









Conformer: Local Features Coupling Global Representations for Recognition and Detection

Zhiliang Peng , Zonghao Guo , Wei Huang , Yaowei Wang , *Member, IEEE*, Lingxi Xie , *Member, IEEE*, Jianbin Jiao , *Member, IEEE*, Qi Tian , *Fellow, IEEE*, and Qixiang Ye , *Senior Member, IEEE*

Abstract—With convolution operations, Convolutional Neural Networks (CNNs) are good at extracting local features but experience difficulty to capture global representations. With cascaded self-attention modules, vision transformers can capture long-distance feature dependencies but unfortunately deteriorate local feature details. In this paper, we propose a hybrid network structure, termed Conformer, to take both advantages of convolution operations and self-attention mechanisms for enhanced representation learning. Conformer roots in feature coupling of CNN local features and transformer global representations under different resolutions in an interactive fashion. Conformer adopts a dual structure so that local details and global dependencies are retained to the maximum extent. We also propose a Conformer-based detector (ConformerDet), which learns to predict and refine object proposals, by performing region-level feature coupling in an augmented cross-attention fashion. Experiments on ImageNet and MS COCO datasets validate Conformer’s superiority for visual recognition and object detection, demonstrating its potential to be a general backbone network.

Index Terms—Feature fusion, image recognition, object detection, vision transformer.

I. INTRODUCTION

CONVOLUTIONAL Neural Network (CNNs) have significantly advanced computer vision tasks such as image recognition, object detection, and instance segmentation [1], [2], [3], [4], [5], [6]. This can attribute to convolution operations which collect hierarchical and rich features as image representation and the pooling operations which handle local object deformation by enlarging the receptive fields.

Despite of the advantages of CNNs [1], [2], [3], [4], [5], [6], they are limited by the local receptive field and thereby

experience difficulty to capture global representations, e.g., long-distance relationships among visual elements. Such long-distance relationships are critical for high-level vision tasks. One solution is enlarging the receptive field by using intensive pooling operations, which however damage feature resolution and discrimination power.

Recently, the transformer architecture [7] has been introduced to vision tasks [8], [9], [10], [11], [12], [13], [14], [15], [16]. The ViT method [8] constructs a sequence of tokens by splitting each image to patches with positional embeddings and applies cascaded transformer blocks to extract token vectors as representations. Thanks to the self-attention mechanism and multi-layer perceptron (MLP) structure, vision transformers significantly enlarged receptive fields, constituting global representations with long-distance feature dependencies.

Unfortunately, vision transformers are observed ignoring local feature details, which decreases the discriminability between backgrounds and foregrounds, Fig. 1(c) and (g). To solve, the tokenization module [8] leveraged CNN feature maps as input tokens [11] or used sliding windows [17] to extract fine-detailed representation. Nevertheless, the problem to fully exploit the complementarities of local features and global representations in a uniform framework remains to be elaborated.

In this paper, we propose a dual network structure, termed Conformer, with the aim to enhance feature representations by coupling CNN-based local features with transformer-based global representations. Conformer consists of a CNN branch and a transformer branch which respectively follow the design of ResNet [4] and ViT [8]. The two branches form a comprehensive combination of local convolution blocks, self-attention modules, and MLP units. During training, the cross entropy losses are used to supervise both the CNN and transformer branches to couple CNN-style and transformer-style features.

Considering the feature misalignment between CNN and transformer features, an Feature Coupling Unit (FCU) is designed as the bridge. On the one hand, to fuse the two-style features, FCU leverages 1×1 convolution to align the channel dimensions, down/up sampling strategies to align feature resolutions, and LayerNorm [18] and BatchNorm [19] to align feature values. In an interactive fashion, FCU eliminates the semantic divergence between two kinds of features. Furthermore, FCU decomposes the features by assigning them learnable coefficients so that the feature components are orthogonal and complementary. In this way, FCU enhances the global perception capability of local features and the local details of global representations to a maximum extent.

Manuscript received 10 February 2022; revised 26 December 2022; accepted 2 February 2023. Date of publication 7 February 2023; date of current version 30 June 2023. This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61836012, 62171431, and 62225208, and in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27000000. Recommended for acceptance by O. Russakovsky. (Corresponding author: Qixiang Ye.)

Zhiliang Peng, Zonghao Guo, Wei Huang, Jianbin Jiao, and Qixiang Ye are with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China (e-mail: pengzhiliang19@mails.ucas.ac.cn; guozonghao19@ucas.ac.cn; huangwei19@mails.ucas.ac.cn; jiaojb@ucas.ac.cn; qxye@ucas.ac.cn).

Yaowei Wang is with the Peng Cheng Laboratory, Shenzhen, Guangdong Province 100013, China (e-mail: wangyw@pcl.ac.cn).

Lingxi Xie and Qi Tian are with the Huawei Cloud, Shenzhen, Guangdong Province 518129, China (e-mail: 198808xc@gmail.com; tian.qi1@huawei.com).

Code is available at github.com/pengzhiliang/Conformer.

Digital Object Identifier 10.1109/TPAMI.2023.3243048

0162-8828 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information.

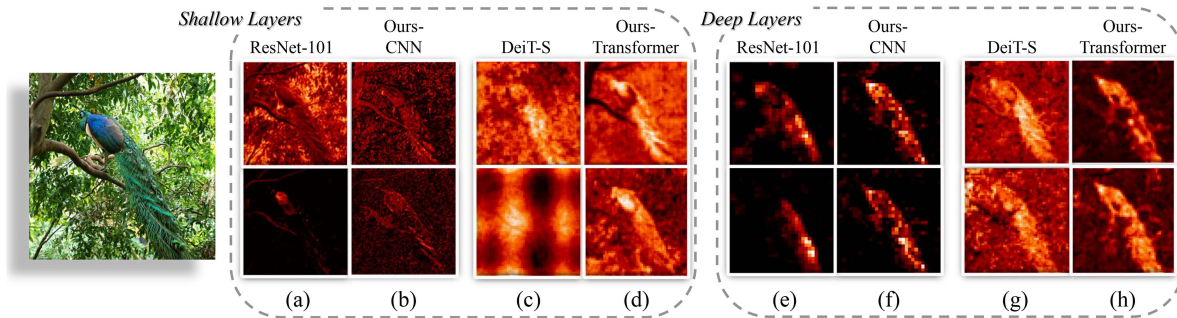


Fig. 1. Comparison of feature maps of CNN (ResNet-101) [4], vision transformer (DeiT-S) [10], and the proposed Conformer. The patch tokens in transformer are reshaped to feature maps for visualization. While CNN activates discriminative local regions (e.g., the peacock’s head in (a) and tail in (e)), the CNN branch of Conformer takes advantage of global cues from the vision transformer and thereby activates complete object (e.g., full extent of the peacock in (b) and (f)). Compared with CNN, local feature details of the vision transformer are deteriorated (e.g., (c) and (g)). In contrast, the transformer branch of Conformer retains the local feature details from CNN while depressing the background (e.g., the peacock contours in (d) and (h) are more complete than those in (c) and (g)). *This figure is best viewed in color.*

In Fig. 1, we visualize and compare the features of CNNs, Transformer, and Conformer. While conventional CNNs (e.g., ResNet-101) tend to retain discriminative local regions (e.g., the peacock’s head or tail), the CNN branch of Conformer can activate the full object extent, Fig. 1(b) and (f). When solely using transformers, for the missing local fine-details (e.g., blurred object boundaries), it is difficult to distinguish the object from the background, Fig. 1(c) and (g). Coupling of local features and global representations significantly enhances discriminability, Fig. 1(d) and (h).

The Conformer method was first proposed in our ICCV 2021 paper [20]. In this full version, it is promoted by introducing the orthogonal feature coupling unit to improve the feature complementary and augmented cross-attention unit (ACU) for feature alignment and region-level feature coupling. Based on feature coupling in both the backbone and the detector head, we design a Conformer-based object detector (ConformerDet). ConformerDet first predicts sparse proposals by using transformer tokens. The predicted object proposals are progressively refined for object detection by iteratively coupling the transformer features with CNN local features. Experiments validate that coupling features from the two architectures can enhance the discriminability and localization accuracy of objects.

The contributions of this paper are summarized as follows:

- We propose the Conformer network with a dual structure, which naturally inherits the structure and generalization advantages of CNNs and vision transformers.
- We propose the Feature Coupling Unit (FCU), which fuses convolutional local features with transformer-based global representations in a complementary and interactive fashion. We further propose the augmented cross-attention unit (ACU) for feature coupling on the detector head, validating the superiority of Conformer structure for object detection.
- Under comparable parameter complexity, Conformer outperforms CNNs and vision transformers by significant margins. With comparable computational costs, ConformerDet outperforms the CNN-based and transformer-based detectors. Experimental results demonstrate the advantages to fuse CNN and Transformer features as visual representation.

II. RELATED WORK

CNNs With Long-Range Feature Dependency. CNNs can be regarded as a hierarchical ensemble of local features with progressively enlarged receptive fields. Unfortunately, CNNs [1], [2], [4], [5], [21], [22], [23] experience difficulty to capture long-range feature dependencies although they are good at organizing local features. To solve, one solution is to define larger receptive fields by introducing deeper architectures and/or more pooling operations [6], [24]. Dilated convolution [25], [26] increased the sampling step size, while deformable convolution [27] learned the sampling positions. SENet [6] and GENet [24] leveraged global average pooling to aggregate global context and then used it to re-weight feature channels, while CBAM [28] respectively used global Maxpooling and Avgpooling to refine features in spatial and across channels.

The other solution is the global attention mechanism [29], [30], [31], [32], [33], which has demonstrated advantages when capturing long-distance dependencies in natural language processing [7], [34], [35]. The non-local operation [29] was introduced to CNNs in a self-attention fashion so that the response at each position is a weighted sum of the features at all (global) positions. Attention augmented convolutional networks [31] concatenated convolutional feature maps with self-attentional feature maps to augment convolution operations. Relation Networks [32] proposed an object attention module, which processes a set of objects simultaneously through interaction between their appearance feature and geometry.

However, existing solutions that introduce global cues to CNNs have obvious disadvantages. For the first solution, larger receptive fields require more intensive pooling operations, which reduce feature resolutions. For the second solution, the straightforward combination of convolutional operations with attention mechanisms could interfere the training procedure.

Vision Transformers. As a pioneered work, ViT [8] validated the feasibility of pure transformer architectures for vision tasks. One important conclusion is that self-attention mechanisms capturing feature long-distance dependencies are crucial for image classification [9], [10], [11], object detection [12], [14], [36], semantic segmentation [15], image enhancement [13],

weakly-supervised object localization [37] and image generation [16], [38].

However, the self-attention mechanisms in vision transformers often ignore local feature details and object structures. To solve, DeiT [10] proposed using a distillation token to transfer CNN-based features to vision transformer while T2T-ViT [11] using a tokenization module to re-organize the image to tokens considering neighboring pixels. Swin Transformer [17] used a hierarchical architecture computed with Shifted windows, which bring not only fine-detailed features but also higher efficiency. The Multiscale Transformer [39] and Pyramid Transformer [40] created feature pyramids with early layers operating at high spatial resolution to model simple low-level visual information, and deeper layers at spatially coarse, but complex, high-dimensional features.

Despite of the substantial progress, vision transformers are far from perfect. While the tokenization in vision transformer could destruct object structures [41], the ignorance of local spatial constrains of features aggregate the training difficulty. This inspires us to resort to CNNs, which possess sophisticated theoretical basis, design criterion and training policies.

In the related works, one commonly used strategy is replacing some self-attention blocks in transformers with convolution operations, so that the spatial priority are defined to easy model training [42], [43], [44], [45]. The other way is cascading the convolution operations with self-attention operations in/across transformer blocks in serial fashions [33], [46], [47]. The cross-covariance image transformer (XCiT) [48] combined the accuracy of transformers with the scalability of convolution architectures by cascading local path interactions with cross-covariance attentions in each layer. In a different way, Conformer defines the first dual network structure which fuses features in an interactive fashion. Such a structure not only naturally inherits the structure advantages of both CNN and transformers but also retains the representation capability of local features and global representations to the maximum extent.

Transformer-Based Detectors. These detectors incorporated long-range semantic dependency in feature representation and improved the discrimination capability on objects of small sizes, irregular layouts and clutter backgrounds. DETR [12] used the transformer encoder-decoder architecture for detection, which extracts the feature dependency between objects and captures the global context in the whole image. ViDT [49] and Deformable DETR [36] introduced the reconfigured transformer decoder to collect feature dependency in multiple scales, which benefits detecting small objects. Anchor DETR [50], Conditional DETR [51] and SMCA DETR [52] improved DETR by introducing spatial priors, which reduced the representation ambiguity and improved the detection efficiency. YOLOS [53] was designed not to be yet another high-performance object detector, but to unveil the versatility and transferability of transformer from image recognition to object detection.

Transformer-based detectors have utilized the long-range feature dependency provided by the self-/cross-attention mechanism of the transformer. However, they unfortunately missed local feature details caused by the coarse-grained image patch inputs, which would be solved by introducing the Conformer network structure.

III. CONFORMER NETWORK

A. Motivation

In the past forty decades, the fundamental question of “where visual processing begins” or “what are the primitives of visual representation” has been extensively explored in the area of cognitive science [54], [55]. Conventional cognitive models are mostly “local-first”: detecting local features (such as oriented line segments) first and then integrating them, typically using attention, to build objects. The global-first approach claims that topological invariants, which constitute a formal description of global, Gestalt-like operations, are the most primitive ones and are extracted at the very beginning of visual processing. After years debate, however, the question remains unanswered.

In the computer vision area, local features and global representations are important counterparts. Local features [56], [57], [58], which are compact vector representations defined within small pixel neighborhoods, have been the building blocks of modern computer vision algorithms. Global representations include, but not limited to, contour representations, shape descriptors, and object typologies at long distance [59]. In the deep learning era, CNNs collects local features in a hierarchical manner via convolutional operations and retains the local cues as feature maps. Vision transformer is believed to aggregate global representations among the compressed patch tokens in a soft fashion by cascaded self-attention modules.

To take both advantages of local features and global representations, we design the dual Conformer structure, Fig. 2(c). Within Conformer, the global context information from the transformer branch is consecutively fed to CNN feature maps, to reinforce their global perception capability. Similarly, local features from the CNN branch are progressively fed back to patch tokens, to enrich the local details. In this way, Conformer implements feature representations which are either/both global-first or/and local-first, providing a computational mechanism to investigate the fundamental problem of “where visual processing begins”.

B. Network Architecture

As shown in Fig. 2, Conformer is composed of a stem module, dual CNN-transformer branches, multiple FCUs to bridge the dual branches, and two classifiers. The stem module, which is a 7×7 convolution with stride 2 followed by a 3×3 max pooling with stride 2, is used to extract local features (e.g., edge and texture information), which are then fed to the dual network branches. The CNN branch and transformer branch are composed of N (e.g., 12) cascaded convolution and transformer units, respectively, Table I. With a concurrent structure, the local details and global context are preserved to the maximum extent.

FCU is defined as a bridge module to fuse local features from the CNN branch with global representations from the transformer branch, Fig. 2(b). FCU is applied from the second block as the initialized features of the two branches are same. Along the branches, FCU progressively fuses feature maps and patch tokens in an interactive fashion. At the last stage of the CNN branch, all the features are pooled and fed to one classifier. At the last stage of the transformer branch, the class token is

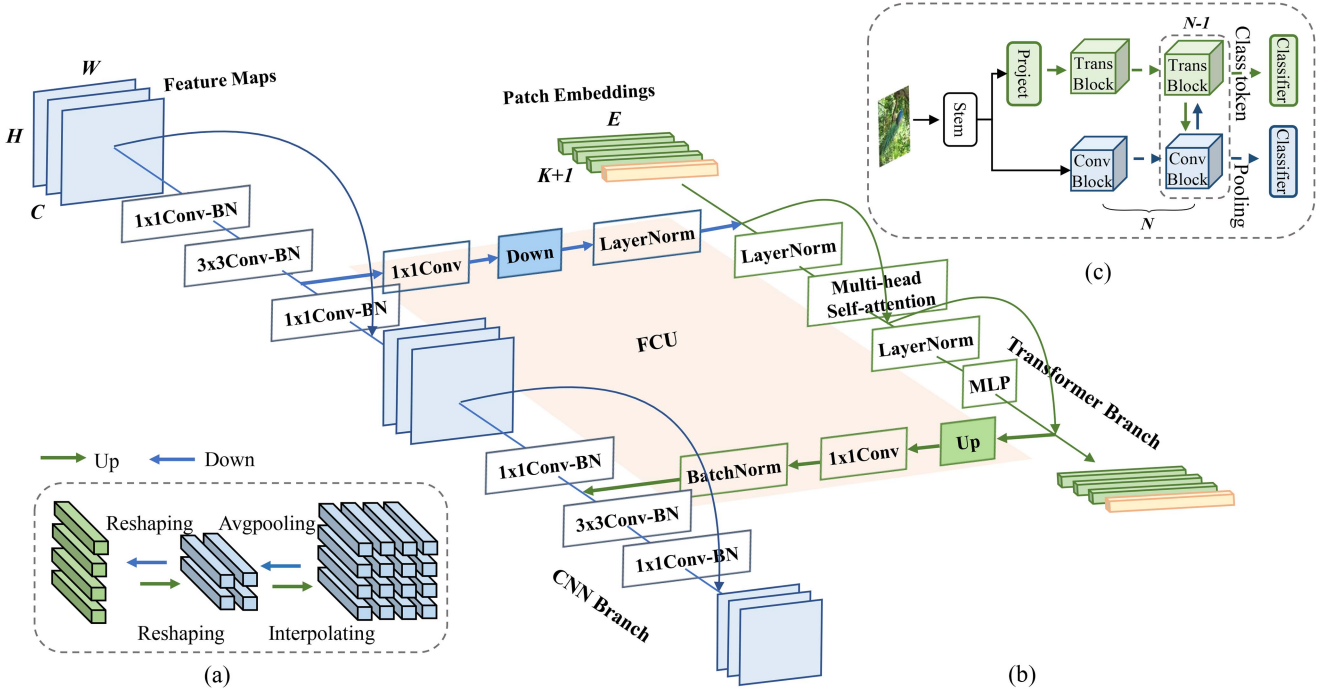


Fig. 2. Conformer network architecture. (a) Up-sampling and down-sampling modules for spatial alignment of feature maps and patch tokens in the Feature Coupling Unit (FCU). (b) Implementation details of the CNN block, the transformer block, and the FCU. (c) Thumbnail of the Conformer network with the dual architecture.

TABLE I
ARCHITECTURE CONFIGURATION OF CONFORMER VARIANTS

Model	Stem module	Blocks (4 stages)	CNN branch (4 stages)		Transformer branch (4 stages)			#Param.	FLOPs
			Channel	Feature Map size	Projection	Tokens	MLP size		
Conformer-Ti			[128, 256, 512, 512]		4×4, 384, stride 4		1152	23.1M	5.7G
Conformer-S	7×7, 64, stride 2		[256, 512, 1024, 1024]		4×4, 384, stride 4		1536	37.7M	10.7G
Conformer-B	3×3, Maxpooling	[4, 4, 3, 1]	[384, 768, 1536, 1536]	$[\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}]$	4×4, 576, stride 4		2304	83.3M	23.5G
Conformer-S/32			[256, 512, 1024, 1024]		8×8, 384, stride 8		1536	38.9M	7.1G

Feature Map Size denotes the size relative to input image size.

taken out and fed to the other classifier. The scores of the two classifiers are summarized as the classification results.

During training, we use two cross-entropy losses to separately supervise the two classifiers. The weights of the loss functions are empirically set to the same. During inference, the outputs of the two classifiers are simply summarized as the prediction results.

CNN Branch. As shown in Fig. 2(b), the CNN branch adopts a feature pyramid structure, where the feature map resolution decreases and the feature channel number increases when network goes deep. Each convolution block is composed of n_c (set to 2 by default) bottlenecks. Denote the feature map of the c -th channel and the l -th layer as X_c^l , the forward process of l -th block in the CNN branch is formulated as:

$$\tilde{X}_{c_1}^l = \text{Conv}_{3 \times 3}(\text{Conv}_{1 \times 1}(X_c^l)), \quad (1)$$

$$\hat{X}_{c_1}^l = \text{Conv}_{1 \times 1}(\tilde{X}_{c_1}^l) + X_c^l, \quad (2)$$

$$\tilde{X}_{c_2}^l = \text{Conv}_{1 \times 1}(\hat{X}_{c_1}^l), \quad (3)$$

$$\hat{X}_{c_2}^l = \text{FCU}(\tilde{X}_{c_2}^l, X_t^{l+1}), \quad (4)$$

$$X_c^{l+1} = \text{Conv}_{1 \times 1}(\text{Conv}_{3 \times 3}(\hat{X}_{c_2}^l)) + \hat{X}_{c_1}^l, \quad (5)$$

where $\text{Conv}_{k \times k}$ denotes a convolutional unit. Each convolutional unit is composed of a convolution layer with convolutional kernel size k , an ReLU activation layer and a batch normalization (BN) layer [19]. X_t^{l+1} is the output of l -th block from the transformer branch.

Transformer Branch. Following ViT [8], this branch first performs tokenization and then cascades N transformer blocks. For tokenization, we compress the feature maps generated by the stem module into 14×14 patch tokens without overlap, by a linear projection layer, which is a 4×4 convolution with stride 4. A class token is then pretended to the patch tokens for classification. As shown in Fig. 2(b), given the input patch tokens X_t^l , the forward process of l -th block in the transformer branch can be formulated as:

$$\tilde{X}_t^l = \text{FCU}(X_t^l, \tilde{X}_{c_1}^l), \quad (6)$$

$$\hat{X}_t^l = \text{ATT}(\text{LN}(\tilde{X}_t^l)) + \tilde{X}_t^l, \quad (7)$$

$$X_t^{l+1} = \text{MLP}(\text{LN}(\hat{X}_t^l)) + \hat{X}_t^l, \quad (8)$$

where ATT, LN, MLP respectively denote the multi-head self-attention, layer normalization [19] and multilayer perceptron. Considering that $\tilde{X}_{c_1}^l$ from (2) encodes both local features and spatial location information because of the 3×3 convolution layer [60], the positional embeddings are no longer required. This facilitates increasing image resolution for downstream vision tasks.

C. Feature Coupling Unit

Denote $\tilde{X}_{c_1}^l \in \mathbb{R}^{N \times C_c \times H \times W}$ (defined in (2)) the CNN feature maps and $X_t^l \in \mathbb{R}^{N \times E \times C_t}$ (defined in (8)) the transformer patch tokens. The FCU is defined to align and fuse features $\tilde{X}_{c_1}^l$ and X_t^l , Fig. 5.

Spatial Alignment. When feeding the CNN feature maps ($\tilde{X}_{c_1}^l$) to the transformer branch, FCU defines a linear layer $W_{ct} \in \mathbb{R}^{C_c \times C_t}$ to align the channel numbers of $\tilde{X}_{c_1}^l$ and X_t^l . It then leverages a down-sampling layer and a reshaping layer to align the spatial resolutions of $\tilde{X}_{c_1}^l$ and X_t^l and a LayerNorm layer to regularize the feature distribution of X_t^l . The regularized CNN features are denoted as $\tilde{X}_{c_1 t}^l \in \mathbb{R}^{N \times E \times C_t}$. This procedure is formulated as

$$\tilde{X}_{c_1 t}^l = \text{LN}(\text{Reshaping}(\text{Downsampling}(W_{ct}\tilde{X}_{c_1}^l))). \quad (9)$$

When feeding the transformer patch tokens $X_t^{l+1} \in \mathbb{R}^{N \times E \times C_t}$ to the CNN branch, FCU leverages a linear layer $W_{tc} \in \mathbb{R}^{C_t \times C_c}$ to align the feature channel numbers of X_t^{l+1} and $\tilde{X}_{c_2}^l$. It then uses a reshaping layer and an up-sampling layer to align the spatial resolutions of $\tilde{X}_{c_2}^l$ and a BatchNorm layer to regularize the feature distribution of $\tilde{X}_{c_2}^l$. The regularized representation is denoted as $X_{tc}^{l+1} \in \mathbb{R}^{N \times C_c \times H \times W}$. This procedure is formulated as:

$$X_{tc}^{l+1} = \text{BN}(\text{Upsampling}(\text{Reshaping}(W_{tc}X_t^{l+1}))). \quad (10)$$

For the Downsampling operation in (9), we compare ‘Max-pooling’, ‘Avgpooling’, ‘Convolution’ and ‘Attention’. For ‘Attention’, we utilize the cross-attention layer to align the spatial resolutions of $\tilde{X}_{c_1}^l$ (after Reshaping) and X_t^l :

$$\tilde{X}_{c_1 t}^l = \text{Softmax} \left(\frac{(X_t^l W_q)(\tilde{X}_{c_1 t}^l W_k)^T}{\sqrt{C_t}} \right) (\tilde{X}_{c_1 t}^l W_v),$$

where $W_q, W_k, W_v \in \mathbb{R}^{C_t \times C_t}$ are learned linear transformations which map the input X_t^l to queries Q , and $\tilde{X}_{c_1 t}^l$ to keys K and values V , respectively.

Similarly, for the Upsampling operation in (10), ‘Interpolation’ and ‘Attention’ are compared. For ‘Attention’, we let the $\tilde{X}_{c_2}^l$ be the queries Q , and X_{tc}^{l+1} (after Reshaping) be the keys K and values V :

$$X_{tc}^{l+1} = \text{Softmax} \left(\frac{(\tilde{X}_{c_2}^l W_q)(X_{tc}^{l+1} W_k)^T}{\sqrt{C_t}} \right) (X_{tc}^{l+1} W_v).$$

The ablation studies suggest that the simple average-pooling and nearest neighbor interpolation are more appropriate for the FCU.

Orthogonal Feature Fusion. After aligning the features, how to fuse $\tilde{X}_{c_1 t}^l$ with X_t^l and couple X_{tc}^{l+1} with $\tilde{X}_{c_2}^l$ remains to

be elaborated. Beyond the off-the-shelf adding or concatenating strategies, we propose the ‘coupling’ operation to maximize the complementary and orthogonality of features. Specifically, when coupling $X_{c_1 t}^l$ with $\tilde{X}_{c_1 t}^l$, taking the $X_t^l/|X_t^l|$ as unit vector e_t , we project $\tilde{X}_{c_1 t}^l$ onto X_t^l and obtain the codirectional component,

$$\tilde{X}_{c_1 t}^l \parallel_{e_t} = \frac{\tilde{X}_{c_1 t}^l \cdot X_t^l}{|X_t^l|} \cdot e_t, \quad (11)$$

and the orthogonal component,

$$\tilde{X}_{c_1 t}^l \perp_{e_t} = \tilde{X}_{c_1 t}^l - \tilde{X}_{c_1 t}^l \parallel_{e_t}, \quad (12)$$

where \parallel and \perp respectively denote ‘parallel’ and ‘vertical’.

We re-scale $\tilde{X}_{c_1 t}^l \parallel_{e_t}$ and $\tilde{X}_{c_1 t}^l \perp_{e_t}$ in a learnable fashion and update (6) as

$$\tilde{X}_t^l = \alpha_{\parallel e_t} \times \tilde{X}_{c_1 t}^l \parallel_{e_t} + \alpha_{\perp e_t} \times \tilde{X}_{c_1 t}^l \perp_{e_t} + X_t^l, \quad (13)$$

where $\alpha_{\parallel e_t}$ and $\alpha_{\perp e_t}$ are two diagonal matrices. For instance, $\alpha_{\parallel e_t}$ is defined as

$$\alpha_{\parallel e_t} = \begin{bmatrix} \alpha_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_{C_t} \end{bmatrix}. \quad (14)$$

Similarly, when coupling X_{tc}^{l+1} with $\tilde{X}_{c_2}^l$, taking the $\tilde{X}_{c_2}^l/|\tilde{X}_{c_2}^l|$ as a unit vector e_c , we update (4) and have the coupled feature $\hat{X}_{c_2}^l$, as

$$\hat{X}_{c_2}^l = \alpha_{\parallel e_c} \times X_{tc}^{l+1} \parallel_{e_c} + \alpha_{\perp e_c} \times X_{tc}^{l+1} \perp_{e_c} + \tilde{X}_{c_2}^l. \quad (15)$$

Considering distribution difference between the CNN branch and the transformer branch, there exist feature/semantic gaps between feature maps and patch tokens in cascaded network branches. Thereby, FCU is applied in each block (except the first block) to progressively fill the feature/semantic gaps.

D. Analysis

Structure Analysis. By considering the FCU as a short connection, we can abstract the dual structure to a serial residual structure, Fig. 3(a). With different residual connection units, Conformer implements different combinations of bottlenecks (as in ResNet, Fig. 3(b)) and transformer blocks (as in ViT, Fig. 3(d)), which implies that it combines the structural advantages of CNNs and vision transformers. Furthermore, it achieves various permutations of bottlenecks and transformer blocks at different depths, including but not limited to Fig. 3(c) and (e). This greatly enhances the representation capacity of the network.

Feature Analysis. We visualize the feature maps, class activation maps and attention maps in in Figs. 1 and 4. Compared with ResNet [4], with the coupled global representations, the CNN branch in Conformer tends to activate full object extent regions rather than object parts, which suggests long-distance feature dependencies Figs. 1(f) and 4(a). Thanks to the fine-detailed local features progressively provided by the CNN branch, the patch tokens of the transformer branch retain detailed local features (Fig. 1(d) and (h)), which are deteriorated by the vision transformers [8], [10] (Fig. 1(c) and (g)). Furthermore, the

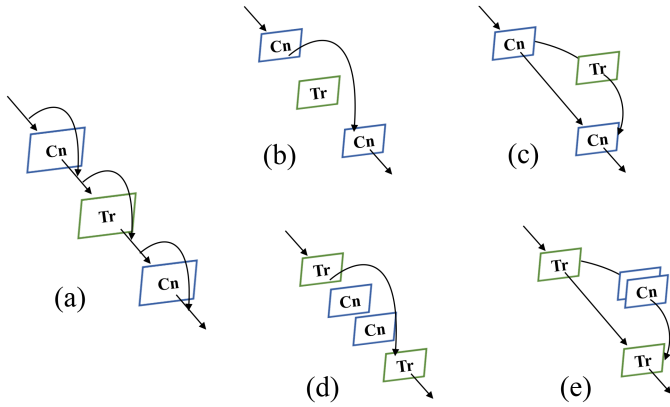


Fig. 3. Network structure analysis. C_n and T_r respectively denote a bottleneck and a transformer block. (a) The dual structure can be considered as a special serial case of the residual structure. (b) The CNN (e.g., ResNet); (c) A special hybrid structure where the transformer block is embedded to bottlenecks. (d) The vision transformers (e.g., ViT); (e) A special case where the bottlenecks are embedded to the transformer blocks.

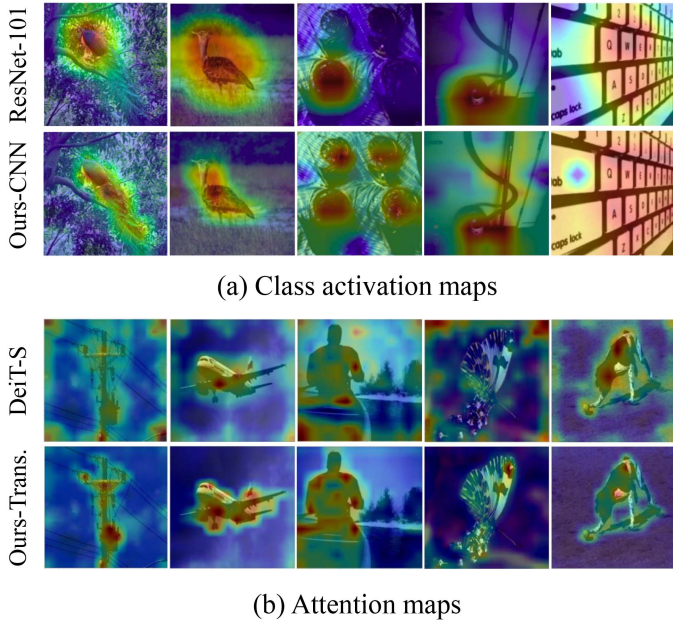


Fig. 4. Visualization of feature activation by CNN (ResNet-101) [4], vision transformer (DeiT-S) [10], and our proposed Conformer. (a) Class activation maps in ResNet-101 and the CNN branch of Conformer-S by using the CAM method [61]. (b) Attention maps in DeiT-S and the transformer branch of Conformer-S by using the Attention Rollout method [62]. This figure is best viewed in color.

attention area in Fig. 4(b) is more complete while the background is significantly suppressed, implying higher discriminative capacity of learned feature representations.

IV. CONFORMERDET

Following the backbone network, Fig. 5, the detector head (ConformerDet) has a dual structure with two network branches. Along the transformer branch, ConformerDet learns to model the long-range semantic dependency, which is coupled with the

local features extracted by the CNN branch to detect objects. ConformerDet is composed of multi-stage proposal prediction and proposal refinement modules.

A. Flowchart

The long-range semantic dependency from the vision transformer benefits extracting representation related to full object extent [53], [63], which enables it to cover objects with learnable sparse proposals. The sparse proposals are learned by tokens (feature vectors) embedded to the vision transformer.

Proposal Token Embedding. We propose to represent each object region using a feature vector $e_i \in \mathbb{R}^{1 \times C}$ (termed proposal token), where i indexes the proposal and C the feature dimension. For each image, we randomly initialize N sparse proposal tokens, which are expected to cover all objects in the image. These proposal tokens construct a token set $\mathbf{E} = \{e_i\}_{i=1}^N \in \mathbb{R}^{N \times C}$, which is embed to the transformer to learn representation for object localization and classification, Fig. 5. Specially, the token set \mathbf{E} is appended to the inputs (image patches \mathbf{T}) of the vision transformer to form a new set of embeddings $\mathbf{L} = \{\mathbf{T}, \mathbf{E}\} \in \mathbb{R}^{(M+N) \times C}$, where M denotes the number of image patches.

Proposal Prediction. The proposal tokens embedded in transformer are trained to predict object proposals, Fig. 5. Each object proposal consists of a classification score s_i and a bounding box $b_i = \{x_i, y_i, w_i, h_i\}$ which respectively denote the normalized center coordinates, object width and object height. The bounding box b_i for proposal token e_i is initialized to cover the whole input image. All the initial bounding boxes construct a box set $\mathbf{B} = \{b_i\}_{i=1}^N \in \mathbb{R}^{N \times 4}$. Each proposal token e_i predicts a classification score s_i by passing a linear layer. Using e_i as input, three perception layers equipped with ReLU activation functions are used to predict bounding box offsets $\delta b_i = \{\delta x_i, \delta y_i, \delta w_i, \delta h_i\}$. Based on the offsets δb_i , a bounding box b_i is updated to $\hat{b}_i = \{\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i\}$ where $\hat{x}_i = x_i + w_i \delta x_i$, $\hat{y}_i = y_i + h_i \delta y_i$, $\hat{w}_i = w_i e^{\delta w_i}$, and $\hat{h}_i = h_i e^{\delta h_i}$. All predicted bounding boxes are updated to a box set $\hat{\mathbf{B}} = \{\hat{b}_i\}_{i=1}^N \in \mathbb{R}^{N \times 4}$. Given multiple proposal tokens, e.g., a token set \mathbf{E} , we construct a proposal set $\mathbf{R} = \{\hat{b}_i, s_i\}_{i=1}^N = \{r_i\}_{i=1}^N \in \mathbb{R}^{N \times (4+D)}$, where D denotes the object category number.

Proposal Refinement As shown in Fig. 5, multi-stage proposal refinement modules incorporate object proposal prediction and feature coupling in a learnable framework for iterative optimization. When performing proposal prediction, tokens are used to update the object proposals, which guide the extraction of local features f_i . In the feature coupling procedure, the local feature f_i are enhanced and fused by the corresponding token e_i and obtain the updated proposal token \hat{e}_i . The proposal prediction and feature coupling procedures are performed alternatively and iteratively, so that the object features are progressively enhanced and the object locations are gradually refined.

B. Proposal Feature Coupling

Given the predicted bounding box set $\hat{\mathbf{B}}$, an RoIAlign module [64] is employed to extract the CNN local features $\mathbf{F} = \{f_i\}_{i=1}^N \in \mathbb{R}^{N \times C \times S \times S}$, where $S = 14$ is the features

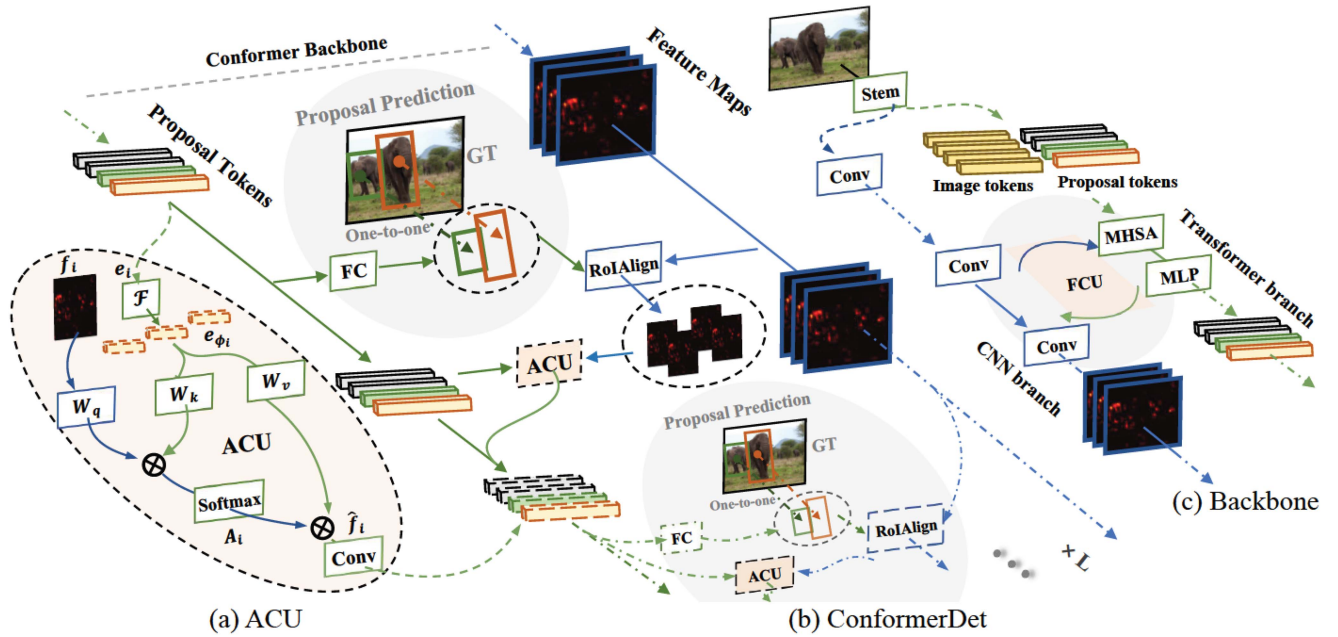


Fig. 5. ConformerDet flowchart. (a) Augmented cross-attention unit (ACU). (b) ConformerDet head. (c) The Conformer backbone. Feature maps and proposal tokens from Conformer are fed to the detector head (ConformerDet), which leverages augmented cross-attention units (ACUs) to couple token representations with local CNN features to enhance the discriminability and localization accuracy.

height/width. The proposal tokens \mathbf{E} incorporating long-range semantic dependency are used to enhance CNN local features using the attention mechanism, Fig. 5(a). Different from the backbone network which processes a whole image, the detector requires to handle multiple proposal regions. Compared with the add operation, the attention mechanism facilitates the alignment of proposal regions with ground-truth objects.

Cross-Attention. The multi-head cross-attention [12] is a natural mechanism to perform feature coupling. Given query embeddings $\mathbf{Query} \in \mathbb{R}^{N \times C}$ to be enhanced and feature maps $\mathbf{Key}, \mathbf{Value} \in \mathbb{R}^{H \times W \times C}$ to be coupled, where H and W are the height and width of the feature maps. The cross-attention operation is defined as

$$\hat{\mathbf{Q}} = \text{Softmax} \left((\mathbf{Q} \mathbf{W}_q) (\mathbf{K} \mathbf{W}_k)^\top / \sqrt{C/h} \right) (\mathbf{V} \mathbf{W}_v), \quad (16)$$

where $\hat{\mathbf{Q}}$ is the enhanced query embeddings. W_q, W_k and W_v respectively denote parameters of the linear transformation layers. \top is a transpose operator and h the number of attention heads. Accordingly, given a proposal token $e_i \in \mathbb{R}^{1 \times C}$ and its corresponding CNN local features $f_i \in \mathbb{R}^{S^2 \times C}$, the feature coupling on detector head is defined as

$$\hat{f}_i = \text{Softmax} \left((f_i W_q) (e_i W_k)^\top / \sqrt{C/h} \right) (e_i W_v), \quad (17)$$

where \hat{f}_i denotes the enhanced feature maps. $A_i = \text{Softmax}((f_i W_q)(e_i W_k)^\top / \sqrt{C/h}) \in \mathbb{R}^{S^2 \times 1}$ is the attention matrix reflecting the correlation between the f_i and e_i . (17) defines a feature coupling procedure, which leverages each feature vector within the feature maps to query a proposal token so that the long-range semantic dependency can be embedded to CNN local features.

Augmented Cross-Attention. Considering that the tokens are sparse while the CNN local features are dense, a token augmentation procedure is proposed to increase the spatial versatility of sparse tokens so that local features couple with the optimal tokens. In specific, we propose to augment each proposal token to multiple (I) proposal tokens ($e_{\phi_i} \in \mathbb{R}^{I \times C}$) using a linear layer, as

$$e_{\phi_i} = \mathcal{F}(e_i). \quad (18)$$

By augmenting e_i to e_{ϕ_i} , feature coupling defined by (17) is rewritten as

$$\hat{f}_i = \text{Softmax} \left((f_i W_q) (e_{\phi_i} W_k)^\top / \sqrt{C/h} \right) (e_{\phi_i} W_v), \quad (19)$$

and the attention matrix A_i is augmented to multiple attention matrices $A_{\phi_i} \in \mathbb{R}^{S^2 \times I}$, each of which corresponds to a spatial location. After feature coupling, a convolutional layer is carried out to convert the feature maps \hat{f}_i to a feature vector for further fusing with proposal token e_i , Fig. 5(a).

C. Detector Loss

Following transformer-based detectors [12], [53], [73], we utilize the bipartite matching strategy to assign predicted proposals \mathbf{R} to ground-truth objects. The ground-truth objects are denoted as $\mathbf{G} = \{g_k, y_k\}_{k=1}^K$, where g_k and y_k respectively denote the bounding box and the one-hot label of the k -th object. K denotes the category number of the objects. A bipartite matching cost is minimized with the Hungarian algorithm [12], which guarantees that ground-truth objects are optimally assigned to proposals. The proposals that do not assigned to any object categorized as negatives. The proposal prediction loss with respect to a matched ground-truth object is defined as the

TABLE II

TOP-1 ACCURACY FOR IMAGE CLASSIFICATION ON THE IMAGENET VALIDATION SET. LATENCY AND PEAK GPU MEMORY (#MEMS) ARE MEASURED WITH BATCH SIZE 16 USING THE TIMM CODEBASE [65]

Models	#Params (M)	MACs (G)	Latency (ms)	#Mems (M)	Top-1 (%)
<i>CNN-based</i>					
ResNet-50 [4]	25.6	4.1	10.5	2528	76.2
ResNet-101 [4]	44.5	7.8	20.8	289	77.4
RegNetY-4.0GF [66]	20.6	4.0	23.0	2771	78.8
RegNetY-12.0GF [66]	51.8	12.1	20.95	11442	80.3
ResNet-50(timm) [67]	25.6	4.1	10.5	2528	80.4
ResNet-101(timm) [67]	44.5	7.8	20.8	289	81.5
EfficientNet-B4 [68]	19	4.2	25.4	2887	82.9
EfficientNet-B5 [68]	30	10	30.4	3767	83.7
EfficientNetV2-S [69]	22	8.8	28.0	1038	83.9
<i>Transformer-based</i>					
ViT-B ₃₈₄ [8]	86	55.5	72.5	1036	77.9
ViT-L ₃₈₄ [8]	307	191.1	194.4	2438	76.5
DeiT-S [10]	22.1	4.6	8.6	202	79.8
DeiT-B [10]	86.6	17.6	16.7	634	81.8
DeiT III-S [70]	22.1	4.6	8.6	202	81.4
DeiT III-B [70]	86.6	17.6	16.7	634	83.8
T2T-ViT _t -14 [11]	21.5	5.2	12.6	200	80.7
T2T-ViT _t -19 [11]	39.0	8.4	16.3	297	81.4
T2T-ViT _t -24 [11]	64.1	13.2	21.7	447	82.2
MViT-S [39]	26.1	32.9	33.1	382	76.0
MViT-B [39]	36.6	70.5	49.0	446	78.4
PVT-Small [40]	24.5	3.8	16.6	872	79.8
PVT-Medium [40]	44.2	6.7	28.6	463	81.2
PVT-Large [40]	61.4	9.8	41.9	529	81.7
Swin-T [17]	29	4.5	11.6	414	81.3
Swin-S [17]	50	8.7	21.9	496	83.0
Swin-B [17]	88	15.4	23.2	743	83.5
<i>Hybrid Model</i>					
BoT-S1-50 [33]	20.8	4.27	N/A	N/A	77.7
CvT-13 [71]	20	4.5	29.0	416	81.6
CvT-21 [71]	32	7.1	59.8	462	82.5
CoAtNet-0 [47]	25	4.2	N/A	N/A	81.6
CoAtNet-1 [47]	42	8.4	N/A	N/A	83.3
LeViT-384 [45]	39.1	2.35	12.2	513	82.6
ConvNeXt-T [72]	28	4.5	31.9	359	82.1
ConvNeXt-S [72]	50	8.7	59.2	442	83.1
Conformer-Ti (ours)	23.1	5.7	31.7	588	82.3
Conformer-S (ours)	37.7	10.7	34.5	1088	83.6
Conformer-B (ours)	83.3	23.5	35.5	3099	84.1

N/A means the codes or model weights are not accessible.

weighted summation of the classification loss $\mathcal{L}_{cls}(s_i, y_k)$ (the focal loss [74]), the object localization loss $\mathcal{L}_{L1}(\hat{b}_i, g_k)$ [74] and the GIoU loss $\mathcal{L}_{giou}(\hat{b}_i, g_k)$ [75], as

$$\mathcal{L}_+ = \lambda_1 \mathcal{L}_{cls}(s_i, y_k) + \lambda_2 \mathcal{L}_{L1}(\hat{b}_i, g_k) + \lambda_3 \mathcal{L}_{giou}(\hat{b}_i, g_k), \quad (20)$$

where λ_1, λ_2 and λ_3 are regularization factors, which are respectively set to 2, 5 and 2. The negative proposals only calculate classification loss, as $\mathcal{L}_- = \lambda_1 \mathcal{L}_{cls}(s_i, y_k)$. The overall loss for detector training is defined as $\mathcal{L} = \mathcal{L}_+ + \mathcal{L}_-$.

V. EXPERIMENTS

A. Image Classification

Conformer is trained on the ImageNet-1k [76] training set with 1.3M images and tested upon the validation set. The Top-1 accuracy is reported in Table II. The model is trained for 300 epochs with the AdamW optimizer [77], batchsize 1024 and weight decay 0.05. The initial learning rate is set to 0.001 and decay in a cosine schedule. For higher performance, we follow the data augmentation and regularization techniques in DeiT [10].

TABLE III

PERFORMANCE UNDER PARAMETER PROPORTIONS

E	d_h	#P	CNN branch		p_p	MACs	Acc.(%)	
			n_c	C				
384	6	22 M	-	-	-	4.6 G	79.8	
			2	64	1.5 M	0.07	5.2 G	81.3
				128	4.5 M	0.2	6.4 G	82.3
				192	9.3 M	0.4	8.2 G	82.8
				256	15.7 M	0.7	10.7 G	83.6
			320	23.7 M	1.0	13.7 G	83.6	
4	192	15.8 M	0.7	10.9 G	83.3			
3	256	21.4 M	1.0	13.0G	83.7			
576	9	48.9 M	-	-	-	10.0 G	79.0	
			2	256	16.4 M	0.3	16.3 G	83.6
			384	36.4 M	0.7	23.3 G	84.1	
768	12	86 M	-	-	-	17.6 G	81.8	
			2	256	17.6 M	0.2	24.2 G	83.0

E and d_h respectively denote the embedding dimensions and the head number in the attention module. C and n_c respectively denote the channels of c_2 and the bottleneck number within each convolution block in CNN. p_p is the proportion of CNN (including stem and FCUs) and transformer branch parameters. #P denotes the number of parameters.

1) *Performance*: Under similar parameters and computational budgets, Table II, Conformers outperform both CNN and vision transformers. For example, Conformer-S (with 37.7M parameters and 10.7G MACs) respectively outperforms ResNet-152 (with 60.2M parameters and 11.6G MACs) by 5.3% (83.6% versus. 78.3%) and DeiT-B (with 86.6M parameters and 17.6G MACs) by 1.8% (83.6% versus. 81.8%). Conformer-B, with comparable parameters and moderate MAC cost, outperforms DeiT-B by 2.3% (84.1% versus. 81.8%). Particularly, under comparable parameters and MACs, Conformer-B outperforms Swin-B by 0.6% (84.1% versus. 83.5%), which is a significant margin demonstrating importance of merging features from two architectures.

2) *Ablation Studies: Number of Parameters*. The parameters of Conformer are the summation of the CNN and transformer parameters while the parameters of FCU is negligible. The parameter proportion of the two branches is a hyper-parameter to be experimentally determined. In Table III, we evaluate performance of the two branches under different parameter settings. For the CNN branch, we tune the parameters by changing the channels and the number of bottlenecks, which respectively control the width and depth of CNN. For the transformer branch, we tune the parameters by changing the head numbers and embedding dimensions. From Table III, one can see that the accuracy is improved by increasing either parameters of the CNN or the transformer branch. More CNN parameters bring greater improvement while the computational cost overhead is lower. Empirically, the parameter proportion of the transformer and CNN branches fall into [1:1, 5:1]. While small Conformer models prefer a small parameter proportion 1:1, large Conformer models tend to use a large parameter proportion.

Feature Coupling Units. In Table IV, we evaluate the numbers of FCUs and the feature fusion strategies. Simply adding the CNN branch to the transformer branch (*i.e.*, DeiT-S) boosts the performance from 79.8% to 80.8%. When the feature fusion is activated, Conformer enjoys a significant gain (83.4% versus. 80.8%), indicating the potential of coupling features from two branches. When updating the simple adding operation in FCUs

TABLE IV
ABLATION STUDY OF STEP-BY-STEP CONSTRUCTION OF CONFORM FROM ViT

Component	#Params (M)	MACs (G)	Accuracy (%)
Trans. branch	22.1	4.6	79.8
+ CNN branch	36.3	10.0	80.8
+ 3 FCUs	36.7	10.2	82.3
+ 6 FCUs	37.0	10.3	83.0
+ 11 FCUs	37.7	10.6	83.4
+ OFF	37.7	10.7	83.6

‘OFF’ denotes orthogonal feature fusion.

TABLE V
COMPARISON OF HYBRID STRUCTURES. DEiT-S/32 MEANS THE MODEL WITH PATCH SIZE 32×32 [10]

Model	#Params	MACs	Accuracy
DeiT-S/32	22.9 M	1.1 G	73.8%
ResNet-26d & DeiT-S	36.5 M	3.7 G	80.2%
ResNet-50d & DeiT-S	46.0 M	5.5 G	80.4%
Conformer-S/32	38.8 M	7.0 G	81.9%

ResNet-26/50d is the variant of ResNet-26/50, and its stem module is composed of three 3×3 convolutions.

TABLE VI
COMPARISON OF POSITIONAL EMBEDDINGS STRATEGIES

Method	Positional embeddings	Accuracy
DeiT-S	✓	79.8%
	✗	77.4% (-2.4%)
Conformer-S	✓	83.7%
	✗	83.6% (-0.1%)

to orthogonal feature fusion, the performance boosts from 83.4% to 83.6%.

Dual Structure. Conformer is a dual model, which is essentially different from the serial hybrid ViT (CNN \rightarrow Transformer) [8]. In Table V, ResNet-26/50d & DeiT-S is a hybrid model which consists of ResNet-26/50d [4] and DeiT-S [10], where DeiT-S construct tokens upon the feature maps extracted by ResNet-26/50d. With a comparable computational cost overhead, Conformer-S/32 outperforms the serial hybrid model, which validates the advantage of the dual structure. Furthermore, such a dual structure is compatible to both CNN-based and transformer-based down-stream tasks (*i.e.*, object detection or instance segmentation). In contrast, the serial hybrid model, without specialized, is only compatible to transformer-based down-stream tasks.

Positional Embeddings. Considering that the CNN branch encodes both local features and spatial location information, the positional embeddings are assumed no longer required for Conformer. In Table VI, when the positional embedding is removed, the accuracy of DeiT-S decreases 2.4%, while that of Conformer-S decreases only 0.1%.

TABLE VII
COMPARISON OF DOWN/UPSAMPLING STRATEGIES

Down	Up	#Params	MACs	Accuracy
Maxpooling	Interpolation	37.7 M	10.7 G	83.5%
Avgpooling	Interpolation	37.7 M	10.7 G	83.6%
Convolution	Interpolation	47.7 M	12.3 G	83.6%
Attention	Attention	39.4 M	11.5 G	83.5%

TABLE VIII
PERFORMANCE COMPARISON OF ENSEMBLE MODELS

Model	#Params	MACs	Acc ^{C_n}	Acc ^{T_r}	Acc ^{All}
DeiT-S	22.0 M	4.2 G	-	79.8%	79.8%
ResNet-101	44.5 M	7.8 G	80.6%	-	80.6%
DeiT-S + ResNet-101	66.5 M	11.2 G	80.6%	79.8%	81.8%
Conformer-S	37.7 M	10.7 G	83.5%	83.3%	83.6%

Acc^{C_n} and Acc^{T_r} respectively denote the accuracy of the CNN and transformer branches.

TABLE IX
PERFORMANCE OF CONFORMER SUB-STRUCTURES FIG. 3

Sub-structure	#Params (M)	MACs (G)	Accuracy (%)
Fig. 3(b)	8.6	9.2	73.9
Fig. 3(c)	37.0	10.8	80.8
Fig. 3(d)	22.1	4.6	79.8
Fig. 3(e)	28.9	6.0	80.2
Conformer-S	37.7	10.7	83.6

Sampling Strategies. In FCU, to make CNN-based feature maps coupling with transformer-based patch tokens, up/down-sampling operations are used to spatially align them. In Table VII, we compare different up/down-sampling strategies including Maxpooling, Avgpooling, convolution and attention-based sampling. Compared with Max/Avgpooling sampling, convolution and attention-based sampling methods that use more parameters and have computational cost achieve comparable accuracies.

Comparison With Ensemble Models. Conformer is compared with the ensemble models combining the outputs of CNN and transformer, Table VIII. For a fair comparison, we use the same data augmentation and regularization strategies and the same epoch number (300) to train ResNet-101 [4], and combine it with the DeiT-S [10] model to construct an ensemble model. The accuracies of the CNN branch, the transformer branch, and the Conformer-S respectively reach 83.3%, 83.1%, and 83.6%. In contrast, the ensemble model (DeiT-S+ResNet-101) archives 81.8%, which is 1.8% lower than that of Conformer-S (83.6%), although it uses more parameters and MACs.

3) *Analysis: Structure Analysis.* As the analyzed in Section III.D, by considering FCUs as short connections the dual structure of Conformer is equivalent to a serial structure. Under different residual connections, Conformer can degenerate to various sub-structures. We sample several sub-structures and report the corresponding performance in Table IX. One can see that the residual structure outperforms other sub-structures.

Feature Analysis. Figs. 1 and 4 demonstrate that Conformer is effective to extract local features and global representations, which facilitates enhancing the localization capability of

TABLE X
 PERFORMANCE OF WEAKLY SUPERVISED OBJECT LOCALIZATION ON
 CUB-200-2011 TEST SET

Method	Backbone	Top-1 Loc.Acc.	GT-known Loc.Acc.
RCAM [78]	VGG16 [2]	59.0%	76.3%
TS-CAM [37]	DeiT-S [10]	71.3%	87.7%
	Conformer-S	72.0%	93.4%

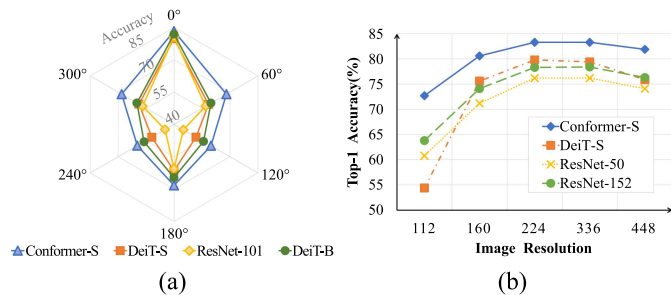


Fig. 6. Generalization capability. (a) Comparison of rotation invariance. The compared models are trained under the same data augmentation settings and directly evaluated on rotated images without model finetuning. (b) Comparison of scale invariance. The models are trained on images with the resolution of 224×224 , and tested on different image resolutions without model finetuning.

models. Weakly-supervised object localization (WSOL), which solely uses image-level category information as supervision signals but requires to learn complete object extent, is a touchstone for the localization capability of representation models. The quantitative experiment is conducted on the CUB-200-2011 dataset [83] and the results are reported in Table X. It can be seen that the localization performance of TS-CAM [37] with Conformer-S significantly outperforms those of CNN-based RCAM [78] and transformer-based DeiT-S, which further validates that Conformer is competent to learn long-range semantic dependency facilitating object localization.

Generalization Capability. To verify Conformer’s generalization capability in terms of object orientations and scales, we rotate test images by 0° , 60° , 120° , 180° , 240° and 300° and evaluate the performance of models trained under same data augmentation settings. As shown in Fig. 6(a), all models report comparable performance for images without rotation (0°). For the rotated test images, the performance of ResNet-101 drops significantly. In contrast, Conformer-S reports higher performance, which implies stronger rotation invariance. In Fig. 6(b), we compare the scale adaptation ability of Conformer with those of vision transformers (DeiT-S) and CNN (ResNet). We interpolate the positional embeddings of DeiT-S to adapt it to input images of different resolutions during inference. When the size of input images reduces from 224 to 112, DeiT-S’s performance drops by 25% and that of ResNet-50/152 drops by 15%. In contrast, the performance of Conformer drops only by 10%, demonstrating higher scale generalization capability of the learned feature representations.

Robustness Evaluation. In Table XI, we compare Conformer with ResNet and DeiT on four ImageNet validation sets,

 TABLE XI
 ROBUSTNESS EVALUATION ON IMAGENET VARIANTS

Models	ImageNet V2 [79]	ImageNet Adversarial [80]	ImageNet Rendition [81]	ImageNet Sketch [82]
ResNet-50	63.4	1.5	35.3	23.0
ResNet-101	65.5	5.0	38.7	26.4
DeiT-S	68.5	19.5	41.9	29.1
DeiT-B	70.9	28.6	44.6	31.9
Conformer-Ti	71.6	32.3	46.8	34.3
Conformer-S	73.5	38.0	48.8	37.7
Conformer-B	74.2	45.0	51.7	40.7

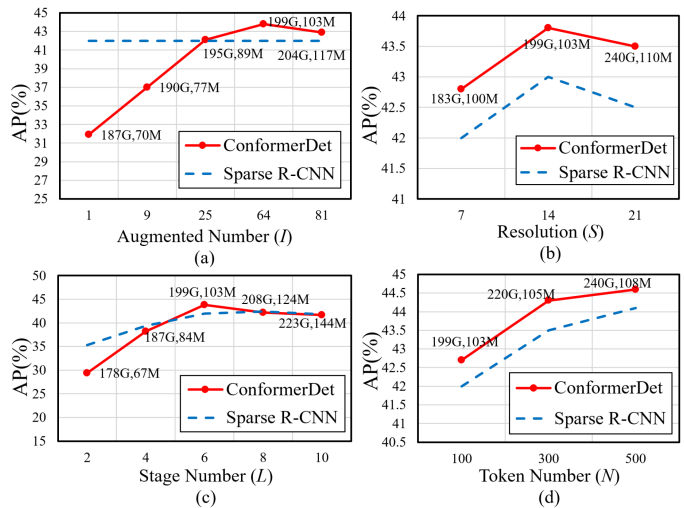


Fig. 7. Evaluation of hyper-parameters. (a) Augmented token number (I). (b) Resolution (S) of local features. (c) Stage number (L). (d) Token number (N). The numbers on the curves denote GFLOPs and model parameters.

i.e., ImageNetV2 [79], ImageNet-Adversarial [80], ImageNet-Rendition [81] and ImageNet-Sketch [82]. We train the models on the original ImageNet and test them on these validation sets. One can see that Conformer models consistently and significantly outperform ResNet and DeiT, indicating the superiority and robustness.

B. Object Detection

1) Experimental Settings: The MS COCO benchmark [90] has 80 object categories and contains 118k images for training, 5k images for validation and 20k images for testing. On the MS COCO benchmark, we use the standard AP metrics for object detection. All models are trained on the COCO train2017 split and evaluated on the val2017/test2017 split.

For MS COCO, data augmentation includes multi-scale variations in range of [480, 800] with stride 32 and random image crop [12]. The detectors are trained with the AdamW optimizer with a batch size 16 on 16 Tesla V100 GPUs. The detectors are trained with 36 epochs in total and learning rate is initialized as 2.5×10^{-5} , which is reduced by a magnitude after the 27th and 33th epochs. Following [73], we plug a self-attention module before each feature coupling module to further enhance proposal tokens.

2) Performance: In Table XII, ConformerDet is compared with the state-of-the-art detectors on the COCO validation split.

TABLE XII
COMPARISON WITH THE STATE-OF-THE-ART OBJECT DETECTORS ON COCO 2017 VALIDATION SPLIT

Method	Backbone	GFLOPs(G)	Param(M)	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS
<i>CNN-based detectors</i>										
Faster R-CNN FPN [84]	Conformer-S	405	54.2	44.2	-	-	28.5	48.1	58.4	20.7
PAA [85]	Conformer-S	-	-	46.5	65.7	50.5	30.0	50.7	62.0	16.3
DyHead [86]	ResNet-101	-	-	46.5	64.5	50.7	28.3	50.3	57.5	-
ATSS [87]	Swin-T	215	36	47.2	66.5	51.3	-	-	-	-
DyHead [86]	Swin-T	-	-	49.7	68.0	54.3	33.3	54.2	64.2	-
Cascade Mask R-CNN [88]	Conformer-S	782	93	49.9	69.6	54.5	30.1	53.3	65.5	12.7
<i>Transformer-based detectors</i>										
YOLOS [12]	DeiT-B	537	127	42.0	-	-	-	-	-	5.3
DETR-C5 [12]	ResNet-101	152	60	43.5	63.8	46.4	21.9	48.0	61.8	-
TSP-FCOS [89]	ResNet-101	255	70	44.4	63.8	48.2	27.7	48.6	57.3	-
DETR-DC5 [12]	ResNet-101	253	60	44.9	64.7	47.7	23.7	49.5	62.3	15.0
Anchor-DETR-DC5 [50]	ResNet-101	-	-	45.1	65.7	48.8	25.8	49.4	61.6	19.0
Conditional DETR-DC5 [51]	ResNet-101	262	63	45.9	66.8	49.5	27.2	50.3	63.3	12.0
Deformable DETR [36]	ResNet-101	173	40	46.2	65.2	50.0	28.8	49.2	61.7	17.0
SMCA-DETR [52]	ResNet-101	218	58	46.3	66.6	50.2	27.2	50.5	63.2	10.0
Sparse R-CNN [73]	ResNet-101	-	-	46.4	64.6	49.5	28.3	48.3	61.6	-
ViDT [49]	Swin-S	-	61	47.5	67.7	51.4	29.2	50.7	64.8	11.5
Sparse R-CNN [73]	Swin-T	172	110	47.9	67.3	52.3	-	-	-	-
Sparse R-CNN [73]	Conformer-S	212	120	48.8	68.4	53.1	31.8	52.1	63.8	16.8
ViDT [49]	Swin-B	263	100	49.2	69.4	53.1	30.6	52.6	66.9	9.0
<i>Dual architectural detectors</i>										
ConformerDet (ours)	Conformer-T	162	89	47.6	66.5	51.8	30.5	50.7	62.3	21.2
ConformerDet (ours)	Conformer-S	201	115	50.5	70.7	55.3	33.6	54.2	65.9	17.9
ConformerDet (ours)	Conformer-B	343	147	52.0	71.4	56.5	34.2	54.5	67.0	8.2
ConformerDet* (ours)	Conformer-S	-	115	52.1	71.7	56.3	34.1	54.8	66.9	-
ConformerDet* (ours)	Conformer-B	-	147	53.1	72.7	57.3	35.3	55.1	67.7	-

“*” Indicates multi-scale test.

TABLE XIII
ABLATION STUDIES OF CONFORMERDET MODULES

PP	RoIAlign	PR	ConformerDet	AP	Param(M)	FPS
✓				14.2	40.1	24.1
✓	✓			23.5	48.2	20.5
✓	✓	✓		26.5	67.0	22.1
✓	✓	✓	✓	43.8	115.0	18.3

With the Conformer-S backbone, ConformerDet outperforms Sparse R-CNN [73] (which also uses learnable sparse proposals) by 1.7% (50.5% versus 48.8%). This not only validates the plausibility of fusing CNN local features with transformer representation but also the iterative optimization strategy with proposal prediction and object feature enhancement.

ConformerDet outperforms ViDT [49] by 1.3% AP (50.5% versus 49.2%), despite it uses a larger backbone (Swin-B) pre-trained on ImageNet-22K [76]. On small objects (AP_s), ConformerDet significantly outperforms ViDT by 3.0% (33.6% versus 30.6%). This validates that it can leverage the details of CNN local features to improve the discriminability, as well as the long-range feature dependency to represent objects. As a transformer-based detector, ConformerDet is comparable to, if not better than, many transformer-based detectors including Deformable DETR [49], Conditional DETR [51] and ViDT [49], and powerful CNN-based detectors like ATSS [87] and Cascade Mask R-CNN [88].

In Table XIV, we compare ConformerDet with state-of-the-art detectors on the COCO test split. Equipped with Conformer-S and Conformer-B backbones, ConformerDet respectively

achieves 50.6% and 51.8% AP. With multi-scale testing, ConformerDet achieves 52.9% AP, which is on par with those of the state-of-the-art detectors.

3) *Ablation Studies*: Conformer-S is selected as the backbone network. The training schedule is 12 epochs, the short side of input image is set to 800 pixels and the number of proposal tokens (N) is set to 100.

Augmented Cross-Attention (ACU). In Fig. 7(a), we evaluate the augmented token number (I). For $I = 1$ (without token augmentation), ConformerDet reports a very low detection performance ($\sim 32\%$ AP). When using a proper number ($I = 64$) of augmented tokens, ConformerDet achieves the best AP by 43.8%. The large performance improvement validates the effectiveness of performing token augmentation (Section IV.B) when fusing the sparse tokens with dense CNN features.

Resolution of Local Features. As described in Section IV.B, the resolution (S) of CNN local features is related to the graininess of object representation. ConformerDet achieves the best performance 43.8% at $S = 14$, Fig. 7(b).

Number of Refinement Stages. In Section IV.A, iterative PP and ACU procedures progressively refine proposals in a learnable framework. In Fig. 7(c), the detection performance increases with the number (L) of refinement stages. When $L = 6$, the performance reaches the best.

Number of Proposal Tokens. This hyper-parameter has a significant impact on the detection performance. In Fig. 7(d), large token numbers imply higher performance, at larger computational and parameter costs. To balance the detection performance and computational cost, the numbers of proposal tokens are set to the range [100, 300]. The token number was set to 300.

TABLE XIV
COMPARISON WITH THE STATE-OF-THE-ART OBJECT DETECTORS ON COCO 2017 TEST SPLIT

Method	Backbone	Param(M)	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
CenterNet [91]	Hourglass-104	-	40.6	56.4	43.2	19.1	42.8	54.3
YOLO [92]	CSPDarkNet-53	27.6	47.5	66.2	51.7	28.2	51.2	59.8
Sparse R-CNN [73]	Conformer-S	37.7	48.6	67.9	52.9	31.7	52.0	63.5
Sparse R-CNN [73]	ResNeXt-64x4d-101-DCN	45.6	48.9	68.3	53.4	29.9	50.9	62.4
DyHead [86]	Conformer-S	37.7	49.5	67.7	54.2	33.1	53.9	64.1
Deformable DETR [36]	Conformer-S	37.7	50.0	70.0	54.4	30.4	52.3	65.1
Deformable DETR [36]	ResNeXt-64x4d-101-DCN	45.6	50.1	69.7	54.6	30.6	52.8	64.7
RelationNet++ [93]	ResNeXt-64x4d-101-DCN	45.6	50.3	69.0	55.0	32.8	55.0	65.8
ATSS* [87]	ResNeXt-64x4d-101-DCN	45.6	50.7	68.9	56.3	33.2	52.9	62.4
Sparse R-CNN* [73]	ResNeXt-64x4d-101-DCN	45.6	51.5	71.1	57.1	34.2	53.4	64.1
Deformable DETR* [36]	ResNeXt-64x4d-101-DCN	45.6	52.3	71.9	58.1	34.4	54.4	65.6
DyHead* [86]	ResNeXt-64x4d-101-DCN	37.7	54.0	72.1	59.3	37.1	57.2	66.3
ConformerDet(ours)	Conformer-S	37.7	50.6	70.3	55.2	33.5	54.3	65.8
ConformerDet(ours)	Conformer-B	83.3	51.8	71.2	56.3	34.1	54.5	66.7
ConformerDet*(ours)	Conformer-S	37.7	52.0	71.5	56.4	33.9	55.1	66.9
ConformerDet*(ours)	Conformer-B	83.3	52.9	72.4	57.1	35.1	55.1	67.5

“*” Indicates multi-scale test. “Param(M)” denotes parameter numbers of backbone networks.

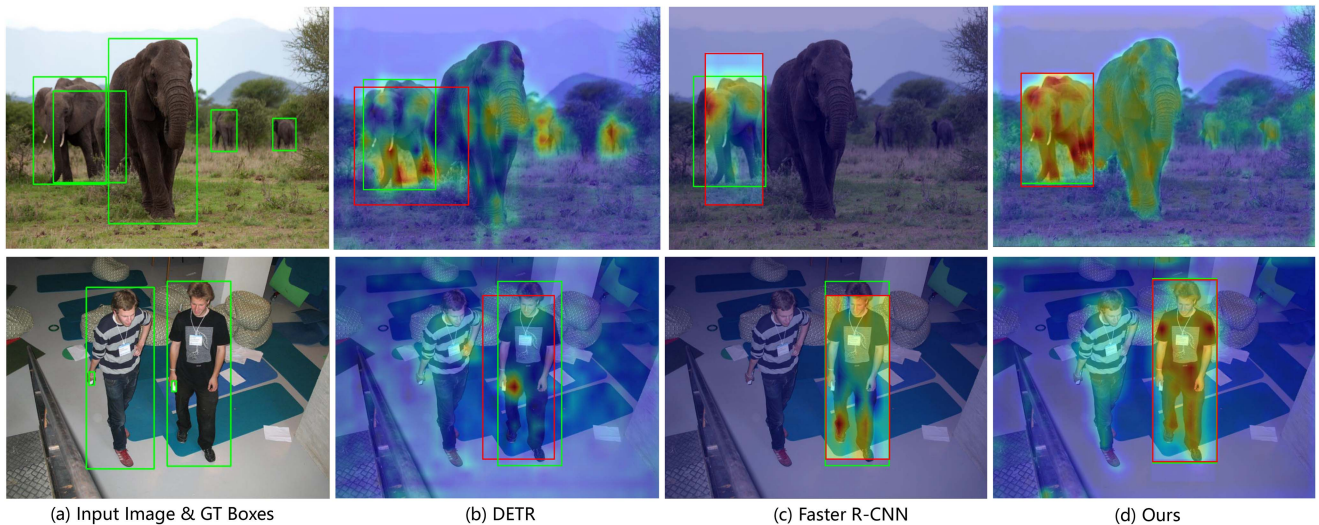


Fig. 8. Object detection responses (in red boxes). Compared with the transformer-based detector (DETR), ConformerDet activates more accurate regions due to the fine-detailed feature maps. Compared with the CNN-based detector (Faster R-CNN), ConformerDet can leverage the long-range semantic dependency to better activate objects from the same categories. *This figure is best viewed in color.*

Modules. In Table XIII, ablation studies validate the composed modules of ConformerDet, where “PP” denotes single-stage proposal prediction, “RoIAlign” predicted proposals with CNN features, “PR” single-stage proposal prediction with refinement, and “ConformerDet” the complete ConformerDet detector with multi-stage proposal refinement. Both the proposal prediction and proposal refinement modules significantly improve performance.

4) **Visualization Analysis:** In Fig. 8, the detection responses by three methods [12], [94] are compared. ConformerDet uniformly activate full object extent, as well as highlighting all object regions by leveraging the long-range semantic dependency. Compared with the transformer-based detector (DETR), ConformerDet activates more accurate regions due to the fine-detailed feature maps. Compared with the CNN-based detector

(Faster R-CNN), ConformerDet better activates objects, validating that it captures the long-range semantic dependencies between objects of same categories.

VI. CONCLUSION

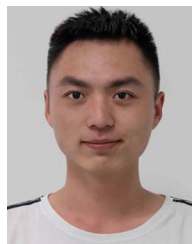
We propose Conformer, the First dual backbone to fuse CNN with vision transformer. Within Conformer, we leverage the convolution operators to extract local features and the self-attention mechanisms to capture global representations. We design the feature coupling unit to fuse local features and global representations, enhancing the ability of visual representations in an interactive fashion. Experiments show that Conformer, with comparable parameters and computation budgets, outperforms both CNNs and vision transformers, in striking contrast with

the state-of-the-arts. By further coupling the CNN features with global representations on the detector head, we proposed the ConformerDet approach, which outperformed the CNN and transformer counterparts with significant margins. The effectiveness of Conformer on image recognition and object detection tasks demonstrated the importance of merging features from CNN and transformer architectures.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [3] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [5] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.
- [6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
- [7] A. Vaswani et al., "Attention is all you need," 2017, *arXiv:1706.03762*.
- [8] A. Dosovitskiy et al., "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [9] B. Wu et al., "Visual transformers: Token-based image representation and processing for computer vision," 2020, *arXiv:2006.03677*.
- [10] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," 2020, *arXiv:2012.12877*.
- [11] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on imagenet," 2021, *arXiv:2101.11986*.
- [12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.
- [13] H. Chen et al., "Pre-trained image processing transformer," 2020, *arXiv:2012.00364*.
- [14] J. Beal, E. Kim, E. Tzeng, D. H. Park, A. Zhai, and D. Kislyuk, "Toward transformer-based object detection," 2020, *arXiv:2012.09958*.
- [15] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," 2020, *arXiv:2012.15840*.
- [16] Y. Jiang, S. Chang, and Z. Wang, "TransGAN: Two transformers can make one strong GAN," 2021, *arXiv:2102.07074*.
- [17] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.
- [18] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
- [19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 448–456.
- [20] Z. Peng et al., "Conformer: Local features coupling global representations for visual recognition," 2021, *arXiv:2105.03889*.
- [21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2017, pp. 4278–4284.
- [22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
- [23] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.
- [24] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," 2018, *arXiv:1810.12348*.
- [25] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.
- [26] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 636–644.
- [27] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.
- [28] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2018, pp. 3–19.
- [29] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.
- [30] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 1971–1980.
- [31] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3285–3294.
- [32] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3588–3597.
- [33] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," 2021, *arXiv:2101.11605*.
- [34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
- [35] T. B. Brown et al., "Language models are few-shot learners," 2020, *arXiv:2005.14165*.
- [36] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=gZ9hCDWe6ke>
- [37] W. Gao et al., "TS-CAM: Token semantic coupled attention map for weakly supervised object localization," 2021, *arXiv:2103.14862*.
- [38] M. Chen et al., "Generative pretraining from pixels," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 1691–1703.
- [39] H. Fan et al., "Multiscale vision transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 6804–6815.
- [40] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 548–558.
- [41] X. Yue et al., "Vision transformer with progressive sampling," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 377–386.
- [42] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, "Incorporating convolution designs into visual transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 559–568.
- [43] X. Chu et al., "Conditional positional encodings for vision transformers," 2021, *arXiv:2102.10882*.
- [44] S. D'ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "ConViT: Improving vision transformers with soft convolutional inductive biases," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 2286–2296.
- [45] B. Graham et al., "LeViT: A vision transformer in convNet's clothing for faster inference," *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 12239–12249.
- [46] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick, "Early convolutions help transformers see better," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 30392–30400.
- [47] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoatNet: Marrying convolution and attention for all data sizes," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 3965–3977.
- [48] A. El-Nouby et al., "XCiT: Cross-covariance image transformers," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 20014–20027.
- [49] H. Song et al., "ViDT: An efficient and effective fully transformer-based object detector," 2021, *arXiv:2110.03921*.
- [50] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query design for transformer-based detector," 2021, *arXiv:2109.07107*.
- [51] D. Meng et al., "Conditional DETR for fast training convergence," 2021, *arXiv:2108.06152*.
- [52] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of DETR with spatially modulated co-attention," 2021, *arXiv:2101.07448*.
- [53] Y. Fang et al., "You only look at one sequence: Rethinking transformer in vision through object detection," 2021, *arXiv:2106.00666*.
- [54] J. R. Pomerantz, "Are complex visual features derived from simple ones," *Formal Theoret. Vis. Percep.*, pp. 217–229, 1978.
- [55] L. Chen, "Topological structure in visual perception," *Science*, vol. 218, no. 4573, pp. 699–700, 1982.
- [56] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

- [57] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognit.*, vol. 24, no. 12, pp. 1167–1186, 1991.
- [58] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [59] D. A. Lisin, M. A. Mattar, M. B. Blaschko, E. G. Learned-Miller, and M. C. Benfield, "Combining local and global image features for object class recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2005, pp. 47–47. [Online]. Available: <https://dblp.org/db/conf/cvpr/cvprw2005.html#LisinMBLB05>
- [60] M. A. Islam, M. Kowal, S. Jia, K. G. Derpanis, and N. D. Bruce, "Position, padding and predictions: A deeper look at position information in CNNs," 2021, *arXiv:2101.12322*.
- [61] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
- [62] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," 2020, *arXiv:2005.00928*.
- [63] W. Gao et al., "TS-CAM: Token semantic coupled attention map for weakly supervised object localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2866–2875.
- [64] K. He, G. Gkioxari, P. Dollár, and B. R. Girshick, "Mask r-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
- [65] R. Wightman, "PyTorch image models," 2019. [Online]. Available: <https://github.com/rwightman/pytorch-image-models>
- [66] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," 2020, *arXiv:2003.13678*.
- [67] R. Wightman, H. Touvron, and H. J'egou, "ResNet strikes back: An improved training procedure in timm," 2021, *arXiv:2110.00476*.
- [68] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 6105–6114.
- [69] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 10 096–10 106.
- [70] H. Touvron, M. Cord, and H. J'egou, "DeiT III: Revenge of the ViT," 2022, *arXiv:2204.07118*.
- [71] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 22–31.
- [72] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convNet for the 2020s," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 11 966–11 976, 2022.
- [73] P. Sun et al., "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14449–14458.
- [74] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.
- [75] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.
- [76] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE Computer Society, 2009, pp. 248–255.
- [77] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
- [78] X. Zhang, Y. Wei, Y. Yang, and F. Wu, "Rethinking localization map: Towards accurate object perception with self-enhancement maps," 2020, *arXiv:2006.05220*.
- [79] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 5389–5400.
- [80] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15257–15266.
- [81] D. Hendrycks et al., "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 8320–8329.
- [82] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 10 506–10 518.
- [83] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200–2011 dataset," California Inst. Tech., Tech. Rep. CNS-TR-2011-001, 2011.
- [84] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.
- [85] K. Kim and H. S. Lee, "Probabilistic anchor assignment with IoU prediction for object detection," 2020, *arXiv:2007.08103*.
- [86] X. Dai et al., "Dynamic head: Unifying object detection heads with attentions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7369–7378.
- [87] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9756–9765.
- [88] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.
- [89] Z. Sun, S. Cao, Y. Yang, and K. Kitani, "Rethinking transformer-based set prediction for object detection," 2020, *arXiv:2011.10881*.
- [90] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [91] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
- [92] C. Wang, A. Bochkovskiy, and H. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13024–13033.
- [93] C. Chi, F. Wei, and H. Hu, "RelationNet++: Bridging visual representations for object detection via transformer decoder," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 13564–13574.
- [94] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.



Zhiliang Peng received the BS degree from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2019. Since 2019, he is currently working toward the PhD degree with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and machine learning, specifically for visual object detection and representation learning.



Zonghao Guo received the BS degree from Wuhan University, Wuhan, China, in 2019. Since 2019, he is currently working toward the PhD degree with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and machine learning, specifically for visual object detection and representation learning.



Wei Huang received the BS degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2019. Since 2019, she is currently working toward the PhD degree with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. Her research interests include computer vision and machine learning, specifically for visual object detection and representation learning.



Yaowei Wang (Member, IEEE) received the PhD degree in computer science from the University of Chinese Academy of Sciences, Beijing, China, in 2005. He is a tenured associate professor with Peng Cheng Laboratory, Shenzhen, China. From 2014 to 2015, he was an academic visitor with the Vision Lab of Queen Mary University of London. He has published more than 100 refereed journals and conference papers in the areas of machine learning and computer vision.



Lingxi Xie (Member, IEEE) received the BE and PhD degrees in engineering, both from Tsinghua University, in 2010 and 2015, respectively. He is currently a senior researcher with Huawei Inc. He also served as a postdoctoral researcher with the CCVL lab from 2015 to 2019, having moved from the University of California, Los Angeles to the Johns Hopkins University. His research covers image classification, object detection, semantic segmentation and other vision tasks. He is working towards automatically designing and optimizing deep learning models and training with limited annotation. He has published more than 40 papers in top-tier international conferences and journals. In 2015, he received the outstanding PhD thesis award from Tsinghua University.



Jianbin Jiao (Member, IEEE) received the BS, MS, and PhD degrees from the Harbin Institute of Technology (HIT), China, in 1989, 1992, and 1995, respectively. From 1997 to 2005, he was an Associate Professor with HIT. Since 2006, he has been a professor with the University of the Chinese Academy of Sciences, Beijing, China. In the research areas about image processing and pattern recognition. He has authored more than 50 papers in refereed conferences and journals.



Qi Tian (Fellow, IEEE) received the BE degree in electronic engineering from Tsinghua University in 1992, the MS degree in ECE from Drexel University in 1996, and the PhD degree in ECE from the University of Illinois at Urbana-Champaign (UIUC) in 2002. He is the chief scientist of Computer Vision with Huawei Noah's Ark Laboratory, a full professor with the Department of Computer Science, University of Texas at San Antonio (UTSA). He was a tenured Associate Professor from 2008-2012 and a tenure-track assistant professor from 2002-2008. During 2008-2009, he took one-year Faculty Leave with Microsoft Research Asia (MSRA) as lead researcher with the Media Computing Group. His research interests include multimedia information retrieval, computer vision, pattern recognition and published more than 360 refereed journal and conference papers. He was the coauthor of a Best Paper in ACM ICMR 2015. He received 2017 UTSA President's Distinguished Award for research Achievement, 2016 UTSA Innovation Award, 2014 Research Achievement Awards from College of Science, UTSA, 2010 Google Faculty Award, and 2010 ACM Service Award.



Qixiang Ye (Senior Member, IEEE) received the BS and MS degrees in mechanical and electrical engineering from the Harbin Institute of Technology, China, in 1999 and 2001, respectively, and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2006. He has been a professor with the University of Chinese Academy of Sciences since 2009, and was a visiting Assistant Professor with the Institute of Advanced Computer Studies (UMIACS), University of Maryland, College Park until 2013. His research interests include image processing, visual object detection and machine learning. He has published more than 100 papers in refereed conferences and journals including IEEE CVPR, ICCV, ECCV, NeurIPS, *IEEE Transactions on Neural Networks and Learning Systems*, and *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He is on the editorial boards of *IEEE Transactions on Circuit and Systems on Video Technology* and *IEEE Transactions on Intelligent Transportation Systems*.