches belong to is used as the label.

$$\hat{\mathbf{z}}_l^{'} = \hat{\mathbf{z}}_{l-1} + \mathbf{z}_{l-1} \tag{9}$$

$$\tilde{\mathbf{z}}_l = ISTA(LN(\hat{\mathbf{z}}_l^{'})) + \hat{\mathbf{z}}_l^{'} \tag{10}$$

Similar to HPAH, the total loss of the model is composed of the supervised and the self-supervised cross-entropy loss, which can be expressed as:

$$L = L_{SUP}(\mathbf{z}_L) + \lambda \cdot L_{MPAH}(\tilde{\mathbf{z}}_L) \tag{11}$$

where λ is a trade-off hyperparameter.

4. **Experiments.** To validate the effectiveness of the proposed self-supervised internal pretext tasks, we conducted a series of experiments on four open-source datasets. In this section, we first introduce the details of the datasets used in the experiments and the configuration of the training parameters. Subsequently, we delve into the advantages and characteristics of our methods, through a series of ablation studies and intuitive visual analyses.

TABLE 1. General Setting of Training Parameters

Parameters	Metric settings
Epochs	200
Batch-size	64
Optimizer	AdamW
Weight decay	0.01
Learning rate	$5e^{-5}$
Image size	224*224
λ	0.2

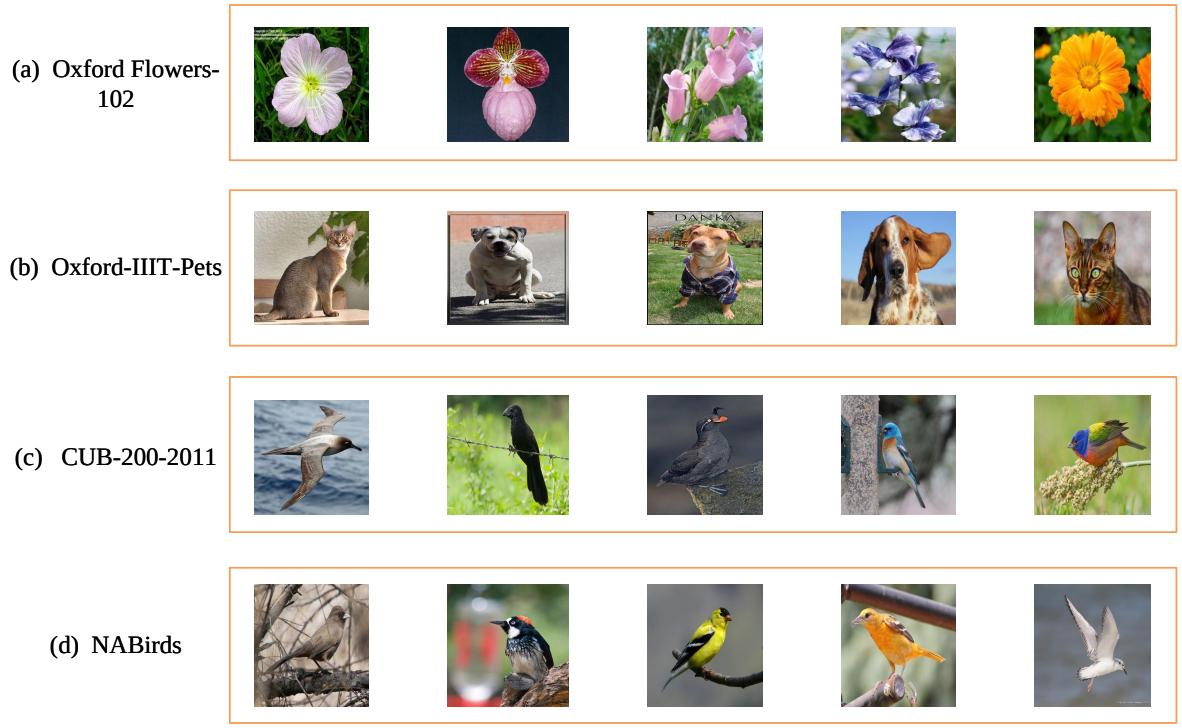

FIGURE 4. Images sampled from different categories in Oxford Flowers-102, Oxford-IIIT-Pets, CUB-200-2011 and NABirds datasets.

4.1. **Datasets And Implementation Details.** In the experiments, four open-source datasets are used: Oxford Flowers-102 [18], Oxford-IIIT-Pets [19], CUB-200-2011 [20], and NABirds [21]. Figure 4 displays sample images from the four datasets. Oxford Flowers-102 is widely used in computer vision research, containing 102 different flower categories with a total of $8,189$ images. The images included are highly diverse, with different angles, scales, and lighting conditions. Oxford-IIIT-Pets consists of 37 different pet breeds, with a total of $7,349$ images. The images therein cover various breeds of cats and dogs, including different sizes, colors, and postures, with approximately 200 images for each breed. CUB-200-2011 is a fine-grained bird recognition dataset, comprising 200 bird species with a total of $11,788$ images. It covers various bird species, with each species having about 60 images, featuring birds in different postures, backgrounds, and lighting conditions. NABirds is a large North American bird dataset, containing 555 bird species and $48,562$ images. The images include a wide variety of bird species, each with different images showing various angles and environments.

In experiments, CRATE_small is used as our main backbone to evaluate the performance of the proposed self-supervised internal pretext tasks, which is pre-trained on ImageNet-21K [22]. Following standard practice, we initially resize the all images to a uniform dimension of $224 * 224$ and segment them into patches of size $16 * 16$. During the training process, we adhered to the same strategy of CRATE, utilizing the AdamW optimizer with a learning rate set to $5e^{-5}$ and weight decay to 0.01. Concurrently, in HPAH and MPAH, λ are both set to 0.2. Considering memory constraints, we set the batch size to 64 and conducted training for 200 epochs. The specific settings of the parameters are summarized in Table 1. As for Vit, we follow the same training strategy as [23] and conducte experiments on CUB-200-2011 and NABirds. Notice that, here we do not pursue SOTA performance with entensive parameter tuning, mainly we aim to show the performance gain offered by supervised internal pretex task.

All experiments are performed with two Nvidia GeForce *A*6000, and utilize PyTorch as the toolkit.

Table 2. Top-1 Accuracy of Related Models on Four Banchmark Datasets

Model	Oxford Flowers-102	Oxford-IIIT-Pets
CRATE	85.8	87.9
CRATE+HPAH	86.6(+0.8)	88.5(+0.6)
CRATE+MPAH	86.5(+0.7)	89.5(+1.6)
Model	CUB-200-2011	NABirds
Vit	90.3	89.9
Vit+HPAH	90.6(+0.3)	90.2(+0.3)
Vit+MPAH	90.8(+0.5)	90.3(+0.4)

4.2. **Main Results Analysis.** In the experimental study, we conduct a comparative analysis between the original (CRATE , Vit) and the one integrated with the proposed self-supervised internal pretext tasks HPAH and MPAH. Through experiments on four open-source datasets, we discover that HPAH and MPAH significantly improve the performance of CRATE and Vit. Experimental results can be found in Table 2. Due to the difference of equipment used in this paper compared to that in the CRATE, we ran CRATE model 10 times and take the averaged accuracy as the baseline. Baseline results of Vit model are from [23].

Specifically, by integrating with HPAH, the performance of CRATE is improved noticeably. Particularly on Oxford Flowers-102, we observe a substantial improvement of 0.8. Additionally, HPAH achieves a 0.6 enhancement on the Oxford-IIIT-Pets. When HPAH is implanted into ViT, both CUB-200-2011 and NABirds achieve an improvement of 0.3. These results clearly demonstrate the effectiveness of HPAH in enhancing the recognition capabilities of CRATE and ViT.

Similarly, when implementing MPAH, CRATE achieves a significant improvement of 1.6 on the Oxford-IIIT-Pets, an impressive finding. MPAH also leds to a performance increase of 0.7 on Oxford Flowers-102. When MPAH is implanted into ViT, CUB-200-2011 achieves an improvement of 0.5, and 0.4 on NABirds. These improvements further confirm the effectiveness of MPAH over different datasets.

Overall, by incorporating proposed self-supervised internal pretext tasks, HPAH and MPAH, into CRATE and Vit, significant performance enhancements can be observed on all datasets. These results validate the effectiveness of self-supervised learning and highlight the potential of internal pretext tasks for performance gain without too much overhead.

4.3. **Visualization.** To more intuitively demonstrate the advantages of the proposed method, we visualized the attention maps of different attention heads in CRATE. Figure 5 shows attention maps of CRATE and those with HPAH. The sample in the image is from Oxford-IIIT-Pets. From Figure 5, we can observe that, without HPAH, some key areas can not be highlighted accurately by attention heads. However, with HPAH, it is noticeable that the attention map scores in key areas are significantly enhanced. These results indicates that, with HPAH, CRATE can focus more on key areas for image classification, leading to improve accuracy.

We believe these improvements are due to the self-supervised internal pretext task, that prompts the model to delve deeper into key features and inherent structure of images. This

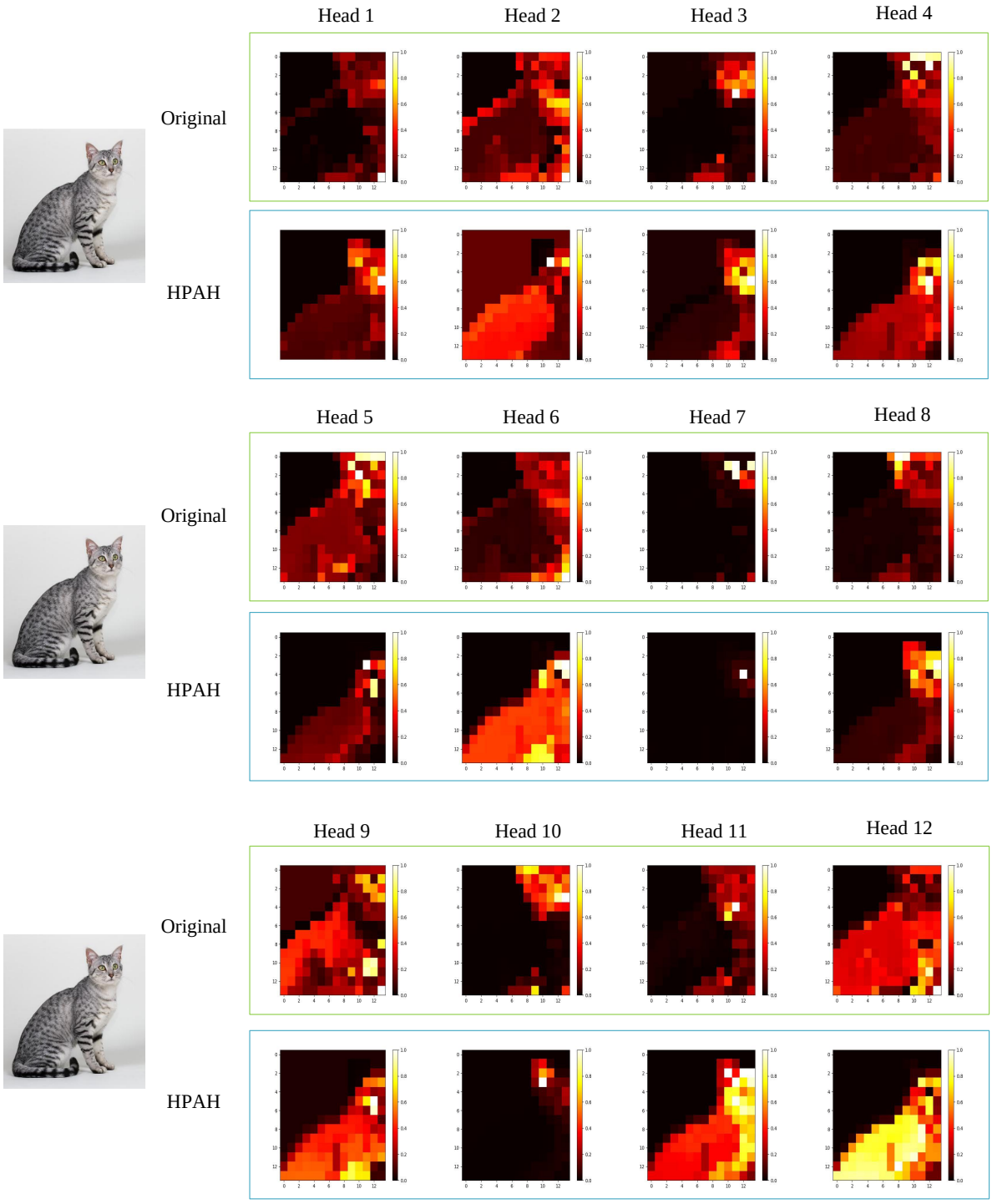

FIGURE 5. Attention map visualizations of different attention heads in the last layer of CRATE. The green box displays the attention maps of each attention head in CRATE, while the blue box shows the attention maps of the corresponding attention heads in CRATE with HPAH. The visualizations indicate that, without HPAH, some key areas can not be highlighted accurately by attention heads. However, with HPAH the attention map scores in key areas are significantly enhanced.

helps the model to identify more informative features and focus more on these features, minimizing redundant information.

4.4. **Ablation Study.** To investigate the impact of experimental settings, we conduct ablation studies. All ablation experiments are carried out with CRATE backbone and performed on Oxford-IIIT-Pets.

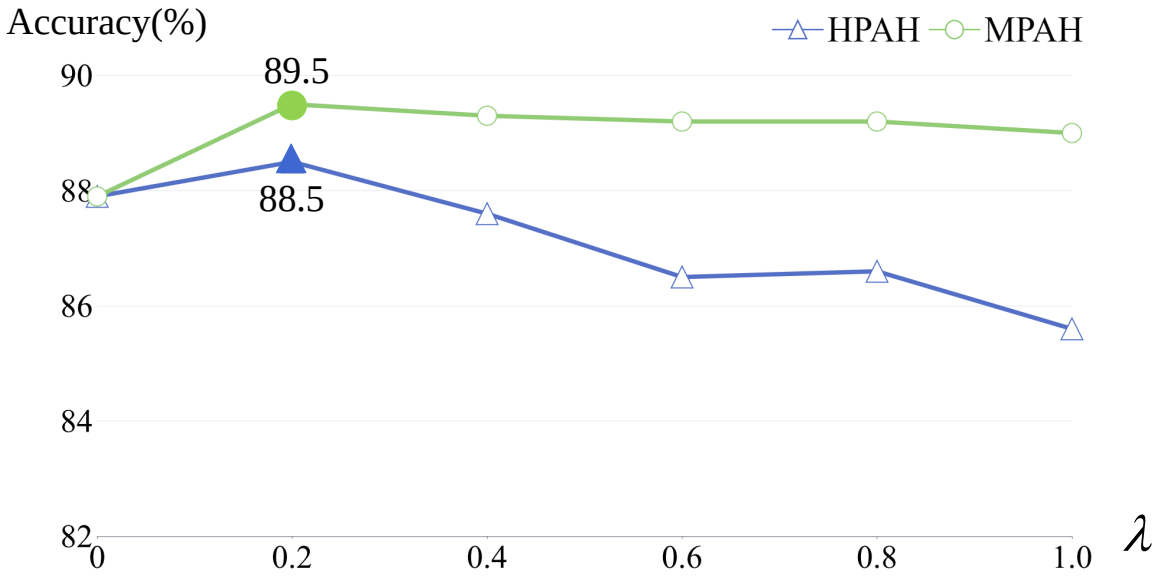

FIGURE 6. The impact of λ value in the total loss of HPAH/MPAH.

Effect of λ value. To investigate the impact of λ value of the total loss, we conduct a set of experiments by set λ to $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. The specific experimental results are displayed in Figure 6. From the figure, it is observed that when HPAH and MPAH are placed in the last (12th) layer, $\lambda = 0.2$ provides the best performance. Conversely, as the λ value increases, the model performance gradually weakens. This finding indicates that a balance between self-supervised loss and supervised loss is crucial.

Location of HPAH. To investigate the impact of location of HPAH, we insert HPAH into the 3rd, 6th, 9th, and 12th CRATE layer respectively, with a fixed λ value, i.e. 0.2. The specific experimental results are displayed in Figure 7. From the figure, it is observed that when HPAH is placed in the last (12th) layer, the performance is the best. Conversely, as the layer number decreases, the model performance gradually weakens. We speculate that this may be due to the shallower layers of the model have not learned sufficiently mature knowledge, thereby failing to provide effective scores to the attention map.

Number of selected patches in HPAH. To investigate the impact of k values in top-k patches in HPAH, we conduct experiments using top-1, top-3, and top-5 scoring patches respectively from the attention map, with a fixed λ value, i.e. 0.2. The specific experimental results are displayed in Figure 8. As observed in the figure, the performance is optimal when the highest-scoring patch is used and more patches can not provid performance gain. Attention heads in CRATE tend to focus on distinct target parts, and HPAH can further enhance the sensitivity of the attention maps.

Number of masked patches in MPAH. To investigate the impact of the number of masked patches in MPAH, we conduct experiments using top-1, top-3, and top-5 scoring

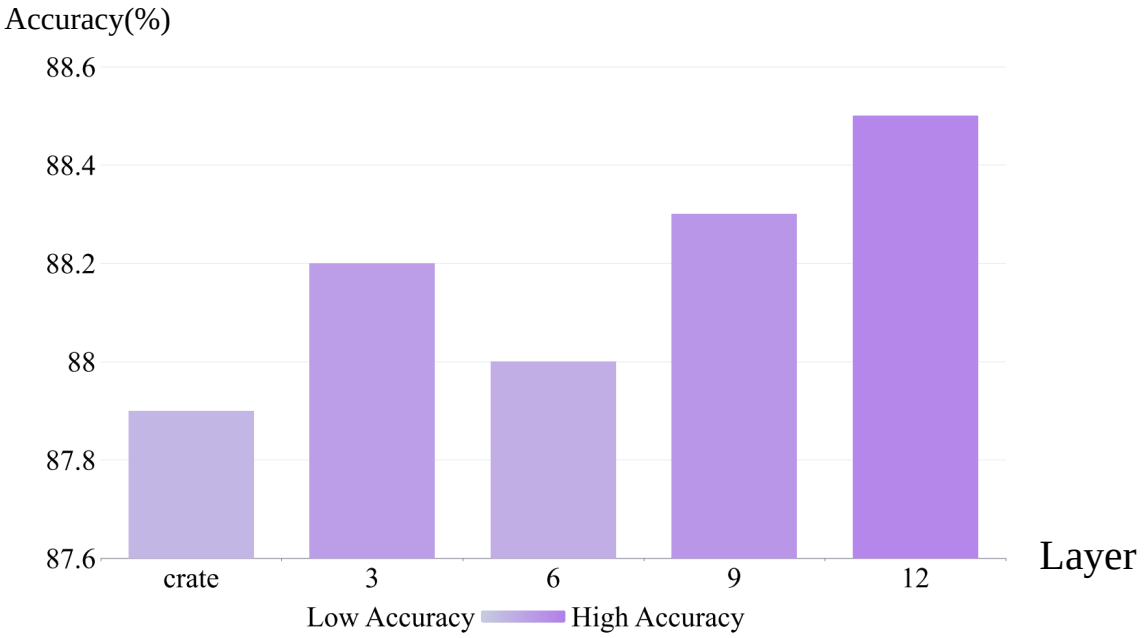

FIGURE 7. The impact of location where HPAH placed in ($\lambda = 0.2$).

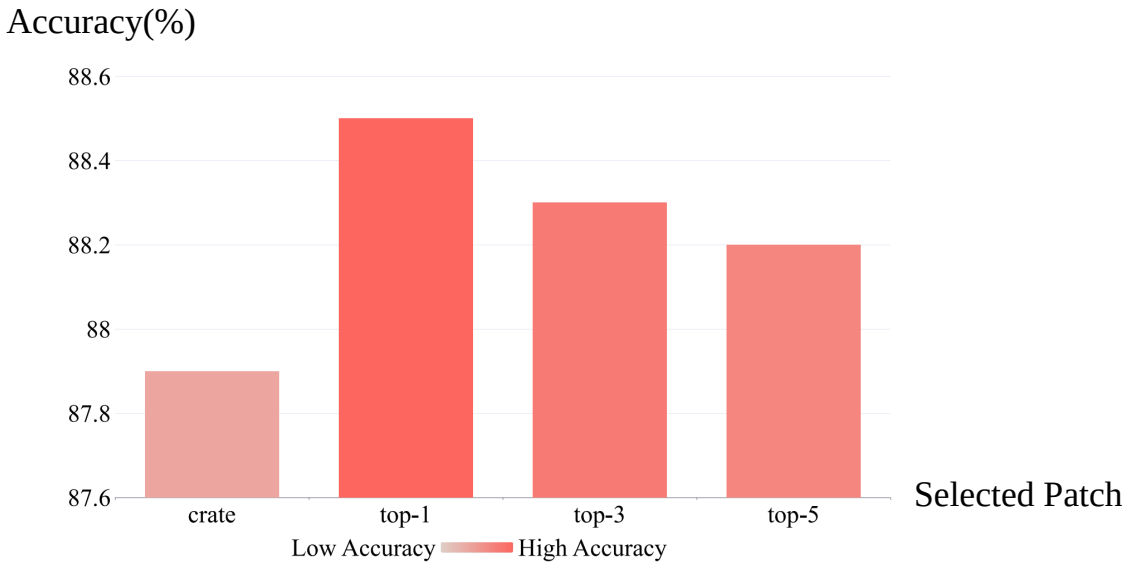

FIGURE 8. The impact of number of top-k scoring patches used for self-supervised learning in HPAH ($\lambda = 0.2$).

patches respectively from the attention map, with a fixed λ value, i.e. 0.2. The specific results are displayed in Figure 9, where we find that top-3 can outperform the two competitors.

5. **Conclusion.** In this study, we propose two self-supervised internal pretext tasks, namely HPAH and MPAH, to enhance the performance of CRATE and Vit in image classification tasks. The core idea of HPAH is to use the index of attention head where the highest-scoring patch belongs to as the self-supervised label, to encourage CRATE and Vit focus more on key areas and improve its classification accuracy. Alternatively,

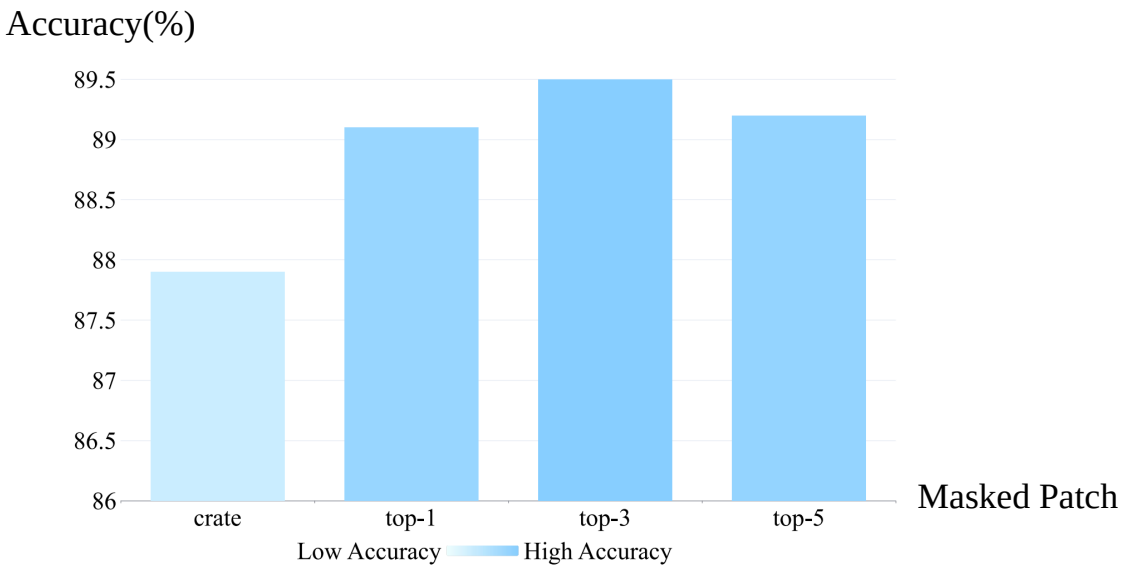

FIGURE 9. The impact of the number of masked patches in MPAH ($\lambda = 0.2$).

MPAH masks top-k scoring patches, and use the index of attention head where masked patches belong to as the self-supervised label, to achieve performance gain. HPAH and MPAH shows significant performance improvements across banchmark datasets. In conclusion, our research confirms the effectiveness of self-supervised learning in enhancing the performance of models where attention heads are applicable. These self-supervised internal pretext tasks offer a new perspective for in-depth understanding key features in images, providing insights for future research in image classification.

Acknowledgment. This work is supported by the Natural Science Foundation of Fujian Province under Grant 2020J01891 and Grant 2018H4005, and partly by the National Natural Science Foundation of China under Grant 41971340.

REFERENCES

[1] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1597–1607.

[3] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.

[4] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "learning of visual features by contrasting cluster assignments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9912–9924, 2020.

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[6] T. Xie, Y. Yang, Z. Ding, X. Cheng, X. Wang, H. Gong, and M. Liu, "Self-supervised feature enhancement: Applying internal pretext task to supervised learning," *IEEE Access*, vol. 11, pp. 1708–1717, 2022.

[7] Y. Yu, T. Chu, S. Tong, Z. Wu, D. Pai, S. Buchanan, and Y. Ma, "Emergence of segmentation with minimalistic white-box transformers," in *Conference on Parsimony and Learning*. PMLR, 2024, pp. 72–93.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[9] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[10] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 649–666.

[11] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[12] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://enpc.hal.science/hal-01864755

[13] X. Chen, H. Fan, R. B. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *ArXiv*, vol. abs/2003.04297, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:212633993

[14] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers. in 2021 ieee," in *CVF International Conference on Computer Vision (ICCV)*, pp. 9620–9629.

[15] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9653–9663.

[16] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 668–14 678.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[18] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008, pp. 722–729.

[19] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3498–3505.

[20] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," Tech. Rep., 2011.

[21] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 595–604.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[23] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, and C. Wang, "Transfg: A transformer architecture for fine-grained recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 852–860.