LCTR: On Awakening the Local Continuity of Transformer for Weakly Supervised Object Localization

Zhiwei Chen[1], Changan Wang[2], Yabiao Wang[2], Guannan Jiang[3], Yunhang Shen[2], Ying Tai[2], Chengjie Wang[2], Wei Zhang[3], Liujuan Cao[1]*

[1]Media Analytics and Computing Lab, Department of Artificial Intelligence, School of Informatics, Xiamen University, China. [2]Tencent Youtu Lab, Shanghai, China. [3]CATL, China.
zhiweichen.cn@gmail.com, {changanwang, caseywang}@tencent.com, jianggn@catl.com, {odysseyshen, yingtai, jasoncjwang}@tencent.com, zhangwei@catl.com, caoliujuan@xmu.edu.cn

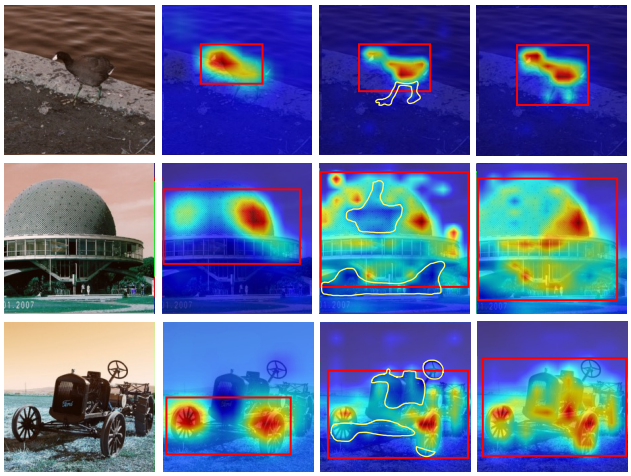
Abstract

Weakly supervised object localization (WSOL) aims to learn object localizer solely by using image-level labels. The convolution neural network (CNN) based techniques often result in highlighting the most discriminative part of objects while ignoring the entire object extent. Recently, the transformer architecture has been deployed to WSOL to capture the long-range feature dependencies with self-attention mechanism and multilayer perceptron structure. Nevertheless, transformers lack the locality inductive bias inherent to CNNs and therefore may deteriorate local feature details in WSOL. In this paper, we propose a novel framework built upon the transformer, termed LCTR (Local Continuity TRansformer), which targets at enhancing the local perception capability of global features among long-range feature dependencies. To this end, we propose a relational patch-attention module (RPAM), which considers cross-patch information on a global basis. We further design a cue digging module (CDM), which utilizes local features to guide the learning trend of the model for highlighting the weak local responses. Finally, comprehensive experiments are carried out on two widely used datasets, *i.e.*, CUB-200-2011 and ILSVRC, to verify the effectiveness of our method.

Introduction

Deep learning based methods have achieved unprecedented success in locating objects under a fully supervised setting (Liu et al. 2016; Bochkovskiy, Wang, and Liao 2020; Sun et al. 2021; Wang et al. 2021). However, these methods rely on a large number of bounding box annotations, which are expensive to acquire. Recently, the research on weakly supervised object localization (WSOL) has gained a significant momentum (Zhou et al. 2016; Zhang et al. 2018a; Gao et al. 2021) since it can learn object localizers using only image-level labels.

The pioneering work (Zhou et al. 2016) aggregated features from classification networks to generate class activation maps (CAM) for object localization. Unfortunately, image classifiers tend to focus only on the most discriminative features to achieve high classification performance. Therefore, the spatial distribution of feature responses may only cover the most discerning regions instead of the whole object range, which limits localization accuracy with large margins, as shown in Figure 1(b).



(a) Image (b) CAM (c) TS-CAM (d) LCTR (Ours)

Figure 1: Comparison of localization results on different methods: (a) Original images. (b) CNN-based method tends to be dominated by the most discriminative region. (c) Transformer-based method maintains coarse long-range dependencies while ignoring the local feature details (light yellow line). (d) The proposed LCTR not only considers finer local details but also retains global information. The predicted bounding boxes are in red. Best viewed in color.

To address this critical problem, many CAM-based approaches have been proposed, such as graph propagation (Zhu et al. 2017), data augmentation (Kumar Singh and Jae Lee 2017; Yun et al. 2019), adversarial erasing (Zhang et al. 2018a; Choe and Shim 2019; Chen et al. 2021) and spatial relation activation (Xue et al. 2019; Zhang, Wei, and Yang 2020; Guo et al. 2021). However, those approaches do alleviate the partial activation issue, but in a compromised manner — the essential philosophy behind it is first obtaining local features and then attempting to recover the non-salient regions to get full object extent. In fact, the fundamental root of this issue is determined by the intrinsic nature of convolution neural networks (CNNs). The CNN features

*Corresponding author.

with the local receptive field only capture small-range feature dependencies.

More recently, the transformer architecture (Vaswani et al. 2017) has been developed in the field of computer vision (Dosovitskiy et al. 2020; Wu et al. 2020; Yuan et al. 2021; Touvron et al. 2021; Jiang, Chang, and Wang 2021), which shows that pure transformers can be as effective in feature extraction for image recognition as CNN-based architectures. Notably, transformers with multi-head self-attention capture long-range dependencies, and retain more detailed information without downsampling operators, which naturally handles the partial activation problem in WSOL. TS-CAM (Gao et al. 2021) proposed token semantic coupled attention map from transformer structure, which captured long-range feature dependency among pixels for WSOL. However, transformer-based methods lack the locality inductive bias inherent to CNNs, ignoring the local information, which leads to weak local feature response on the target object, as shown in Figure 1(c). Therefore, how to precisely mine local features in global representations for WSOL still remains an open problem.

In this paper, we propose a novel Local Continuity TRansformer (LCTR) for discovering entire objects of interest via end-to-end weakly supervised training. The key idea of LCTR is to rearrange local-continuous visual patterns with global-connective self-attention maps, thereby bringing locality mechanism to transformer-based WSOL. To this end, we first propose a relational patch-attention module (RPAM) to construct a powerful patch relation map, which takes advantage of the patch attention maps under the guidance of a global class-token attention map. The RPAM maintains the cross-patch information and models a global representation with more local cues. Second, a cue digging module (CDM) is designed succinctly to induce the model to highlight the weak local features (*e.g.*, blurred object boundaries) by a hide-and-seek manner under a local view. In the CDM, to reward the weak response parts, we propose to employ the erased strategy, and induce the learnable convolutional kernels to be weighted by the weak local features. To validate the effectiveness of the proposed LCTR, we conduct a series of experiments on the challenging WSOL benchmarks.

Collectively, our main contributions are summarized as:

- We propose a simple LCTR for WSOL, which greatly enhances the local perception capability of global self-attention maps among long-range feature dependencies.
- We design a relational patch-attention module (RPAM) by considering cross-patch information, which facilitates global representations.
- We introduce a cue digging module (CDM) that encodes weak local features by learnable kernels to highlight the local details of global representations.
- LCTR achieves new state-of-the-art performance on CUB-200-2011 and ILSVRC dataset with 79.2% and 56.1% Top-1 localization accuracy, respectively.

Related Work

CNN-based Methods for WSOL. WSOL aims to learn object localizers with solely image-level supervision. There are many state-of-the-art methods based on the CNN structure. A representative pipeline of CNN-based WSOL is to aggregate deep feature maps with a class-specific fully connected layer to produce class attention maps (CAMs), from which final predicted bounding boxes are extracted (Zhou et al. 2016). Later on, the last fully connected layer is dropped for simplifying (Hwang and Kim 2016). Unfortunately, CAMs tend to be dominated by the most discriminative object part. Therefore, different extensions (Selvaraju et al. 2017; Chattopadhay et al. 2018; Xue et al. 2019; Zhang, Wei, and Yang 2020) have been proposed to improve the generation process of localization maps in order to recover the non-salient regions. HaS (Kumar Singh and Jae Lee 2017) and CutMix (Yun et al. 2019) adopted a random-erasing strategy from input images to force the classification networks to focus on relevant parts of objects. ACoL (Zhang et al. 2018a) introduced two adversarial classification classifiers to locate different object parts and discovered the complementary regions belonging to the same objects or categories. ADL (Choe and Shim 2019) further promoted the localization maps by applying dropout on multiple intermediate feature maps. Besides the erasing strategy, DANet (Xue et al. 2019) used a divergent activation method to learn better localization maps. SPG (Zhang et al. 2018b) and I^2C (Zhang, Wei, and Yang 2020) introduced the constraint of pixel-level correlations into the WSOL network. SPA (Pan et al. 2021) leveraged structure information incorporated in convolutional features for WSOL. Some other methods (*e.g.*, GC-Net (Lu et al. 2020), PSOL (Zhang, Cao, and Wu 2020), SPOL (Wei et al. 2021) and SLT-Net (Guo et al. 2021)) divided WSOL into two independent sub-tasks, including classification and the class-agnostic localization.

These studies alleviate the problem by extending from local activations to global ones in an implicit way, which is difficult to balance the image classification and the object localization. In fact, CNNs are prone to capture partial semantic features with local receptive fields, which belongs to the principal problem of CNNs. The problem of how to explore global cues from local receptive fields still exists. In this paper, we introduce a transformer-based structure, where the local-continuity and long-range feature dependencies can be simultaneously activated.

Transformer-based Methods for WSOL. The transformer model (Vaswani et al. 2017) is proposed to handle sequential data in the field of natural language processing. Recent studies also reveal its effectiveness for computer vision tasks (Dosovitskiy et al. 2020; Beal et al. 2020; Carion et al. 2020; Zheng et al. 2021; Hu et al. 2021). Since the local information extracted by the CNNs is deficient, various methods adopt the self-attention mechanism to capture the long-range feature dependencies. ViT (Dosovitskiy et al. 2020) applied the pure transformer directly to sequences of image patches for exploring spatial correlation on the image classification task. DETR (Carion et al. 2020) employed a transformer encoder-decoder architecture for the object detection task. As a pioneered work in WSOL, TS-CAM (Gao et al. 2021) proposed a semantic coupling strategy based on Deit (Touvron et al. 2021) to fuse the patch tokens with the semantic-agnostic attention map to achieve semantic-aware

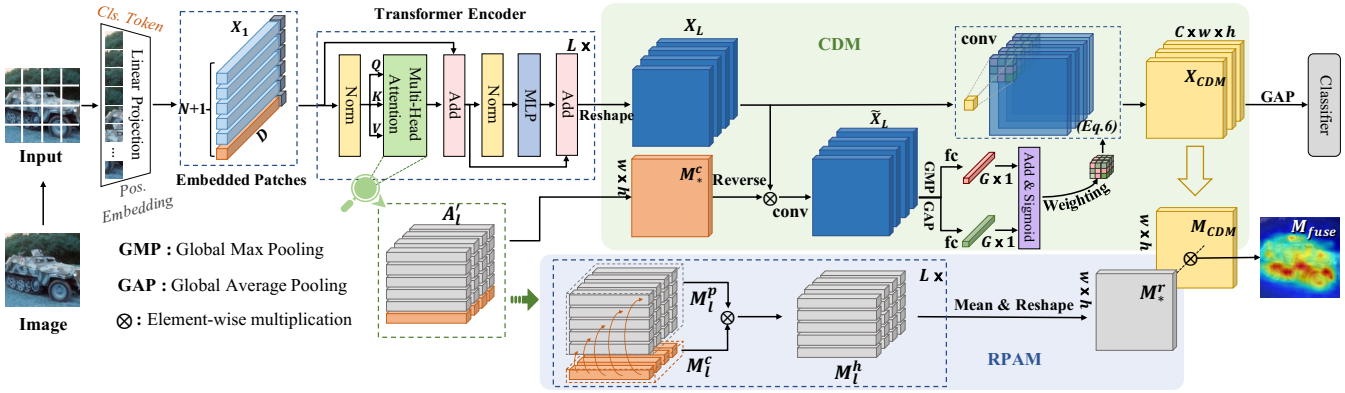


Figure 2: Overview of the proposed LCTR, which consists of vision transformer backbone for feature extraction, relational patch-attention module (RPAM) and cue digging module (CDM).

localization results, which has inspired many scholars to study transformer in weakly supervised object localization.

Despite of the progress, TS-CAM is relatively rough as it single-mindedly tries to get the long-range features but ignores the local information. Compared with the existing methods, our LCTR retains long-dependent features while mining for detailed feature cues based on the transformer structure for WSOL.

Methodology

We first present the overview of the proposed LCTR, then give a detailed description of RPAM and CDM, and finally incorporate them with the transformer structure in a joint optimization framework, as shown in Figure 2.

Overview

In accordance with the long-range info-preserving ability of the transformer architecture, LCTR is designed to offer precise localization maps for WSOL. We denote the input images as $I = \{(I_i, y_i)\}_{i=0}^{M-1}$, where $y_i \in \{0, 1, \ldots, C-1\}$ indicates the label of the image I_i, M and C are the number of images and classes, respectively. We first split I_i into N same-sized patches $x_p \in \mathbb{R}^{1\times D}$, where D denotes the dimension of each patch. We set $N = w \times h$, $w = W/P$ and $h = H/P$, where P is the width/height of a patch, H and W denote image height and width. For simplicity, we omit the mini-batch dimension. A learnable class token $x_{cls} \in \mathbb{R}^{1\times D}$ is embedded into the patches. These patches are flattened and linearly projected before being fed to L sequential transformer blocks, which can be formulated as:

$$X_1 = [x_{cls}; \mathcal{F}(x_p^1); \mathcal{F}(x_p^2); \cdots; \mathcal{F}(x_p^N)] + \mathcal{P}, \qquad (1)$$

where X_1 denotes the input of the first transformer block, $\mathcal{P} \in \mathbb{R}^{(N+1)\times D}$ is the position embedding and $\mathcal{F}$ is a linear projection. In particular, the proposed RPAM is employed in each transformer block to obtain a patch relation map M_*^r, which aggregates cross-patch information on a global basis.

Denote $X_L \in \mathbb{R}^{N\times D}$ as the output feature of the last transformer block. We reshape $X_L \in \mathbb{R}^{D\times w\times h}$ and apply the proposed CDM for further highlighting weak local responses. After that we obtain the feature map $\mathrm{X}_{CDM} \in \mathbb{R}^{C\times w\times h}$. Finally, the X_{CDM} are fed to a global average pooling (GAP) layer (Lin, Chen, and Yan 2013) followed by a softmax layer to predict the classification probability $p \in \mathbb{R}^{1\times C}$. The loss function is defined as

$$\mathcal{L} = -\log p. \qquad (2)$$

During testing, we extract the object map $M_{CDM} \in \mathbb{R}^{w\times h}$ from X_{CDM} according to the predicted class and obtain the final localization map by element-wise multiplication, given as

$$M^{fuse} = M_{CDM} \otimes M_*^r. \qquad (3)$$

The M^{fuse} is then resized to the same size as the original images by linear interpolation. For a fair comparison, we apply the same strategy detailed in CAM (Zhou et al. 2016) to produce the object bounding boxes.

Relational Patch-Attention Module

The proposed relational patch-attention module (RPAM) (Figure 3) strengthens the global feature representation from two stages: First, we utilize the attention vectors of the class token in the transformer block to generate a global class-token attention map. To fully exploit the feature dependencies of the transformer structure, we then use all the attention vectors of the patches containing the correlation between local features to generate a patch relation map under the guidance of the class-token attention map.

In the l-th transformer block, we hypothesize that the output feature map is $X_l \in \mathbb{R}^{(N+1)\times D}$. The attention matrix $A_l \in \mathbb{R}^{S\times(N+1)\times(N+1)}$ of multi-head self-attention module in the block is formulated as:

$$A_l = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}_l \cdot \boldsymbol{K}_l^{\top}}{\sqrt{D/S}}\right), \qquad (4)$$

where $\boldsymbol{Q}_l$ and $\boldsymbol{K}_l$ denote the queries and keys projected by X_l of self-attention operation in $(l-1)$-th transformer block, respectively. S represents the number of head and $\top$ is a transpose operator.

At this point, we first take the average operator to A_l based on S heads to obtain $A'_l \in \mathbb{R}^{(N+1)\times(N+1)}$. Then, the class-token attention vector $M_l^c \in \mathbb{R}^{1\times(N+1)}$ is extracted from A'_l. The M_l^c reveals how much each patch contributes to the object regions for image classification. Unfortunately, this map simply captures the global interactions of the class token to all patches, while ignoring the cross-patch correlations, which affects the modeling of local features. To remedy it, we take advantage of the patch attention map $M_l^p \in \mathbb{R}^{(N+1)\times N}$ in A'_l to structure a patch relation vector M_l^r under the guidance of M_l^c. The M_l^p learns the correlation between each patch but couldn't tell which one is more important. Therefore, we weight each patch attention map by multiplying M_l^p by M_l^c to obtain a new map $M_l^h \in \mathbb{R}^{(N+1)\times N}$. Note that M_l^c is reshaped ($\mathbb{R}^{(N+1)\times 1}$) before the multiplication. After that, we squeeze the first dimension of M_l^h to a vector ($M_l^r \in \mathbb{R}^{1\times N}$) by an average operation. The final patch relation map M_*^r is calculated by

$$M_*^r = \Gamma^{w\times h}(\frac{1}{L}\sum_l M_l^r), \tag{5}$$

where $\Gamma^{w\times h}(\cdot)$ indicates the reshape operator which coverts the vector ($\mathbb{R}^{1\times N}$) to the map ($\mathbb{R}^{w\times h}$).

The patch relation map M_*^r obtains the long-range dependencies that depends on the class-token attention vector M_l^c. Aggregating cross-patch information from the patch attention maps, M_*^r facilitates better global representations of the object without extra parameters in a simple way.

Cue Digging Module

RPAM considers the cross-patch information by using the class-token attention map from self-attention mechanism block of the transformer structure, but it is vulnerable if the transformer gets a poor class-token attention map. We thus further propose a cue digging module (CDM) to supply the long-range features based on a hide-and-seek manner.

Inspired by erasing-based methods that remove the most discriminative parts of the target object to induce the model to cover the integral extent of the object, we erase the object regions based on the global class-token attention map, leaving the weak response ones and the background. Then by weighting the learnable convolution kernels in the CDM on the basis of them, we shift part of the attention to object regions with weak responses. With the help of weighted kernels, we can highlight the local details as a supplement to the global representations.

Specifically, we convert the patch parts of class-token attention vectors to the map $\tilde{M}_*^c \in \mathbb{R}^{w\times h}$, and apply it to the feature map X_L by spatial-wise multiplication after being reversed. Note that $\tilde{M}_*^c$ is calculated by $\tilde{M}_*^c = \frac{1}{L}\sum_l M_l^c$. The feature map then passes through a convolutional layer to generate a new feature map $\tilde{X}_L \in \mathbb{R}^{D\times w\times h}$. Next, we score the features into G parts corresponding to G learnable convolution kernels. In particular, we apply two separate operators, the global average pooling and max pooling, to $\tilde{X}_L$. Then the feature maps are vectorized and sent to a fully connected layer, respectively. Besides, we add them together and apply a sigmoid function. In this light, we obtain the scores as $\{S_g \mid g = 1, 2, \ldots, G\}$. Finally, X_L passes through a convolutional layer with weighted convolution kernels by the scores S_g for encouraging the model to learn the object regions with weak responses, which can be formulated as

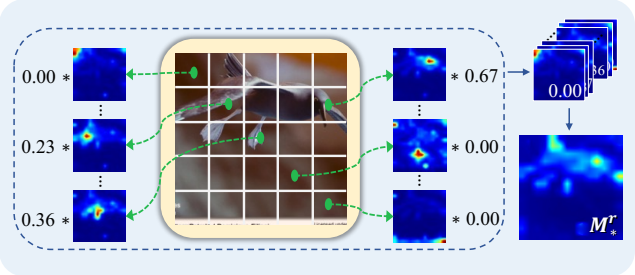


Figure 3: The RPAM aggregates all patch attention maps based on the scores (values) of the class-token attention map to learn local visual patterns.

$$X_{SGR} = X_L \sum_{g=1}^{G} S_g W_g^e, \tag{6}$$

where $W_g^e \in \mathbb{R}^{D\times C\times k_w\times k_h}$ denotes the kernel weights of the convolutional layer, which are initialized with kaiming uniform initialization (He et al. 2015). k_w, k_h represent the width and height of the kernels, respectively.

The convolution layer with the weighted kernels is applied to the global feature map X_L drawn from the transformer structure. Once the loss $\mathcal{L}$ in Eq. 2 is optimized, the weighted convolution kernels become more sensitive to the features (*i.e.*, the weak response features of object regions) that favor image classification. In this way, the model pays more attention to local cues and forms better global representations for the target object.

Experiments

Experimental Settings

Datasets. We evaluate the proposed methods on two challenging datasets, including CUB-200-2011 (Wah et al. 2011) and ILSVRC (Russakovsky et al. 2015). We only use image-level labels for training. CUB-200-2011 is a fine-grained bird dataset of 200 categories, which contains $5,994$ images for training and $5,794$ for testing. ILSVRC has about 1.2 million images in the training set and $50,000$ images in the validation set, with a total of $1,000$ different categories.

Evaluation Metrics. Following previous methods (Zhou et al. 2016; Russakovsky et al. 2015), we adopt the Top-1/Top-5 classification accuracy (Top-1/Top-5 *Cls.*), Top-1/Top-5 localization accuracy (Top-1/Top-5 *Loc.*) and localization accuracy with known ground-truth class (*Gt-k.*) as our evaluation metrics. Specifically, Top-1/Top-5 *Cls.* is correct if the Top-1/Top-5 predicted category contains the correct label. *Gt-k.* is correct when the intersection over union (IoU) between the ground-truth and the prediction is larger

Methods (Yr)	Backbone	Loc. Acc		
		Top-1	Top-5	Gt-k.
CAM ('16)	GoogLeNet	41.1	50.7	55.1
SPG ('18)	GoogLeNet	46.7	57.2	-
RCAM ('20)	GoogLeNet	53.0	-	70.0
DANet ('19)	InceptionV3	49.5	60.5	67.0
ADL ('19)	InceptionV3	53.0	-	-
PSOL ('20)	InceptionV3	65.5	-	-
SPA ('21)	InceptionV3	53.6	66.5	72.1
SLT-Net ('21)	InceptionV3	66.1	-	86.5
CAM ('16)	VGG16	44.2	52.2	56.0
ADL ('19)	VGG16	52.4	-	75.4
ACoL ('18)	VGG16	45.9	56.5	59.3
SPG ('18)	VGG16	48.9	57.2	58.9
DANet ('19)	VGG16	52.5	62.0	67.7
MEIL ('20)	VGG16	57.5	-	73.8
PSOL ('20)	VGG16	66.3	-	-
RCAM ('20)	VGG16	59.0	-	76.3
GC-Net ('20)	VGG16	63.2	-	-
SPA ('21)	VGG16	60.2	72.5	77.2
SLT-Net ('21)	VGG16	67.8	-	87.6
TS-CAM ('21)	Deit-S	71.3	83.8	87.7
LCTR (Ours)	Deit-S	**79.2**	**89.9**	**92.4**

Table 1: Localization accuracy on the CUB-200-2011 test set.

than 0.5, and does not consider whether the predicted category is correct. Top-1/Top-5 *Loc.* is correct when Top-1/Top-5 *Cls.* and *Gt-k.* are both correct.

Implementation Details. We adopt the Deit (Touvron et al. 2021) as the backbone network, which is pre-trained on ILSVRC (Russakovsky et al. 2015). Particularly, we replace the MLP head with our proposed CDM. Finally, a GAP layer and a softmax layer are added on the top of the convolutional layers. The input images are randomly cropped to 224×224 pixels after being resized to 256×256 pixels. We adopt AdamW (Loshchilov and Hutter 2017) with ϵ=$1e$-8, β_1=0.9, β_2=0.99 and weight decay of $5e$-4. On CUB-200-2011, we use a batch size of 128 with a learning rate of $5e$-5 to train the model for 80 epochs. For ILSVRC, the training process lasts 14 epochs with a batch size of 256 and a learning rate of $5e$-4. After meticulous experiments, we set G=4 in the CDM. All the experiments are performed with four Nvidia Tesla V100 GPUs using the PyTorch toolbox.

Comparison with the State-of-the-Arts

Localization. We first compare the proposed LCTR with the SOTAs on the localization accuracy on the CUB-200-2011 test set, as illustrated in Table 1. We observe that LCTR outperforms the baseline (*i.e.*, TS-CAM (Gao et al. 2021)) by 7.9% in terms of Top-1 *Loc.*, and is obviously superior to these CNN-based methods. Besides, Table 2 illustrates the localization accuracy on the ILSVRC validation set. It reports 0.4% performance improvement over the state-of-the-art SLT-Net (Guo et al. 2021).

Classification. Table 3 and Table 4 show the Top-1 and

Methods (Yr)	Backbone	Loc. Acc		
		Top-1	Top-5	Gt-k.
CAM ('16)	VGG16	38.9	48.5	-
ACoL ('18)	VGG16	45.8	59.4	63.0
CutMix ('19)	VGG16	42.8	54.9	59.0
ADL ('19)	VGG16	44.9	-	-
I^2C ('20)	VGG16	47.4	58.5	63.9
MEIL ('20)	VGG16	46.8	-	-
RCAM ('20)	VGG16	44.6	-	60.7
PSOL ('20)	VGG16	50.9	60.9	64.0
SPA ('21)	VGG16	49.6	61.3	65.1
SLT-Net ('21)	VGG16	51.2	62.4	67.2
CAM ('16)	InceptionV3	46.3	58.2	62.7
SPG ('18)	InceptionV3	48.6	60.0	64.7
ADL ('19)	InceptionV3	48.7	-	-
ACoL ('18)	GoogLeNet	46.7	57.4	-
DANet ('19)	GoogLeNet	47.5	58.3	-
RCAM ('20)	GoogLeNet	50.6	-	64.4
MEIL ('20)	InceptionV3	49.5	-	-
I^2C ('20)	InceptionV3	53.1	64.1	68.5
GC-Net ('20)	InceptionV3	49.1	58.1	-
PSOL ('20)	InceptionV3	54.8	63.3	65.2
SPA ('21)	InceptionV3	52.8	64.3	68.4
SLT-Net ('21)	InceptionV3	55.7	65.4	67.6
TS-CAM ('21)	Deit-S	53.4	64.3	67.6
LCTR (Ours)	Deit-S	**56.1**	**65.8**	**68.7**

Table 2: Localization accuracy on the ILSVRC validation set.

Methods (Yr)	Backbone	Cls. Acc	
		Top-1	Top-5
CAM ('16)	GoogLeNet	73.8	91.5
RCAM ('20)	GoogLeNet	73.7	-
DANet ('19)	InceptionV3	71.2	90.6
ADL ('19)	InceptionV3	74.6	-
SLT-Net ('21)	InceptionV3	76.4	-
CAM ('16)	VGG16	76.6	92.5
ACoL ('18)	VGG16	71.9	-
ADL ('19)	VGG16	65.3	-
DANet ('19)	VGG16	75.4	92.3
SPG ('18)	VGG16	75.5	92.1
MEIL ('20)	VGG16	74.8	-
RCAM ('20)	VGG16	75.0	-
SLT-Net ('21)	VGG16	76.6	-
TS-CAM ('21)	Deit-S	80.3	94.8
LCTR (Ours)	Deit-S	**85.0**	**97.1**

Table 3: Classification accuracy on the CUB-200-2011 test set.

Top-5 classification accuracy on the CUB-200-2011 test set and ILSVRC validation set, respectively. For the fine-grained recognition dataset CUB-200-2011, LCTR achieves remarkable performance of 85.0%/97.1% on Top1/Top-5 *Acc.*. In addition, LCTR obtains comparable results with SLT-Net (Guo et al. 2021) on Top-1 *Acc.* and surpasses other

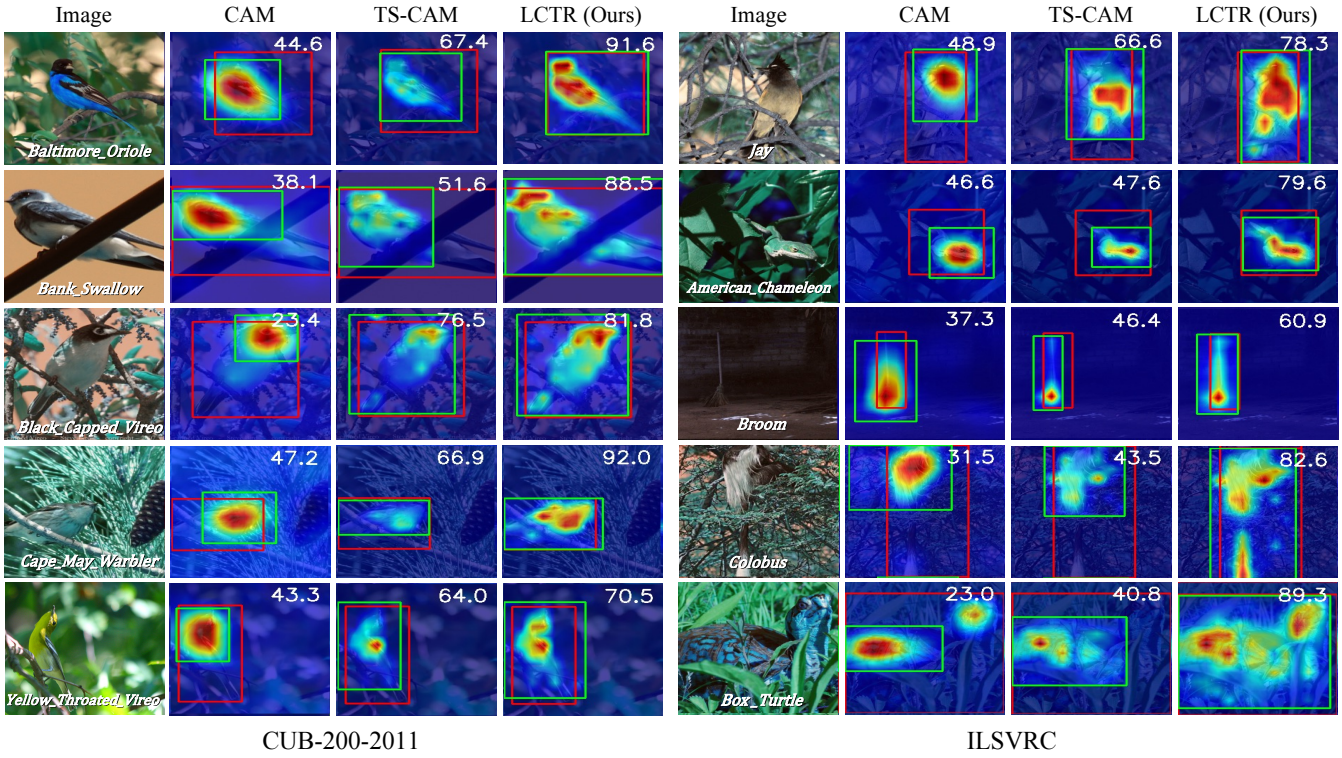


Figure 4: Visual comparisons of localization results on different methods. 1st Column: Input images. 2nd Column: Results of CAM based on CNNs. 3rd Column: Results of TS-CAM based on Transformer. 4th Column: Results of our LCTR. Note that the groundtruth bounding boxes are in red, the predictions are in green, and the IoU values (%) are shown in white text.

methods significantly on the ILSVRC validation set. Note that SLT-Net used a separated localization-classification framework, it cannot retain the global information for the objects in the individual classification network. To sum up, the proposed LCTR can greatly improve the quality of object localization while keeping high classification performance.

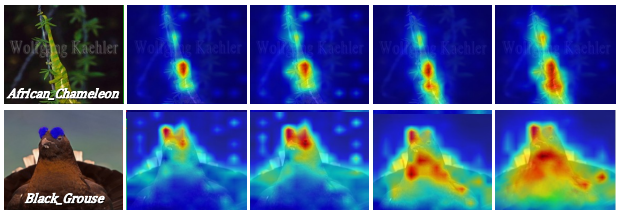
Visualization. For qualitative evaluation, Figure 4 visualizes the final localization results of CAM (Zhou et al. 2016) based on the CNNs, TS-CAM (Gao et al. 2021) based on the transformer and our method on CUB200-2011 and ILSVRC datasets. From the results, compared with the CAM, we consistently observe that our method can cover a more complete range of object regions instead of focusing only on the most discriminative ones. In addition, we capture more localized cues than the TS-CAM method, resulting in more accurate localization. For example, the tail regions of the *Bank Swallow* and the *Colobus* are ignored by CAM and TS-CAM methods, while our LCTR is able to aggregate more detailed features of the target object, which enhances the local perception capability of global features among long-range feature dependencies. Please refer to the supplementary materials for more visualized localization results of our method.

Methods (Yr)	Backbone	Cls. Acc	
		Top-1	Top-5
CAM ('16)	VGG16	68.8	88.6
ACoL ('18)	VGG16	67.5	88.0
I^2C ('20)	VGG16	69.4	89.3
MEIL ('20)	VGG16	70.3	-
RCAM ('20)	VGG16	68.7	-
SLT-Net ('21)	VGG16	72.4	-
CAM ('16)	InceptionV3	73.3	91.8
SPG ('18)	InceptionV3	69.7	90.1
ADL ('19)	InceptionV3	72.8	-
ACoL ('18)	GoogLeNet	71.0	88.2
DANet ('19)	GoogLeNet	63.5	91.4
RCAM ('20)	GoogLeNet	74.3	-
MEIL ('20)	InceptionV3	73.3	-
I^2C ('20)	InceptionV3	73.3	91.6
SLT-Net ('21)	InceptionV3	**78.1**	-
TS-CAM ('21)	Deit-S	74.3	92.1
LCTR (Ours)	Deit-S	77.1	**93.4**

Table 4: Classification accuracy on the ILSVRC validation set.

Ablation Studies

First, we visualize the localization maps with different settings in Figure 5. We observe that the RPAM strengthens the global representations of the baseline (Gao et al. 2021), *e.g.*, the tail-feature response of the *African chameleon* is en-



(a) Image (b) Baseline (c) RPAM (d) CDM (e) RPAM+CDM

Figure 5: Visualization of localization map with different settings. (a) Input images. (b) The baseline obtains coarse long-range dependencies. (c) The global representations are facilitated when applying the RPAM. (d) The local cues are rewarded with the CDM. (e) The global perception capability of the target object is fully exploited.

Applied Mode	Top-1 *Loc.*	*Gt-k.*	Top-1 *Cls.*
GMP-*fc*	75.6	88.7	84.7
GAP-*fc*	75.8	88.8	84.8
(GMP+GAP)-*fc*	75.8	89.2	84.9
(GMP-*fc*) + (GAP-*fc*)	**76.0**	**90.0**	**85.0**

Table 5: The effect of different type of classifier in the CMD on CUB-200-2011 test set. GMP/GAP denotes the global max/average pooling. *fc* is the fully connected layer.

G	Top-1 *Loc.*	*Gt-k.*	Top-1 *Cls.*
2	75.2	88.7	84.7
4	**76.0**	**90.0**	**85.0**
8	74.3	87.8	84.4
16	74.1	88.6	83.2
32	74.0	88.4	82.9

Table 6: The impact of the parameter G in the CDM on CUB-200-2011 test set.

Kernel Size ($k_w \times k_h$)	Top-1 *Loc.*	*Gt-k.*	Top-1 *Cls.*
1×1	73.5	88.8	82.5
3×3	**76.0**	**90.0**	**85.0**

Table 7: The impact of different kernel size in the CDM on CUB-200-2011 test set.

hanced, as it considers more cross-patch information. When only using CDM, we find that the local feature details are mined. For example, the abdominal features of the *Black grouse* are further activated compared to the baseline. By applying both RPAM and CDM, the final localization map (Figure 5 (e)) highlights the full object extent.

Next, we explore the concise design of the CDM. From the results on Table 5, we can observe that the mode of using separate *fc*s with GAP and GMP reports the best performance. These results also verify that GAP and GMP work differently in the CDM. Then, we evaluate the accuracy under different parameters G in the CDM, as shown in Table 6.

Methods	Dataset	RPAM	CDM	Top-1 *Loc.*	Top-1 *Cls.*
TS-CAM	CUB			71.3	80.3
TS-CAM*				73.1	81.6
LCTR	CUB	✓		74.0	81.6
			✓	76.0	**85.0**
		✓	✓	**79.2**	**85.0**
TS-CAM	ILSVRC			53.4	74.3
TS-CAM*				53.0	74.0
LCTR	ILSVRC	✓		54.2	74.0
			✓	55.1	**77.1**
		✓	✓	**56.1**	**77.1**

Table 8: Performance on both CUB-200-2011 test set and ILSVRC validation set when using different configurations. Note that * indicates the re-implement method.

From the experimental results, we observe that the best performance is achieved when $G = 4$. Setting a larger G leads to a larger number of parameters and degrades accuracy, which we believe is caused by overfitting. Besides, we examine the impact of the weighted kernel size (*i.e.*, $k_w \times k_h$). Results shown in Table 7 indicate that a kernel size of 3×3 yields better performance.

Lastly, we investigate the effect with different configurations on the accuracy, as reported in Table 8. On the CUB-200-2011 test set, we can see that RPAM increases the Top-1 *Loc.* by 0.9% compared with the baseline TS-CAM method. Note that the lightweight RPAM is directly applied in the test phase, so the classification performance remains unchanged. When applying the CDM to the network, we observe an improvement in both classification and localization performance. From this, we believe that the local cues captured by the CDM are important for both two tasks. The best localization/classification accuracy can be achieved when employing both RPAM and CDM. Meanwhile, we conduct the similar experiments on the ILSVRC validation set, which also validate the effectiveness of two modules, as shown in the lower part of Table 8.

Conclusion

In this paper, we propose a novel Local Continuity TRansformer, termed LCTR, for weakly supervised object localization, which induces the model to learn the entire extent of the object with more local cues. We first design a relational patch-attention module (RPAM), considering cross-patch information based on the multi-head self-attention mechanism, which gathers more local patch features for facilitating the global representations. Moreover, we introduce a cue digging module (CDM), which employs a hide-and-seek manner to wake up the weak local features for enhancing global representation learning. Extensive experiments show the LCTR can successfully mine integral object regions and outperform the state-of-the-art localization methods.

Acknowledgments

This work is supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No.U1705262, No.62072386, No.62072387, No.62072389, No.62002305, No.61772443, No.61802324 and No.61702136), Guangdong Basic and Applied Basic Research Foundation (No.2019B1515120049), the Natural Science Foundation of Fujian Province of China (No.2021J01002), and the Fundamental Research Funds for the central universities (No.20720200077, No.20720200090 and No.20720200091).

References

Beal, J.; Kim, E.; Tzeng, E.; Park, D. H.; Zhai, A.; and Kislyuk, D. 2020. Toward transformer-based object detection. *arXiv*.

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-Y. M. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*.

Chattopadhay, A.; Sarkar, A.; Howlader, P.; and Balasubramanian, V. N. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*.

Chen, Z.; Cao, L.; Shen, Y.; Lian, F.; Wu, Y.; and Ji, R. 2021. E2Net: Excitative-expansile learning for weakly supervised object Localization. In *ACMMM*.

Choe, J.; and Shim, H. 2019. Attention-based dropout layer for weakly supervised object localization. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*.

Gao, W.; Wan, F.; Pan, X.; Peng, Z.; Tian, Q.; Han, Z.; Zhou, B.; and Ye, Q. 2021. TS-CAM: Token semantic coupled attention map for weakly supervised object localization. *ICCV*.

Guo, G.; Han, J.; Wan, F.; and Zhang, D. 2021. Strengthen learning tolerance for weakly supervised object localization. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*.

Hu, J.; Cao, L.; Lu, Y.; Zhang, S.; Wang, Y.; Li, K.; Huang, F.; Shao, L.; and Ji, R. 2021. ISTR: End-to-end instance segmentation with transformers. *arXiv*.

Hwang, S.; and Kim, H.-E. 2016. Self-transfer learning for weakly supervised lesion localization. In *MICCAI*.

Jiang, Y.; Chang, S.; and Wang, Z. 2021. Transgan: Two transformers can make one strong gan. *arXiv*.

Kumar Singh, K.; and Jae Lee, Y. 2017. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. *arXiv*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *ECCV*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv*.

Lu, W.; Jia, X.; Xie, W.; Shen, L.; Zhou, Y.; and Duan, J. 2020. Geometry constrained weakly supervised object localization. In *ECCV*.

Pan, X.; Gao, Y.; Lin, Z.; Tang, F.; Dong, W.; Yuan, H.; Huang, F.; and Xu, C. 2021. Unveiling the potential of structure preserving for weakly supervised object localization. In *CVPR*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. In *IJCV*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *CVPR*.

Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. 2021. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *ICML*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset. *Computation & Neural Systems Technical Report*.

Wang, J.; Song, L.; Li, Z.; Sun, H.; Sun, J.; and Zheng, N. 2021. End-to-end object detection with fully convolutional network. In *CVPR*.

Wei, J.; Wang, Q.; Li, Z.; Wang, S.; Zhou, S. K.; and Cui, S. 2021. Shallow feature matters for weakly supervised object localization. In *CVPR*.

Wu, B.; Xu, C.; Dai, X.; Wan, A.; Zhang, P.; Yan, Z.; Tomizuka, M.; Gonzalez, J.; Keutzer, K.; and Vajda, P. 2020. Visual transformers: Token-based image representation and processing for computer vision. *arXiv*.

Xue, H.; Liu, C.; Wan, F.; Jiao, J.; Ji, X.; and Ye, Q. 2019. Danet: Divergent activation for weakly supervised object localization. In *ICCV*.

Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.; Tay, F. E.; Feng, J.; and Yan, S. 2021. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv*.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*.

Zhang, C.-L.; Cao, Y.-H.; and Wu, J. 2020. Rethinking the route towards weakly supervised object localization. In *CVPR*.

Zhang, X.; Wei, Y.; Feng, J.; Yang, Y.; and Huang, T. S. 2018a. Adversarial complementary learning for weakly supervised object localization. In *CVPR*.

Zhang, X.; Wei, Y.; Kang, G.; Yang, Y.; and Huang, T. 2018b. Self-produced guidance for weakly-supervised object localization. In *ECCV*.

Zhang, X.; Wei, Y.; and Yang, Y. 2020. Inter-image communication for weakly supervised localization. In *ECCV*.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*.

Zhu, Y.; Zhou, Y.; Ye, Q.; Qiu, Q.; and Jiao, J. 2017. Soft proposal networks for weakly supervised object localization. In *ICCV*.