An Entropy-guided Reinforced Partial Convolutional Network for Zero-Shot Learning

Yun Li, Zhe Liu, Lina Yao, Xianzhi Wang, Julian McAuley, Xiaojun Chang

***Abstract*—Zero-Shot Learning (ZSL) aims to transfer learned knowledge from observed classes to unseen classes via semantic correlations. A promising strategy is to learn a global-local representation that incorporates global information with extra localities (i.e., small parts/regions of inputs). However, existing methods discover localities based on explicit features without digging into the inherent properties and relationships among regions. In this work, we propose a novel Entropy-guided Reinforced Partial Convolutional Network (ERPCNet), which extracts and aggregates localities progressively based on semantic relevance and visual correlations without human-annotated regions. ERPCNet uses reinforced partial convolution and entropy guidance; it not only discovers global-cooperative localities dynamically but also converges faster for policy gradient optimization. We conduct extensive experiments to demonstrate ERPCNet's performance through comparisons with state-of-the-art methods under ZSL and Generalized Zero-Shot Learning (GZSL) settings on four benchmark datasets. We also show that ERPCNet is time efficient and explainable through visualization analysis.**

***Index Terms*—Zero-shot learning, reinforcement learning, image representation.**

I. INTRODUCTION

Zero-shot Learning (ZSL) mimics the human ability to perceive unseen concepts [1], [2]. In image classification, ZSL models should still work when only semantic descriptions of a class (i.e., attributes that describe the visual characteristics of an image, such as *the object is black*) are given. A typical scheme for ZSL is to extract visual representations from images and then learn visual-semantic associations [3]. However, approaches following this scheme often focus on global features while failing to capture subtle local differences between classes. Then they may fail to handle difficult tasks in the real-world applications, e.g., fine-grained image classification [4], where classes are highly similar. A few studies have paved the way to incorporate 'locality' knowledge, i.e., discriminative parts/regions in the original image, into global information [5]–[9]. These approaches are either annotation-based or weakly-supervised [5]–[7], [10]. Annotation-based methods [10]–[13] use extra annotations of important local regions to supervise the locality learning, although manual annotations are often time-consuming and costly to obtain. Weakly-supervised methods mitigate the challenge for labeled data acquisition by detecting salient local regions without ground-truth annotations. They adopt multi-attention [5], [14]–[17] or pre-defined strategies [6], [7], [18] to capture diverse localities.

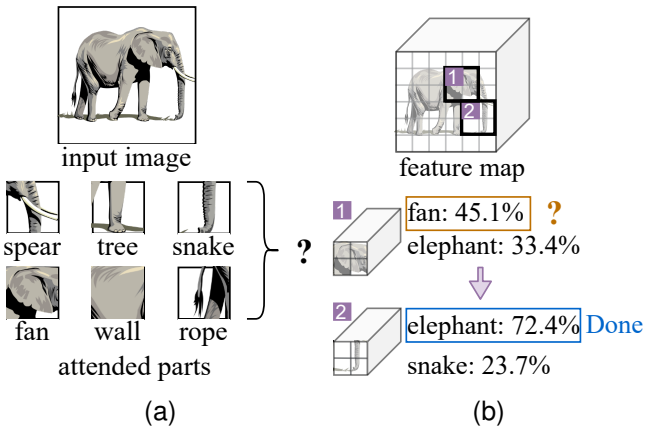


Fig. 1. Locality comparisons. (a) Conventional locality. (b) Progressive locality.

Overall, existing studies [5], [6], [12], [15], [19] only consider fixed numbers of localities while neglecting that different images may need different numbers of localities. The demand for locality exploration increases when the images are harder to classify. Thus, methods that use a fixed number of localities are inefficient and may introduce noise. Moreover, such methods [5], [14], [15], [20], [21] learn regions independently without accounting for inter-dependencies among regions, leading to poor performance on downstream tasks. As an example, Figure 1(a) shows a conventional deep learning version of the blind man and the elephant parable. In this example, six attention-maps/extractors each extract a different part as the locality and tend to identify the elephant as different objects (namely snake, spear, fan, wall, tree, and rope, respectively). Then, all the extracted localities will confuse the final classifier that aims to distinguish the elephant.

To address the above challenges, we introduce Reinforcement Learning (RL) to highlight localities based on region correlations progressively. Since it is challenging to train the reinforced model under weak supervision and to scale to real-world datasets [22], we learn localities at the level of abstraction hierarchies, i.e., convolution-level, to enable fast training. As shown in Figure 1(b), our model first selects an ear-related feature map and speculates the object could

Y. Li, Z. Liu, and L. Yao are with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: yun.li5@unsw.edu.au; zhe.liu4@unsw.edu.au; lina.yao@unsw.edu.au).

X. Wang is with the School of Computer Science, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: xianzhi.wang@uts.edu.au).

J. McAuley is with the School of Computer Science and Engineering, University of California San Diego, San Diego, California, United States (e-mail: jmcauley@eng.ucsd.edu).

X. Chang is with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail:cxj273@gmail.com)

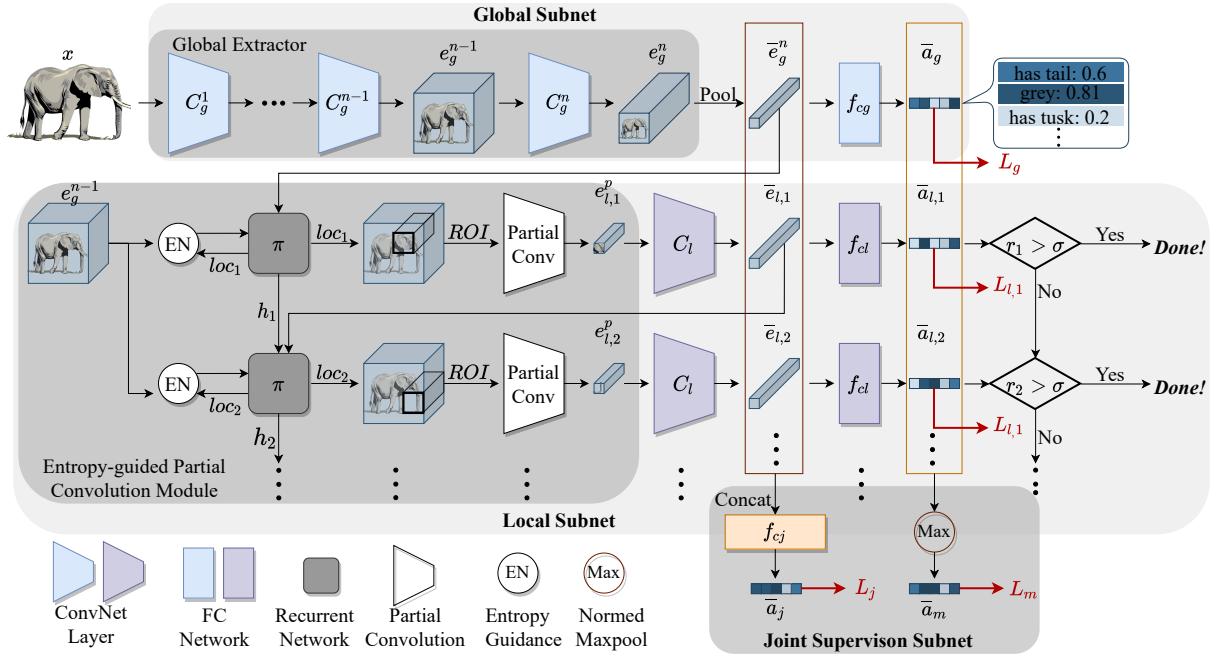


Fig. 2. Overview of ERPCNet. Given an input image x, the model extracts global embeddings $\overline{e}_g^n$ from x and progressively processes a sequence of local regions at the abstraction hierarchies located at $\{loc_1, loc_2, \dots\}$. At t-th step, ERPCNet conducts partial convolution and local extraction on the local region, as well as selecting the next location using an entropy-guided sampler π. The global and local visual features are fed into the corresponding predictors f_{cg} and f_{cl}, respectively, for zero-shot recognition. The local loss L_l guarantees the distinctiveness of a locality and its embedding $\overline{e}_{l,t}$. The joint supervision subnet optimizes the model to improve global and local cooperation (L_{jnt}) and strengthens divergence across localities (L_m). Locality selection terminates once a sufficient reward is obtained.

be a fan, an elephant, etc. Finally, the model chooses the nose-related feature map based on the former selection and recognizes the object as an elephant.

In particular, we propose a novel Entropy-guided Reinforced Partial Convolutional Network (ERPCNet) for effective global-local learning. It leverages RL to learn localities progressively based on semantic relevance and inherent relationships among regions. We design a reinforced partial convolution module to ease the sample efficiency problem for better RL optimization. Sample efficiency refers to the action amount needed for an RL agent to reach certain levels of performance. Partial convolution can reduce the action space by integrating RL in the abstraction hierarchies instead of at the conventional image level to allow fast training. Also, partial convolution is more efficient and flexible than processing image-level localities from scratch at each step. Besides, the entropy, introduced as expert knowledge, can complement the reward of the reinforced module to accelerate reward learning and improve performance.

Contributions. In summary, we make the following contributions in this paper:

—We present ERPCNet for zero-shot learning. The network learns a robust and powerful global-local representation in a weakly-supervised manner. It can effectively extract localities that complement the global representation based on semantic relevance and locality relationships. The partial convolution module can mitigate the high training cost of RL and allow training to converge faster.

—We design an entropy-guided reward function and use an entropy ratio to reflect the informativeness of localities. We harness the ratio as expert knowledge to guide the training of the reinforcement module, which can significantly boost and improve the model's efficacy and performance.

—We carry out extensive experiments on four benchmark datasets in both ZSL and Generalized Zero-shot Learning (GZSL) settings to demonstrate the improvement of our proposed model over the state-of-the-art. We further analyze and shed light on the effectiveness, efficiency, and explainability of our model.

II. Methodology

We start by introducing the problem definition of ZSL/GZSL and notations used in the paper. Let $\mathcal{S} = \{(x, y, a)|x \in X^S, y \in Y^S, a \in A^S\}$ be the training data from seen classes (i.e., classes with labeled samples), where $x \in X^S$ denotes the data instance (i.e., an image), $y \in Y^S$ denotes the class label of x, and $a \in A^S$ represents an attribute (or other semantic side information) of y. Similarly, we define test data from unseen classes as $\mathcal{U} = \{(x, y, a)|x \in X^U, y \in Y^U, a \in A^U\}$. Given an image x from an unseen class and a set of attributes of unseen classes A^U, ZSL aims to predict the class label $y \in Y^U$ of the image, where seen and unseen classes are disjoint, i.e., $Y^S \cap Y^U = \emptyset$. GZSL is more challenging, aiming to predict images from both seen and unseen classes, i.e., $y \in Y^U \cup Y^S$.

A. Overview

The procedure of ERPCNet is described in Figure 2. ERPCNet consists of the global subnet, the local subnet, and

the joint supervision subnet. The global subnet extracts global information and provides inspiration for determining the initial patch location. The local subnet adopts the entropy-guided policy network π as the sampler to select discriminative parts and then conducts partial convolution for locality extraction. The joint supervision subnet, composed of two branches, takes the global/local visual and semantic embeddings as the input to conduct joint supervision for better optimization.

The global subnet consists of the global extractor f_G and the corresponding predictor f_{cg}. Let e_g^i be the corresponding output of the i-th layer of f_G. The global extractor f_G takes raw images as input and plays two important roles in the network: 1) extracting the global representation $\bar{e}_g^n$ of the original images and 2) providing the preliminary information e_g^{n-1} for the local subnet, where f_{cg} optimizes $\bar{e}_g^n$ and e_g^{n-1} to carry the attribute information.

Given $\bar{e}_g^n$ and e_g^{n-1} produced by the global subnet, the local subnet employs the partial convolution module f_P, the locality extractor C_l and the predictor f_{cl} to progressively learn localities to complement our global representation. f_P provides localities by an entropy-guided sampler π (for region selection) and a convolution kernel (for partial convolution). C_l further extracts high-level locality representation, and f_{cl} ensures attribute-richness of the locality.

With the global representation and extracted localities from global/local subnets, the joint supervision subnet optimizes the extracted embeddings. It consists of a fusion module f_{cj} and a normalized max pool for joint attribute regularization and highlighted attribute regularization, respectively.

B. Global subnet

The global subnet aims to extract discriminative global features for zero-shot recognition and provide adequate preliminary information for the local subnet. Given an input image x, the global extractor $f_G = \{C_g^1, C_g^2, ..., C_g^n\}$ (a CNN backbone) embeds the input to a visual feature map $e_g^n \in \mathbb{R}^{H\times W\times CH}$: $e_g^n = C_g^n(...(C_g^1(x)))$, where H, W and CH denote height, width and channel, respectively; n denotes the n-th layer in the global extractor; C denotes a convolutional block.

The extractor is followed by global average pooling to learn a visual embedding $\bar{e}_g^n$, which is further projected into the semantic space by the predictor f_{cg}. f_{cg} optimizes the global subnet using the loss L_g to promote the compatibility between the learned embedding and the corresponding attribute:

$$L_g = CE(\bar{a}_g, y) = -\log \frac{\exp(f_{cg}(\bar{e}_g^n)^T \phi(y))}{\sum_{\hat{y}\in Y^S} \exp(f_{cg}(\bar{e}_g^n)^T \phi(\hat{y}))} \quad (1)$$

where $\bar{e}_g^n = AdaptiveAvgPool(e_g^n)$; y denotes the label for x; $\phi(y)$ denotes the attribute vector of y; CE denotes CrossEntropy.

C. Local subnet

The local subnet aims to progressively discover the localities $\bar{e}_{l,t}$ to complement the global embedding. We propose the entropy-guided reinforced partial convolution module f_P, the local extractor C_l, and the local predictor f_{cl} to select regions and extract localities iteratively based on semantic relevance and region correlation.

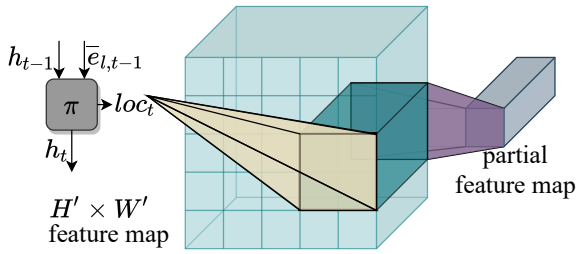


Fig. 3. Partial convolution f_P.

Entropy-guided reinforced partial convolution. Traditional convolution starts with a kernel that slides over the input data. The kernel repeatedly conducts element-wise multiplication and aggregates the results on all locations that it slides over. Unlike traditional convolution, to explore and strengthen localities, our main idea is to conduct partial convolution, i.e., we carry out the multiplication and summation procedure only on selected regions that are critical for classification, as shown in Figure 3.

Suppose that $H' \times W'$ is the input size of the partial convolution, and $k \times k$, q, and p are the kernel size, stride, and padding, respectively. Partial region location search can be transferred to a grid search problem with the grid size of $(\frac{H'+2p-k}{q} + 1) \times (\frac{W'+2p-k}{q} + 1)$. The search is fulfilled by a recurrent network π that aggregates all the previous information. Since partial convolution is non-differentiable, we consider π as an agent and optimize it using an RL method called Proximal Policy Optimization (PPO) [23]. When the entropy-guided partial convolution module f_P takes the global feature map (i.e., the intermediate output of f_G) from more forward layers as the input, on the one hand, the computational cost to explore localities increases since there are more search locations with larger H' and W', and more subsequent convolution operations are needed to encode the selected locality to the same dimension of the global embedding e_g^n. On the other hand, in more forward layers, there may exist more useful information ignored in the global embedding. Therefore, as a trade-off between performance and computational cost, we assign f_P after the $(n-1)$-th convolutional block C_g^{n-1} of f_G. This way, the action space of π is drastically reduced, and the regions become more information-intensive. It also becomes easier for π to make decisions and achieve higher rewards. All the above-mentioned factors can help mitigate the sample efficiency problem of RL.

To better utilize global and local information, we design the state s_t for π to cover two situations during progressive selection:

$$s_t = \begin{cases} < \bar{e}_g^n, \varnothing > & t = 1 \\ < \bar{e}_{l,t-1}, h_{t-1} > & t > 1 \end{cases} \quad (2)$$

where s_t denotes the state for the t-th step; $\varnothing$ denotes the empty set; h_{t-1} denotes the hidden state from the previous selection in the recurrent network. In the first step, we highlight the most helpful region for global embedding. In the following steps, we keep previous selections as the hidden

information and find the best locality for the current extracted representation.

Given the current state s_t, the sampler π chooses a locating action $loc_t \sim \pi(loc_t|s_t)$. $loc_t = \{i, j\}$ is a coordinate where $i \in [1, \frac{H'+2p-k}{q} + 1]$ and $j \in [1, \frac{W'+2p-k}{q} + 1]$. Then, we can obtain the region locality $e^p_{l,t}$ as: $e^p_{l,t} = Conv(Crop(\pi(s_t), e^{n-1}_g))$, where $Crop$ is a Region of Interest (RoI) pool; $Conv$ is the convolution kernel for f_P. We use $Crop$ to align the output size during selection.

We repeat the procedure of selecting and extracting localities until the reward of π exceeds a pre-defined threshold σ. Regions that have been visited will not be chosen again. The definition of the reward and the details of π will be discussed in Section II-E. At this stage, $e^p_{l,t}$ is rough and insufficient for predicting attribute vectors, so we apply a locality extractor C_l to further distill: $\bar{e}_{l,t} = C_l(e^p_{l,t})$.

To optimize the convolution kernels in f_P and C_l, we apply a local predictor f_{cl} to help train the kernels to effectively extract attribute-related localities by a locality loss L_l:

$$\begin{aligned} L_l &= \frac{1}{|step|}\sum_t L_{l,t}(\bar{a}_{l,t}, y) \\ &= -\frac{1}{|step|}\sum_t \log \frac{\exp(f_{cl}(\bar{e}_{l,t})^T\phi(y))}{\sum_{\hat{y}\in Y^S}\exp(f_{cl}(\bar{e}_{l,t})^T\phi(\hat{y}))} \end{aligned} \tag{3}$$

where $|step|$ denotes the selection number. It may be insufficient to use the same ground-truth attributes to optimize the local subnet since we aim to capture diverse localities across steps. Therefore, we apply a maximum prediction loss L_m to maximize locality diversity in Section II-D.

D. Joint supervision subnet

We conduct joint supervision over the learned global and local embeddings. Joint supervision consists of two losses: a joint prediction loss L_{jnt} and a maximum prediction loss L_m. Both are evaluated by *CrossEntropy*:

$$L_{jnt} == -\log \frac{\exp(f_{cj}(<\bar{e}^n_g, \bar{e}_{l,1}, ..., \bar{e}_{l,t}>)^T\phi(y))}{\sum_{\hat{y}\in Y^S}\exp(f_{cj}(<\bar{e}^n_g, \bar{e}_{l,1}, ..., \bar{e}_{l,t}>)^T\phi(\hat{y}))} \tag{4}$$

$$L_m == -\log \frac{\exp(\max_i(<\bar{a}^n_g, \bar{a}_{l,1}, ..., \bar{a}_{l,t}>_i)^T\phi(y))}{\sum_{\hat{y}\in Y^S}\exp(\max_i(<\bar{a}^n_g, \bar{a}_{l,1}, ..., \bar{a}_{l,t}>_i)^T\phi(\hat{y}))} \tag{5}$$

where f_{cj} is a fusion module to predict joint attributes $\bar{a}_j$ based on global and local visual embeddings of all steps; $\bar{a}_m$ is a vector composed of the maximum value in each dimension of the learned global and local attributes.

Global-local cooperation. We concatenate the global embedding with the corresponding localities to predict the attribute vectors through f_{cj}. Therefore, f_{cj} can be optimized by the loss L_{jnt} to help the learned global and local embeddings collaborate better. Note that we use zero-padding to align the input of f_{cj} since the lengths of action sequences differ across images.

Locality diversity. Our network aims to enable the local subnet to capture diverse localities. Therefore, representations from different steps should emphasize different parts of the attribute vectors. The loss L_m is designed to optimize the combinations of the most significant parts from global and local attribute embeddings. L_m along with the locality loss L_l jointly improve the locality diversity and discrimination.

E. Entropy-guided policy network

The entropy-guided policy network π is based on the global-local structure and joint feature learning. We introduce information entropy as expert knowledge to help optimize the policy network. A common obstacle of RL training is the sparse-reward problem, which occurs when the RL agent does not observe enough reward signals to reinforce its actions and then hinders the learning. Information entropy is a common tool to measure information quantity and can be used to guide the module towards informative regions that are more likely to contain useful localities, which, intuitively, can help alleviate the sparse-reward problem of RL.

Given an arbitrary instance (x, y, a), we obtain the corresponding locality sequence as $\{\bar{e}^n_g, \bar{e}_{l,1}, ..., \bar{e}_{l,t}\}$. During selection, we conduct the joint prediction for each step: $\bar{a}_{j,t} = f_{cj} < \bar{e}^n_g, \bar{e}_{l,1}, ..., \bar{e}_{l,t} >$. Then, we use the union prediction probability of the ground-truth label as the reward:

$$r_t = \beta(\frac{exp(\bar{a}^T_{j,t}\phi(y))}{\sum_{\hat{y}\in Y^S}\exp(\bar{a}^T_{j,t}\phi(\hat{y}))} + \frac{exp(\bar{a}^T_g\phi(y))}{\sum_{\hat{y}\in Y^S}\exp(\bar{a}^T_g\phi(\hat{y}))}) \tag{6}$$

where β is the entropy weight of instances. The weight β is calculated as follows:

$$\beta = \frac{Entropy(Crop(loc_t, e^{n-1}_g))}{Entropy(e^{n-1}_g)} \tag{7}$$

$$Entropy(e) = -\sum_i\sum_j\sum_k p(e_{i,j,k})\log p(e_{i,j,k}) \tag{8}$$

where i, j, k denote the location coordinates; loc_t denotes the action for step t; $Entropy$ calculates the information entropy of the given region. We assess the entropy ratio of the selected region to the whole and use this ratio to represent the relative information richness. The entropy ratio can scale the prediction confidence to boost the policy network optimization.

Finally, we can optimize the following loss function according to the work of Schulman et al. [23]: $\max_\pi \mathbb{E}[\sum_t \gamma^{t-1} r_t]$, where γ denotes a discount parameter. The detailed optimization procedure is given in *Appendix A*.

F. Training and Inference

We train our model in an end-to-end manner. To prevent overfitting, we set a maximum step number T and halt the selection once the reward exceeds the threshold $r_t \geq \sigma$ $(1 \leqslant t \leqslant T)$ or after T steps.

Training We use a two-stage strategy to maximize the prediction capability with the fewest locality proposals. At **stage I**, we train the model to predict correctly for an arbitrary sequence of local regions. Instead of using π, we randomly select local regions at each step without early-stopping. Then, we optimize the rest of the model by minimizing the overall loss: $L_{erpc} = L_g + L_l + L_{jnt} + L_m$. At **stage II**, we fix the parameters of modules trained in Stage I and use π to

TABLE I
STATISTICS OF EXPERIMENTAL DATASETS

Datasets	Attribute dim	Image num	Seen/Unseen classes
SUN	102	14340	645/72
CUB	1024	11788	150/50
aPY	64	15339	20/12
AWA2	85	37322	40/10

select locations (with early-stopping). Then, we apply PPO to optimize π to pick the most discriminative localities.

Inference Once our model is trained, we use the union of the global and local prediction for zero-shot learning inference: $\bar{a} = \bar{a}_{j,t} + \bar{a}_g$. For ZSL, given an image x, the model extracts global information and then performs locality search iteratively until the termination condition. During inference, the model considers the predicted label as ground-truth to calculate the reward. Then, we take the class with the highest compatibility as the final prediction: $y^U = \arg\max \bar{a}^T_{\hat{y}\in Y^U} a_{\hat{y}\in Y^U}$. For GZSL, since both seen and unseen classes may occur during testing, there exists a strong bias toward seen classes. To help eliminate the bias, we adopt Calibrated Stacking (CS) [24] to decrease the confidence of seen classes by a constant. The final prediction would be: $y^{U\cup S} = \arg\max < \bar{a}^T_{\hat{y}\in Y^U} a_{\hat{y}\in Y^U}, \bar{a}^T_{\hat{y}\in Y^S} a_{\hat{y}\in Y^S} - \delta >$, where δ is a pre-defined parameter.

III. EXPERIMENTS

We conduct experiments on four benchmark datasets for both ZSL and GZSL: SUN [25], CUB [26], aPY [27], and AwA2 [28]. SUN and CUB are fine-grained datasets, containing 14,340 images from 717 scene classes with 102 attributes and 11,788 images from 200 bird species with 312 attributes, respectively; aPY contains 15,339 images from 32 classes with 64 attributes, where images are from two distinct main types (buildings and animals); AwA2 is a large coarse-grained dataset comprising 37,322 images from 50 diverse animals with only 85 attributes. We adopt Proposed Split (PS) [28], which is commonly used to avoid unseen data leak, to divide datasets into seen/unseen classes. Table I shows the statistics of the datasets and splits.

We adopt Resnet101 [29] pretrained on ImageNet [30] as the backbone (i.e., the global extractor f_G) and divide f_G into blocks $\{C^1_g, C^2_g, \ldots, C^n_g\}$ following [29]. C_l shares the same structure and initial parameters with C^n_g but will have different parameters after optimization. At Stage I, we use SGD [31] with image size of 224×224, momentum of 0.9, weight decay of 10^{-5}, and a learning rate of 10^{-3}. The learning rate decays by 0.1 every 30 epochs. At Stage II, we use Adam [32] to optimize π with a learning rate of 3×10^{-4} and γ of 0.99. The maximum step T is set to be 10 for AwA2, and 6 for other datasets. More parameters and network architecture are given in *Appendix B*.

A. Comparisons with Baselines and Ablation Study

ZSL: We compare our method with two groups of state-of-the-art methods: non-end-to-end methods (including embedding methods and generative methods) and end-to-end methods. We evaluate the methods by average per-class Top-1 (T1) accuracy to mitigate the influence of class imbalance. Results are shown in Table III. For competitors, we use the accuracy reported in the original papers. Since APN [6] additionally uses group side information (besides class labels), we list the results of its without-group version to make a fair comparison.

TABLE II
OVERALL COMPARISON IN ZSL. THE PERFORMANCE IS EVALUATED BY AVERAGE PER-CLASS TOP-1 ACCURACY (%). NON-END-TO-END AND END-TO-END METHODS ARE LISTED AT THE TOP AND BOTTOM, RESPECTIVELY. THE BEST RESULTS ARE MARKED IN BOLD.

Method	SUN	CUB	aPY	AwA2
Non-End-to-End				
SP-AEN [33]	59.2	55.4	24.1	58.5
RelationNet [34]	-	55.6	-	64.2
PSR [35]	61.4	56.0	38.4	63.8
PREN [36]	60.1	61.4	-	66.6
SGV-18 [37]	59.0	67.2	-	67.5
Generation Methods				
cycle-CLSWGAN [38]	60.0	58.4	-	67.3
f-CLSWGAN [39]	58.6	57.7	-	68.2
TVN [40]	59.3	54.9	40.9	68.8
SE-GAN [41]	61.8	60.8	-	68.8
Zero-VAE-GAN [42]	58.5	51.1	34.9	66.2
End-to-End				
QFSL [43]	56.2	58.8	-	63.5
SGMA [5]	-	71.0	-	68.8
LFGAA [44]	61.5	67.6	-	68.1
VisEn [45]	-	58.6	-	65.7
AREN [15]	60.6	71.5	39.2	67.9
SELAR-GMP [46]	58.3	65.0	-	57.0
APN [6]	60.9	71.5	-	68.4
GlobalNet (ours)	61.3	68.1	39.4	66.9
PCNet (ours, random)	62.8	71.2	41.8	69.6
RPCNet (ours, reinforced)	**63.3**	72.0	**43.5**	71.6
ERPCNet (ours, entropy-guided)	63.1	**72.5**	**43.5**	**71.8**

Table II shows that our method consistently outperforms other models (and especially other end-to-end methods) by a large margin. In particular, ERPCNet outperforms the second-best method by 1.5%, 1%, 2.6%, and 3% on SUN, CUB, aPY and AWA2, respectively. The performance gain on SUN and CUB (which contains fewer images for each class) is not as significant as on AwA2 and aPY.

Ablation study: We also compare with GlobalNet (classification using only global subnet), PCNet (randomly selecting locality), and RPCNet (ERPCNet without entropy guidance) as ablations in ZSL. Our proposed entropy-guided reinforced partial convolution is effective on the four benchmark datasets, demonstrated by the reinforced partial convolution module improving T1 by up to 2.0%, 4.4%, 4.1% and 4.9% on SUN, CUB, aPY, and AwA2, respectively, when compared with GlobalNet (shown in Table II). The improvement derives from three aspects: 1) the incorporation of local information (proved by the superiority of PCNet over GlobalNet), 2) the use of RL to progressively select localities (confirmed by the advantage of RPCNet over PCNet), and 3) the guidance of entropy (demonstrated by comparing ERPCNet with RPCNet). We also conduct an ablation study on L_m to prove the effectiveness of our proposed loss function. We do the ablation study on PCNet to eliminate the influence of the RL component. The results on CUB dataset are 71.2 % with L_m, and 70.4% without L_m.

TABLE III
OVERLL COMPARISON IN GZSL. THE PERFORMANCE IS EVALUATED BY AVERAGE PER-CLASS TOP-1 ACCURACY (%) ON SEEN CLASSES(S), UNSEEN CLASSES (U), AND THEIR HARMONIC MEAN (H). WE EMBOLDEN THE BEST RESULT ON EACH DATASET.

Method	SUN			CUB			aPY			AWA2		
	U	S	H	U	S	H	U	S	H	U	S	H
Non End-to-End												
SP-AEN [33]	24.9	**38.6**	30.3	34.7	70.6	46.6	13.7	63.4	22.6	23.0	90.9	37.1
RelationNet [34]	-	-	-	38.1	61.1	47.0	-	-	-	30.0	**93.4**	45.3
PSR [35]	20.8	37.2	26.7	24.6	54.3	33.9	13.5	51.4	21.4	20.7	73.8	32.3
PREN [36]	35.4	27.2	30.8	35.2	55.8	43.1	-	-	-	32.4	88.6	47.4
Generative Methods												
cycle-CLSWGAN [38]	**47.9**	32.4	38.7	43.8	60.6	50.8	-	-	-	56.0	62.8	59.2
f-CLSWGAN [39]	42.6	36.6	39.4	43.7	57.7	49.7	-	-	-	57.9	61.4	59.6
TVN [40]	22.2	38.3	28.1	26.5	62.3	37.2	16.1	**66.9**	25.9	27.0	67.9	38.6
SE-GAN [41]	44.7	37.0	**40.5**	48.4	57.6	52.6	-	-	-	55.1	61.9	58.3
Zero-VAE-GAN [42]	44.4	30.9	36.5	41.1	48.5	44.4	30.8	37.5	33.8	56.2	71.7	63.0
End-to-End												
QFSL [43]	30.9	18.5	23.1	33.3	48.1	39.4	-	-	-	52.1	72.8	60.7
SGMA [5]	-	-	-	36.7	71.3	48.5	-	-	-	37.6	87.1	52.5
LFGAA [44]	20.8	34.9	26.1	43.4	**79.6**	56.2	-	-	-	50.0	90.3	64.4
AREN [15]	40.3	32.3	35.9	63.2	69.0	66.0	30.0	47.9	36.9	54.7	79.1	64.7
SELAR-GMP [46]	22.8	31.6	26.5	43.5	71.2	54.0	-	-	-	31.6	80.3	45.3
APN [6]	41.9	34.0	37.6	65.3	69.3	67.2	-	-	-	56.5	78.0	65.5
Ours ERPCNet	47.2	31.9	38.1	**67.1**	69.6	**68.4**	**32.7**	49.3	**39.3**	**59.1**	82.0	**68.7**

We also do ablation on SUN dataset and vary the ratio of L_m. The results are 62.8% and 61.5% when the ratios are 1 and 0.5, respectively.

GZSL: Following [28], we evaluate the average per-class accuracy on seen classes (denoted by S), unseen classes (denoted by U), and their harmonic mean (defined as $H = \frac{2US}{U+S}$) in the GZSL setting. Table III shows that our model outperforms all other embedding approaches, especially on the aPY and AwA2 datasets, yielding 2.4% and 3.2% improvements of H, respectively. The results demonstrate that our model can transfer knowledge from seen classes to unseen classes successfully.

Since GZSL needs to predict labels for both seen and unseen classes, there may exist a strong bias towards seen classes during testing. Generative methods can, to some extent, address the problem naturally by synthesizing instances for unseen classes. This explains why generative methods perform better than non-generative methods in GZSL. Interestingly, our model's performance is comparable to or better than generative models, demonstrating our model's generalization ability.

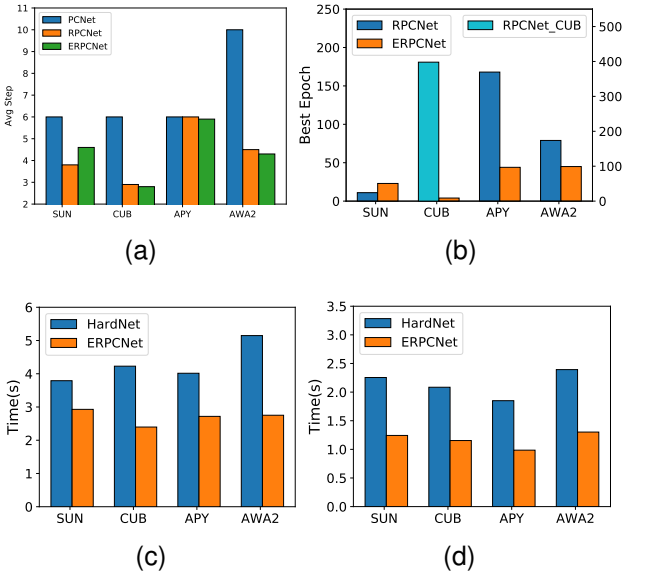


Fig. 4. (a)-(b) Comparison between using and not using entropy-guidance. (a) Average Step. (b) Best epoch. (c)-(d) Efficiency analysis (unit: s). (c) Average train time. (d)Average test time.

B. Efficacy of entropy-guided reinforcement learning

Figures 4(a)-(b) show the average terminating steps and best epochs (at which the model achieves the highest accuracy) of RPCNet and ERPCNet on the four datasets. Both RPCNet and ERPCNet take fewer steps than PCNet (6/10 steps) but achieve higher accuracy, indicating the effectiveness of the reinforced module π. Entropy-guided reinforcement learning can largely decrease the number of epochs required to obtain the best performance on CUB, aPY and AwA2. Specifically, ERPCNet converges in 45 epochs for all datasets, which is fast for RL training. Also, entropy knowledge can slightly reduce the steps during testing. Entropy knowledge does not work well on the SUN dataset. We analyze the value ranges of the entropy weight β and find that β on SUN (on average, 1.09) is slightly smaller than on other datasets (on average, 1.12), which may impair the results.

C. Efficiency of partial convolution

To examine the efficiency of partial convolution, we compare our model against using hard attention [47] to explore localities (denoted by HardNet). Hard attention finds important image patches and extracts localities from the cropped images. We train two feature extractors sharing the same structure with our f_G to learn from the original images and the cropped patches, respectively. We also adopt a PPO agent π' for HardNet optimization. Since the size ($H' \times W'$) of the feature map for partial convolution is 14×14, with the kernel size

being 5×5 and the HardNet input image size being 224×224, we set the patch size in HardNet to 80×80 proportionally. The average training and testing time of a single instance for the optimization of π and π' is shown in Figures 4(c)-(d), and our model consumes around $2/3$ and $1/2$ of the HardNet training/testing time, respectively. The results demonstrate the efficiency of our partial convolution design. Integrating RL with convolution reduces the action space from any location in 224×224 images to 4×4, thus reducing the time cost.

D. Training Convergence Analysis

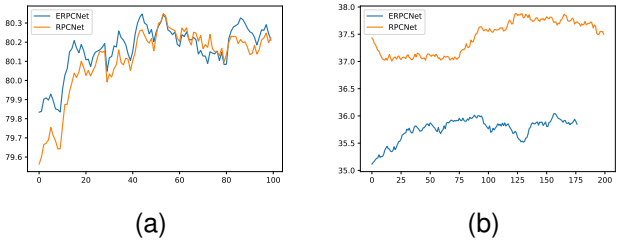


Fig. 5. Acc-epoch curves. (a) CUB. (B) SUN.

To further demonstrate our claim that the proposed entropy guidance can accelerate training convergence and improve performance, we show how the training accuracy changes as more epochs are performed on CUB and SUN in Figure 5. We can find that, for CUB, the training converges around 16 epochs with the entropy guidance compared with 44 epochs without entropy guidance. Besides, the training accuracy with entropy guidance is higher. On the contrary, the entropy slightly impairs the performance on SUN, which is consistent with our observation in Section III-B. This may be due to the lower average entropy of SUN.

E. Hyper-parameters

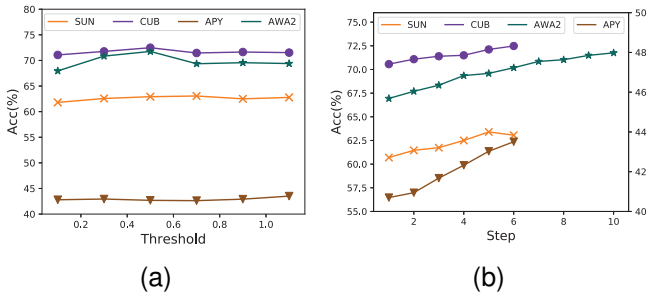


Fig. 6. Hyper-parameter analysis. (a) Threshold analysis. (b) Step-acc curve.

Threshold σ of π: We show the performance of ZSL varying σ from 0.1 to 1.1 with a step of 0.2 in Figure 6(a). The results are stable when σ is over 0.7 and slightly influenced by σ when $\sigma \in [0.1, 0.5]$.

Step-acc curve: We fix the maximum steps T to 6 on three datasets (SUN, CUB, and aPY) and 10 on AwA2. The step-accuracy curves in Figure 6(b) show the accuracy

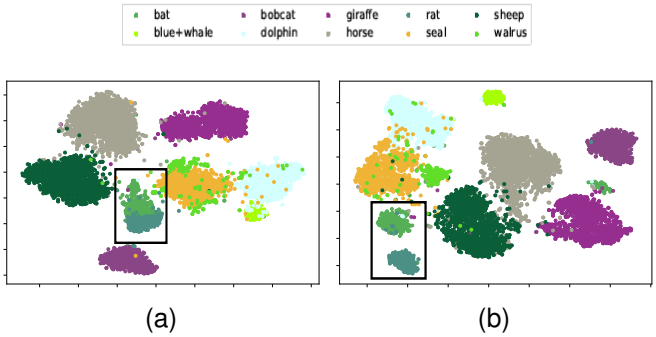


Fig. 7. t-SNE visualization of unseen classes on AwA2. Rat and Bat are circled. (a) Global embedding. (b) Union embedding.

increases as more steps are performed, and the improvement tends to be subtle after five steps or even diminishes on SUN. The results indicate that the local subnet incorporation benefits classification, but introducing excessive locality could be disadvantageous. Analysis of hyper-parameters for RPCNet is provided in *Appendix* C.

F. Visualization

a) Embedding distribution visualization: Figures 7 (a) and (b) visualize the distributions of the global embedding $\bar{e}_g^n$ and the union embedding $\bar{e}_u = \bar{e}_g^n + e_j$ of unseen classes, respectively, on AwA2 by t-SNE visualization [48]. e_j is the intermediate output of f_{cj}. The results show that the global embedding can distinguish most classes but can still be confused on some unseen classes, such as *bat* and *rat* (circled in Figure 7); in contrast, the union embedding, combined with localities, is discriminative enough to distinguish the confused classes.

b) Progressive process visualization: Figures 8 (a) and (b) visualize the distinct instances that are easily predictable with the global representation and ones that can only be correctly classified with progressive localities on CUB and SUN, respectively. For easier understanding, we project the selected locality in the abstract hierarchies loc_t $(1 \leqslant t \leqslant T)$ into the original image-level and use bounding boxes to represent the locations. For the CUB dataset, green violetears can be easily classified with probability of 99.9%, due to their distinctiveness from other species. When classifying other similar bird species, the progressive locality detector can gradually increase the probability of correct predictions by locating the regions of wing, neck, head, and tail to highlight the birds' discriminative characteristics. Different from the CUB dataset, experiments show that the model tends to progressively choose representative regions for diverse objects on the SUN dataset. For example, the proposed ERPCNet first focuses on the wall, then concentrates on decorations and chairs to distinguish indoor and indoor seats. This indicates that our method can progressively pick up the best locality to distinguish similar or diverse objects effectively.

G. Failure Modes of RL Component

Apart from the examples that ERPCNet can successfully find progressive localities in Section III-F0b, we also perform

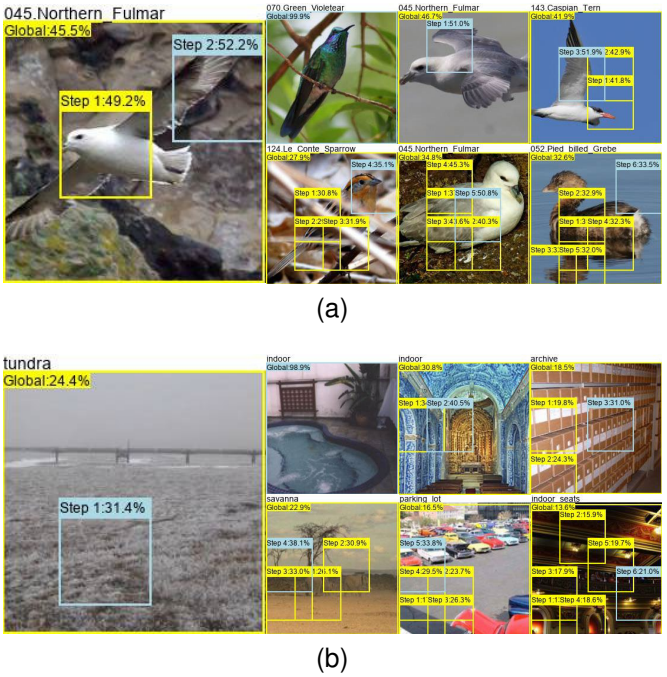


Fig. 8. Visualization of progressive locality selection. The labels above the boxes denote the step index and the prediction confidence after this selection. The box color indicates the prediction correctness at the current step (blue: correct; yellow: wrong). (a) CUB dataset. (b) SUN dataset.

TABLE IV
FAILURE OCCUR STEPS.

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
8	3	13	3	17	13

a detailed investigation to explore failure modes of RL, including the definition, the statistics, and the reasons for failure mode.

We use trained models and do experiments on the CUB dataset in the ZSL setting. There are 2,697 pictures in the test set, with an accuracy rate of 72.5%, i.e., 816 pictures are misclassified. There are two types of misclassifications: 1) The model keeps misclassifying the images during the entire decision-making process of extracting global information and exploring the localities; 2) The model first classifies the images correctly but then misclassifies the images after performing several steps of locality exploration.

We attribute the first type of misclassification to the images being beyond the classification capabilities of our model and the second type to the failure mode of the RL component. Specifically, 759 images belong to the first misclassification, and 57 images belong to the second misclassification, i.e., the failure mode. In most cases, RL components are qualified (57 failures compared with 1,938 images that are within the model capability).

TABLE V
ACCURACY AT DIFFERENT STEPS.

Global	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
48.7	47.5	46.3	45.2	44.2	42.8	37.6

We show the steps of failure occurrence and the number of failures in Table IV. We find that failure may occur after any step of locality exploration, and there is no obvious regularity in the number distribution. We also find that no matter which steps the failure happens in the failure mode, the predicted probabilities of correct labels decrease as more localities are explored. Taking a failure image of *northern fulmar* (a seabird) for example, the failure happens after exploring five localities. We show the trend of the predicted probability (%) of the correct label in Table V. We can find that the probability declines throughout the process.

We observe the 57 images belonging to the failure mode and find that in these pictures, the main objects account for a relatively small area and are often hidden in cluttered environments, e.g., a bird hiding in a dense tree. Considering that the predicted probabilities of the correct labels decline during the decision-making process, we infer that the following results in the occurrence of the failure mode: the initial prediction probability does not reach the threshold of RL, so the model continues to perform locality exploration; then the messy background is incorporated as localities, which introduces noise, making the probability of correct label decrease and finally leading to wrong results.

IV. RELATED WORK

Zero-Shot Learning (ZSL). ZSL aims to classify classes not seen during training [4], [49]–[51]. A typical strategy is to view ZSL as a visual-semantic embedding problem, which reduces the problem to designing an appropriate projection that maps visual [4], [33], [34], [36], [52] and/or semantic features [49], [53], [54] to a latent space, where ZSL measures the compatibility score of the latent representation for classification. For example, Ye et al. [36] design an ensemble network to learn an embedding function from the same extracted features to diverse labels. Zhang et al. [4] further propose a two-branch ensemble network to learn intra-class compactness and inter-class separability, which can provide a cross-class classifier to ease the model bias problem in ZSL. Zhu et al. [50] propose to fuse the prediction of semantic attributes and the object detection, which can directly predict object bounding boxes for both seen and unseen classes. Several recent efforts [38], [40], [42], [55], [56] convert ZSL to traditional supervised classification by exploring generative models to generate samples for unseen classes. For example, Chi et al. [57] propose a dual adversarial distribution network for the generalized embedding generation and reconstruction in the cross-media retrieval problem, which can effectively learn the underlying semantics and information for the classification across heterogeneous distributions of different media types. Yu et al. [58] propose to fuse knowledge distillation in two different strategies (i.e., class augmentation and semantics guidance) to improve the supervision process of the visual classifier.

Besides, some modern works further enhance ZSL by adopting other learning manners. For example, Chen et al. [59] propose to conduct semantic-visual adaptation in a hierarchical manner instead of the conventional one-step adaptation,

which can analyze the heterogeneous nature of the feature representations. Guo et al. [60] propose a one-step ensemble method to avoid information loss in the conventional two-step recognition, which relies on the support vector machine providing pseudo labels for samples from source classes. Yu et al. [61] simulate episodes of zero-shot settings during the training phases and thus progressively optimize the model to be more generalized. Alamri et al. [62] propose an adapted Vision Transformer to split images into sub-parts and capture discriminative attributes. Some other works focus on a more challenging setting, i.e., GZSL. For example, Chen et al. [63] propose a feature refinement network and a self-adaptive margin center loss to ease the cross-dataset bias between the pre-trained dataset (i.e., ImageNet) and GZSL benchmarks. Liu et al. [64] propose to enhance the indirect attribute prediction by a label-activating framework, which can utilize label information to ease the domain shift in the GZSL.

More related to our work, some end-to-end models have been proposed for better image representation [5], [6], [15], [43]–[45]. LFGAA [44] uses instance-based attribute attention to disambiguate semantic characteristics. Xie et al. [15] combine two branches of the multi-attention module to facilitate embedding learning and attribute prediction. However, multi-attention discovers a fixed number of localities independently while neglecting their region relations, thus restricting the attention weights to the global level. In contrast, our ERPCNet can uncover refined local regions progressively while preserving attribute relevance and inherent correlations.

Locality and representation learning. Locality has been extensively investigated for better representation [5]–[8]. Annotation-based methods [10]–[12] leverage extra annotations in the form of ground-truth bounding boxes to extract local information or train local detectors. Weakly-supervised methods [7], [8], [16], [65] can avoid labor-intensive annotations. [14], [15], [66], [67] adopt multi-attention to independently search important regions and treat them equally. Xu et al. [6] propose a prototype network to improve localities by concentrating on semantic groups. Wang et al. [68] use a patch proposal network to focus on discriminative regions and remove spatial redundancy.

Summary. Our model differs from previous studies on three aspects. 1) We first propose a new reinforced framework to find localities in ZSL and jointly learn zero-shot recognition, reinforced locality exploration, and global-local representations in an end-to-end manner. 2) We design entropy as guidance to identify information-rich regions in order to accelerate the training phase and alleviate sparse-reward problems. 3) We propose reinforced partial convolution to discover localities, which converges faster and reduces the computational cost.

V. Conclusion

We propose an Entropy-guided Reinforced Partial Convolutional Network (ERPCNet) to gain better global-local representations in Zero-Shot Learning (ZSL). We perform partial convolution by incorporating a reinforced region sampler with a convolution kernel to dynamically find and learn localities as complements for the global representation. We further introduce entropy knowledge into the reward design to guide the model toward informative regions. We evaluate our model through extensive experiments against state-of-the-art methods in both ZSL and GZSL settings on four benchmark datasets. The results demonstrate the superior performance and robustness of ERPCNet in global-local representation learning. Ablation studies show our model's effectiveness in locality exploration and efficiency in training/testing the reinforced module. In the future, we will extend ERPCNet to handle a broader variety of multi-modality learning problems, e.g., visual question answering or audio-visual speech recognition that incorporates acoustic and linguistic modalities. We will further explore augmenting other convolutional networks with ERPCNet as a plug-and-play component to boost their performance.

Appendix A
Proximal Policy Optimization

Our reinforcement module is implemented by an Actor-critic network, which consists of an actor π and a critic V. The critic V aims to estimate the state value [23]. The detailed module architecture is shown in Section B-A.

During the training process of the reinforcement module, we sample actions following $loc \sim \pi(loc|s_t)$ to optimize the policy network, where s_t denotes the state for the t-th step by maximizing the following rewards:

$$\max_{\pi} \mathbb{E}[\sum_{t} \gamma^{t-1} r_t] \tag{9}$$

where $\gamma = 0.99$ is a pre-defined discounted parameter and r_t denotes the reward. According to the work of Schulman et al. [23], the optimization problem can be addressed by a surrogate objective function using stochastic gradient ascent:

$$L_t^{CPI} = \frac{\pi(loc|s_t)}{\pi_{old}(loc|s_t)} \hat{D}_t \tag{10}$$

where π_{old} and π represent the before and after updated policy network, respectively. $\hat{D}_t$ is the advantages estimated by an Actor-critic network V by:

$$\hat{D}_t = -V(s_t) + \sum_{t \le i \le T} \gamma^{i-t} r_t \tag{11}$$

where T denotes the maximum length of the action sequence. The policy network usually gets trapped in local optimality via some extremely great update steps when directly optimizing L^{CPI}, so we optimize a clipped surrogate objective:

$$L_t^{CLIP} = min\{\frac{\pi(loc|s_t)}{\pi_{old}(loc|s_t)} \hat{D}_t, Clip(\frac{\pi(loc|s_t)}{\pi_{old}(loc|s_t)}) \hat{D}_t\} \tag{12}$$

where $Clip$ is the operation that clips input to $[1-\epsilon, 1+\epsilon]$. We set $\epsilon = 0.2$ in our experiments.

Then, to further promote the exploration of policy and the performance of V, we take the following loss function as the final optimization goal:

$$\max_{\pi, V} \mathbb{E}_{x,t}[L_t^{CLIP} - \alpha_1 MSE(V(s_t, \sum_{t \le i \le T} \gamma^{i-t} r_t) + \alpha_2 S_\pi(s_t)] \tag{13}$$

where $\alpha_1 = 0.5$, $\alpha_2 = 0.01$, MSE is the mean square error loss, and $S_\pi(s_t)$ denotes the entropy bonus [23], [69].

APPENDIX B
MORE IMPLEMENTATION DETAILS

All algorithms are implemented in Pytorch 1.7.0 and compiled with GCC 7.3.0. The system is Linux 3.10.0, and the GPU type is GP102 TITANX. The cuda version is 10.0.130. The stop threshold $sigma$ of π is set to be 0.7, 0.5, 1.1 and 0.5 for SUN, CUB, aPY, and AwA2, respectively. For GZSL, the factor δ of CS is set to 0.2, 0.8, 0.5, and 0.5 for SUN, CUB, aPY, and AwA2, respectively.

A. Architecture Implementation

Our model relies on the convolution layer and fully connected layer. $FC(n)$ represent a fully-connected layer with output size n. We use the same network structure for all four benchmark datasets yet different parameters for dropout layers. In the following, we introduce the detailed network architecture of global subnet, local subnet, and other prediction layers, respectively.

First, we introduce the common setting for the layers. We use adaptive average pool (AdaptiveAvgPool) with output size 1×1, rectified linear activation function (ReLU) with default parameter and sigmoid activation function with default parameter for each module. In the global subnet, which is composed of f_G followed by an adaptive average pool, the input is the cropped image with the size of $3 \times 224 \times 224$. We use the pre-trained ResNet-101 [29] for f_G and set the output size to 1×1 for AdaptiveAvgPool, where the output size of the global subnet is $N \times 2048$, and N denotes batch size.

The local subnet consists of a partial convolution module f_P and a convolution layer module C_l. To keep the same structure as the global subnet, we use the last block of ResNet-101 as C_l. As for the reinforced partial convolution module, f_P contains a policy network π and a convolution kernel (size 5×5 and stride step 3). π shares the same state encoder structure f_E with the state value estimator V. The state structure f_E is a recurrent network as follows:

$$f_E =< FC(1024) - ReLU - FC(256) - ReLU - GRU) > \tag{14}$$

where GRU denotes a gated recurrent unit with input size 256 and hidden size 256. Then, we design π and V by:

$$\begin{aligned} \pi &=< f_E - FC(|Action|) - Sigmoid > \\ V &=< f_E - FC(1) > \end{aligned} \tag{15}$$

where $|Action|$ denotes the action dim.

In respect of the prediction layers, $f_{cj} =< FC(2048) - Dropout - FC(|A|) - Dropout >$, f_{cg} and f_{cl} share the same structure as $< FC(|A|) - Dropout >$, where $|A|$ denotes the attribute vector dim and $Dropout$ is the dropout layer. The dropout layer parameters for CUB, aPY, AwA2 and SUN are 0.5, 0.5, 0 and 0 respectively.

APPENDIX C
MORE EXPERIMENTS

Threshold σ of π: For RPCNet, we show the results of average per-class accuracy of ZSL when varying σ from 0.1 to 1.1 with a step of 0.2. The results in Figure 9(a) are stable on σ except for AwA2 dataset.

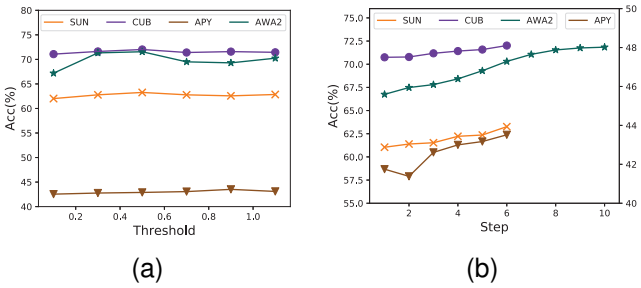


Fig. 9. (a) Threshold hyper-parameter analysis and (b) step-acc curve on RPCNet.

Step-acc curve: We fix the maximum step T of RPCNet to be 6 for SUN, CUB, and aPY, whereas 10 for AwA2. The step-acc curves in Figure 9(b) illustrate the changing tendency of accuracy when the locality is progressively explored. Overall, the accuracy increases as more steps are performed, and the improvements tend to be subtle after exploring sufficient localities.

REFERENCES

[1] C. Yan, Q. Zheng, X. Chang, M. Luo, C. Yeh, and A. G. Hauptmann, "Semantics-preserving graph propagation for zero-shot object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 8163–8176, 2020.

[2] Z. Li, L. Yao, X. Chang, K. Zhan, J. Sun, and H. Zhang, "Zero-shot event detection via event-adaptive concept relevance mining," *Pattern Recognit.*, vol. 88, pp. 595–603, 2019.

[3] H. Cui, L. Zhu, J. Li, Y. Yang, and L. Nie, "Scalable deep hashing for large-scale social image retrieval," *IEEE Trans. Image Process.*, vol. 29, pp. 1271–1284, 2020.

[4] L. Zhang, P. Wang, L. Liu, C. Shen, W. Wei, Y. Zhang, and A. Van Den Hengel, "Towards effective deep embedding for zero-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2843–2852, 2020.

[5] Y. Zhu, J. Xie, Z. Tang, X. Peng, and A. Elgammal, "Semantic-guided multi-attention localization for zero-shot learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 943–14 953.

[6] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata, "Attribute prototype network for zero-shot learning," in *34th Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2020.

[7] T. Sylvain, L. Petrini, and D. Hjelm, "Locality and compositionality in zero-shot learning," in *International Conference on Learning Representations*, 2019.

[8] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *International Conference on Learning Representations*, 2018.

[9] D. Yuan, X. Chang, P. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 976–985, 2021.

[10] Z. Ji, Y. Fu, J. Guo, Y. Pang, Z. M. Zhang *et al.*, "Stacked semantics-guided attention model for fine-grained zero-shot learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 5995–6004.

[11] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele, "Multi-cue zero-shot learning with strong supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 59–68.

[12] M. Elhoseiny, Y. Zhu, H. Zhang, and A. Elgammal, "Link the head to the" beak": Zero shot learning from noisy text description at part precision," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 6288–6297.

[13] L. Yang, C. Kong, X. Chang, S. Zhao, Y. Cao, and S. Zhang, "Correlation filters with adaptive convolution response fusion for object tracking," *Knowl. Based Syst.*, vol. 228, p. 107314, 2021. [Online]. Available: https://doi.org/10.1016/j.knosys.2021.107314

[14] D. Huynh and E. Elhamifar, "Fine-grained generalized zero-shot learning via dense attribute-based attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4483–4493.

[15] G.-S. Xie, L. Liu, X. Jin, F. Zhu, Z. Zhang, J. Qin, Y. Yao, and L. Shao, "Attentive region embedding network for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9384–9393.

[16] D. Huynh and E. Elhamifar, "Compositional zero-shot learning via fine-grained dense feature composition," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[17] Y. Yang, Y. Zhuang, and Y. Pan, "Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies," *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 12, pp. 1551–1558, 2021.

[18] Y. Xiao, W. Lei, L. Lu, X. Chang, X. Zheng, and X. Chen, "CS-GAN: cross-structure generative adversarial networks for chinese calligraphy translation," *Knowl. Based Syst.*, vol. 229, p. 107334, 2021.

[19] C. Yan, X. Chang, M. Luo, Q. Zheng, X. Zhang, Z. Li, and F. Nie, "Self-weighted robust LDA for multiclass classification with edge classes," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 1, pp. 4:1–4:19, 2021.

[20] X. Lu, L. Liu, L. Nie, X. Chang, and H. Zhang, "Semantic-driven interpretable deep multi-modal hashing for large-scale multimedia retrieval," *IEEE Trans. Multim.*, vol. 23, pp. 4541–4554, 2021.

[21] R. Zhou, X. Chang, L. Shi, Y. Shen, Y. Yang, and F. Nie, "Person reidentification via multi-feature fusion with adaptive graph learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 5, pp. 1592–1601, 2020.

[22] P. Sermanet, A. Frome, and E. Real, "Attention for fine-grained categorization," *arXiv preprint arXiv:1412.7054*, 2014.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[24] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *European conference on computer vision*. Springer, 2016, pp. 52–68.

[25] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2751–2758.

[26] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," 2010.

[27] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1778–1785.

[28] Y. Xian, C. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251–2265, 2019.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[31] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang, "Zero-shot visual recognition using semantics-preserving adversarial embedding networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1043–1052.

[34] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[35] Y. Annadani and S. Biswas, "Preserving semantic relations for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7603–7612.

[36] M. Ye and Y. Guo, "Progressive ensemble networks for zero-shot recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 728–11 736.

[37] Y. Hu, G. Wen, A. Chapman, P. Yang, M. Luo, Y. Xu, D. Dai, and W. Hall, "Semantic graph-enhanced visual network for zero-shot learning," *arXiv preprint arXiv:2006.04648*, 2020.

[38] R. Felix, V. B. Kumar, I. Reid, and G. Carneiro, "Multi-modal cycle-consistent generalized zero-shot learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 21–37.

[39] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5542–5551.

[40] H. Zhang, Y. Long, Y. Guan, and L. Shao, "Triple verification network for generalized zero-shot learning," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 506–517, 2019.

[41] A. Pambala, T. Dutta, and S. Biswas, "Generative model with semantic embedding and integrated classifier for generalized zero-shot learning," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1237–1246.

[42] R. Gao, X. Hou, J. Qin, J. Chen, L. Liu, F. Zhu, Z. Zhang, and L. Shao, "Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 3665–3680, 2020.

[43] J. Song, C. Shen, Y. Yang, Y. Liu, and M. Song, "Transductive unbiased embedding for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1024–1033.

[44] Y. Liu, J. Guo, D. Cai, and X. He, "Attribute attention for semantic disambiguation in zero-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[45] M. Bustreo, J. Cavazza, and V. Murino, "Enhancing visual embeddings through weakly supervised captioning for zero-shot learning," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[46] S. Yang, K. Wang, L. Herranz, and J. van de Weijer, "Simple and effective localized attribute representations for zero-shot learning," *arXiv*, pp. arXiv–2006, 2020.

[47] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.

[48] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[49] J. Shen, Z. Xiao, X. Zhen, and L. Zhang, "Spherical zero-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[50] P. Zhu, H. Wang, and V. Saligrama, "Zero shot detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 4, pp. 998–1010, 2019.

[51] Y. Tian, Y. Kong, Q. Ruan, G. An, and Y. Fu, "Aligned dynamic-preserving embedding for zero-shot action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1597–1612, 2019.

[52] Z. Liu, Y. Li, L. Yao, X. Wang, and G. Long, "Task aligned generative meta-learning for zero-shot learning," in *Procedings of The Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021.

[53] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2021–2030.

[54] Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto, "Ridge regression, hubness, and zero-shot learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2015, pp. 135–151.

[55] Z. Li, X. Chang, L. Yao, S. Pan, G. Zongyuan, and H. Zhang, "Grounding visual concepts for zero-shot event detection and event captioning," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 297–305.

[56] Y. Li, Z. Liu, L. Yao, and X. Chang, "Attribute-modulated generative meta learning for zero-shot classification," *IEEE Transactions on Multimedia*, 2021.

[57] J. Chi and Y. Peng, "Zero-shot cross-media embedding learning with dual adversarial distribution network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 4, pp. 1173–1187, 2019.

[58] C. Yan, X. Chang, M. Luo, H. Liu, X. Zhang, and Q. Zheng, "Semantics-guided contrastive network for zero-shot object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[59] S. Chen, G. Xie, Y. Liu, Q. Peng, B. Sun, H. Li, X. You, and L. Shao, "Hsva: Hierarchical semantic-visual adaptation for zero-shot learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

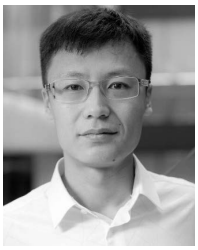
[60] Y. Guo, G. Ding, J. Han, and Y. Gao, "Zero-shot learning with transferred samples," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3277–3290, 2017.

[61] Y. Yu, Z. Ji, J. Han, and Z. Zhang, "Episode-based prototype generating network for zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 035–14 044.
[62] F. Alamri and A. Dutta, "Multi-head self-attention via vision transformer for zero-shot learning," *arXiv preprint arXiv:2108.00045*, 2021.
[63] S. Chen, W. Wang, B. Xia, Q. Peng, X. You, F. Zheng, and L. Shao, "Free: Feature refinement for generalized zero-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 122–131.
[64] Y. Liu, X. Gao, Q. Gao, J. Han, and L. Shao, "Label-activating framework for zero-shot learning," *Neural Networks*, vol. 121, pp. 1–9, 2020.
[65] Z. Liu, L. Yao, L. Bai, X. Wang, and C. Wang, "Spectrum-guided adversarial disparity learning," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 114–124.
[66] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang, "Multiple granularity descriptors for fine-grained categorization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2399–2406.
[67] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian, "Picking deep filter responses for fine-grained image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1134–1142.
[68] Y. Wang, K. Lv, R. Huang, S. Song, L. Yang, and G. Huang, "Glance and focus: a dynamic approach to reducing spatial redundancy in image classification," *arXiv preprint arXiv:2010.05300*, 2020.
[69] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

Yun Li currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney, NSW, Australia. Her current research interests include zero-shot learning, attentive network, machine learning and their applications to computer vision, and genetic engineering.

Zhe Liu has achieved his Ph.D. degree with the School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney, NSW, Australia. His current research interests include zero-shot learning, generative network, data mining and their applications to computer vision, and human activity recognition.

Lina Yao is currently a Scientia associate professor with the School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney, NSW, Australia. Her current research interests include data mining and machine learning with applications to Internet of Things, information filtering and recommending, human activity recognition, and brain-computer interface.



Xianzhi Wang is currently a Lecturer with the School of Computer Science, University Technology of Sydney, Ultimo, NSW, Australia. His current research interests include data fusion, Internet of Things, and recommender systems.

Julian McAuley is currently a professor with the Computer Science Department, University of California San Diego (UCSD), La Jolla, California, USA. His current research interests include recommender systems, web mining, and personalization.

Xiaojun Chang is a Professor at Faculty of Engineering and Information Technology, University of Technology Sydney. He was an ARC Discovery Early Career Researcher Award (DECRA) Fellow between 2019-2021. After graduation, he was worked as a Postdoc Research Associate in School of Computer Science, Carnegie Mellon University, a Senior Lecturer in Faculty of Information Technology, Monash University, and an Associate Professor in School of Computing Technologies, RMIT University. He mainly worked on exploring multiple signals for automatic content analysis in unconstrained or surveillance videos and has achieved top performance in various international competitions. He received his Ph.D. degree from University of Technology Sydney. His research focus in this period was mainly on developing machine learning algorithms and applying them to multimedia analysis and computer vision.