The Lottery Ticket Hypothesis for Self-attention in Convolutional Neural Network♣

Zhongzhan Huang, Senwei Liang*, Mingfu Liang, Wei He, Haizhao Yang†, and Liang Lin†

***Abstract*—Recently many plug-and-play self-attention modules (SAMs) are proposed to enhance the model generalization by exploiting the internal information of deep convolutional neural networks (CNNs). In general, previous works ignore where to plug in the SAMs since they connect the SAMs individually with each block of the entire CNN backbone for granted, leading to incremental computational cost and the number of parameters with the growth of network depth. However, we empirically find and verify some counterintuitive phenomena that: (a) Connecting the SAMs to all the blocks may not always bring the largest performance boost, and connecting to partial blocks would be even better; (b) Adding the SAMs to a CNN may not always bring a performance boost, and instead it may even harm the performance of the original CNN backbone.**

Therefore, we articulate and demonstrate the Lottery Ticket Hypothesis for Self-attention Networks: a full self-attention network contains a subnetwork with sparse self-attention connections that can (1) accelerate inference, (2) reduce extra parameter increment, and (3) maintain accuracy. In addition to the empirical evidence, this hypothesis is also supported by our theoretical evidence. Furthermore, we propose a simple yet effective reinforcement-learning-based method to search the ticket, *i.e.*, the connection scheme that satisfies the three above-mentioned conditions. Extensive experiments on widely-used benchmark datasets and popular self-attention networks show the effectiveness of our method. Besides, our experiments illustrate that our searched ticket has the capacity of transferring to some vision tasks, *e.g.*, crowd counting and segmentation.

***Index Terms*—Self-attention, Lottery Ticket Hypothesis, Reinforcement Learning, Neural Architecture Search.**

I. Introduction

RECENTLY, various plug-and-play self-attention modules (SAMs) which enhance instance specificity by the interior network information [1] are proposed to boost the generalization of convolutional neural networks (CNNs) [2]–[7]. The SAM is usually plugged into every block of a CNN, *e.g.*, the residual block of ResNet [8]. We display the structure of a ResNet in Fig. 1 (a) and a full self-attention network (Full-SA) whose each block connects to an individual SAM as in Fig. 1 (b). As illustrated in Fig. 1, the implementation of SAMs incorporates three steps: extraction, processing, and recalibration. These operations and trainable components in SAMs require extra computational cost and parameters, resulting in slow inference and cumbersome network [9]. This limits self-attention usability on industrial applications that need a real-time response or small memory consumption, such as robotics, self-driving car, and mobile device. Therefore, other than only improving the capacity of the SAM, previous works also focus on the light-weight SAM design, *e.g.*, reducing the parameters of an individual SAM [4], [10].

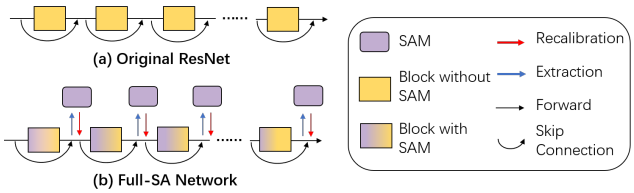

Fig. 1. (a) Original ResNet; (b) Full-SA network. A network is called a Full-SA network if the SAM is individually defined for each block. The SAM can be divided into three steps [5]: (1) Extraction: the plug-in module extracts internal features of a network by computing their statistics, like mean, variance; (2) Processing: the SAM utilizes the extracted features to adaptively generate a mask via a trainable module; (3) Recalibration: the mask is used to calibrate the feature maps by element-wise multiplication or addition.

However, the extra cost of the lightweight SAM is still non-negligible for a deep network [4], [9]. The main reason lies in the conventional paradigm where the SAMs are individually plugged into every block of a CNN for granted [2], [3]. The additional inference time and the number of parameters increase with the growth of the network depth, which creates a bottleneck when applying SAMs to a deep network. On the other hand, the network compression algorithms, such as network pruning [11]–[13] and neural architecture search [14], [15], effectively reduce the network size by removing the redundant components while maintaining the accuracy of the slimmed network. Note that these techniques inherently alternate the connections between different ingredients (*e.g.*, neurons, layers, or weights) in the CNN. This inspires us to pay more attention to the connections between the SAMs and the CNN backbone instead of the individual SAM design. If the number of connections is lessened, the computation and parameter cost will be reduced obviously. Based on these motivations, we articulate the Lottery Ticket Hypothesis for Self-attention Networks (LTH4SA):

A full self-attention (Full-SA) network contains a subnetwork with sparse self-attention connections that can (1)

* Zhongzhan Huang and Senwei Liang have equal contributions.

† Correspondence should be addressed to yang1863@purdue.edu, linliang@ieee.org.

Zhongzhan Huang and Liang Lin are with School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: huangzhzh23@mail2.sysu.edu.cn; linliang@ieee.org).

Senwei Liang and Haizhao Yang are with the Department of Mathematics, Purdue University, West Lafayette, IN, United States (e-mail: liang339@purdue.edu; yang1863@purdue.edu)

Mingfu Liang is with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, United States (e-mail: mingfuliang2020@u.northwestern.edu).

Wei He is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (email: wei005@e.ntu.edu.sg).

♣ Technical report

accelerate inference, (2) reduce extra parameter increment, and (3) maintain accuracy.

Every SAM connects or disconnects to the block, and we call the set of these connection states for a CNN as a connection scheme (see Section II-B). A connection scheme is called a ticket if the self-attention subnetwork with this scheme satisfies the three above-mentioned conditions of LTH4SA. In Section III, for the first time we both empirically and formally investigate the existence of LTH4SA. Our main observations are: (1) Empirically, there exist some self-attention subnetworks with sparse connection schemes achieving even better accuracy than the full self-attention network, which is also supported by our theoretical evidence where the large network can be approximated by its subnetwork; (2) The additional statistical analysis of the connection scheme shows that no specific block of the CNN dominates the accuracy when connecting SAM to the block. These observations indicate the key to obtaining a ticket is how to combine different connections of block and the SAMs. Certainly, finding the combination of the blocks and the SAMs to be a ticket is equivalent to solving a searching problem, and the corresponding algorithm should satisfy the following requirements based on the definition of LTH4SA and our empirical observations: (1) The searched connection scheme should be sparse and accurate enough to be a ticket; (2) The algorithm itself should have sufficient capacity to cover diverse connection schemes. Therefore, we propose a simple yet effective reinforcement-learning-based (RL-based) baseline method to search for a ticket given that the RL-based method can naturally handle multi-targets searching problems, and the rewards are designed exactly based on the requirements mentioned above. We call our proposed baseline method Efficient Attention Network (EAN). In Section II, we will briefly review the formulation of self-attention networks. Our proposed method for searching a ticket is introduced in Section IV and extensive experiments on widely-used benchmark datasets and popular self-attention networks are shown in Section V. The property of EANs will be discussed in Section VI. Finally, we discuss the related works in Section VII. We summarize **our contribution** as follows:

1) We empirically find some counterintuitive phenomena: (a) Connecting the SAMs to all the blocks may not bring the largest performance boost; (b) Some connection schemes are harmful.
2) We propose a lottery ticket hypothesis for self-attention networks and provide both numerical and theoretical evidence for the existence of the ticket. Besides, we propose an effective searching method as a baseline to obtain a ticket and avoid harmful connection schemes.

II. Preliminaries

In this section, we first briefly review ResNet [8] and the Full-SA network and then introduce the connection scheme.

A. ResNet and the Full Self-attention (Full-SA) Network

ResNet. The structure of a ResNet is shown in Fig. 1 (a). In general, the ResNet has several stages, and each stage, whose feature maps have the same size, is a collection of consecutive blocks. Suppose a ResNet has m blocks. Let x_ℓ be the input of the ℓ^{th} block and $f_\ell(\cdot)$ be the residual mapping, then the output $x_{\ell+1}$ of the ℓ^{th} block is defined as

$$x_{\ell+1} = x_\ell + f_\ell(x_\ell). \tag{1}$$

Full Self-attention (Full-SA) Network. A network is called a Full-SA network if the SAM is individually defined for each block as Fig. 1 (b). Note that the term "full" refers to a scenario when all blocks in a network connect to the SAMs. Many popular SAMs adopt this way to connect with the ResNet backbone [2], [3]. We denote the SAM in the ℓ^{th} block as $M(\cdot; W_\ell)$, where W_ℓ are the parameters. Then the attention will be formulated as $M(f_\ell(x_\ell); W_\ell)$ which consists of the extraction and processing operations introduced in Fig. 1. In the recalibration step, the attention is applied to the residual output $f_\ell(x_\ell)$, *i.e.*,

$$x_{\ell+1} = x_\ell + M(f_\ell(x_\ell); W_\ell) \odot f_\ell(x_\ell), \tag{2}$$

where $\ell = 1, ..., m$ and $\odot$ is the element-wise multiplication. Eq.(2) indicates that the computational cost and the number of parameters grow with the increasing number of blocks m.

B. Connection Scheme

Suppose that a ResNet has m blocks. A sequence $\mathbf{a} = (a_1, a_2, \cdots, a_m)$ denotes a connection scheme, where $a_i = 1$ if the i^{th} block is connected to a SAM, otherwise it equals 0. A subnetwork specified by a scheme $\mathbf{a}$ can be formulated by:

$$x_{\ell+1} = x_\ell + \Big(a_\ell \cdot M(f_\ell(x_\ell); W_\ell) + (1 - a_\ell) \cdot \mathbf{1}\Big) \odot f_\ell(x_\ell), \tag{3}$$

where $\mathbf{1}$ denotes an all-one vector and ℓ is from 1 to m. In particular, it becomes a Full-SA network if $\mathbf{a}$ is an all-one vector, or an original ResNet if $\mathbf{a}$ is a zero vector.

III. Lottery Ticket Hypothesis for Self-attention

In this section, we first study the existence of LTH4SA from empirical and theoretical perspectives. Then we investigate which block we should connect the SAMs to such that the corresponding connection scheme can achieve good accuracy.

A. Empirical evidence of LTH4SA Existence

We empirically validate the proposed LTH4SA by investigating the accuracy of the self-attention subnetwork under different connection schemes.

Specifically, we conduct classification on CIFAR100 using SAMs with the shallow and deep network backbones, *i.e.,* ResNet38 and ResNet164, under different SAM connection ratios, *i.e.,* the ratio of the number of connections to the number of blocks. Squeeze-and-Excitation SAM [2] is used in our experiments. We traverse all the connection schemes for ResNet38. However, as ResNet164 contains 54 blocks, there are 2^{54} different connection schemes. Traversal of all schemes is infeasible, and hence we randomly sample 100 connection schemes under each ratio. For simplicity and clear clarification, we choose the connection ratio 0.2, 0.4, 0.6, 0.8

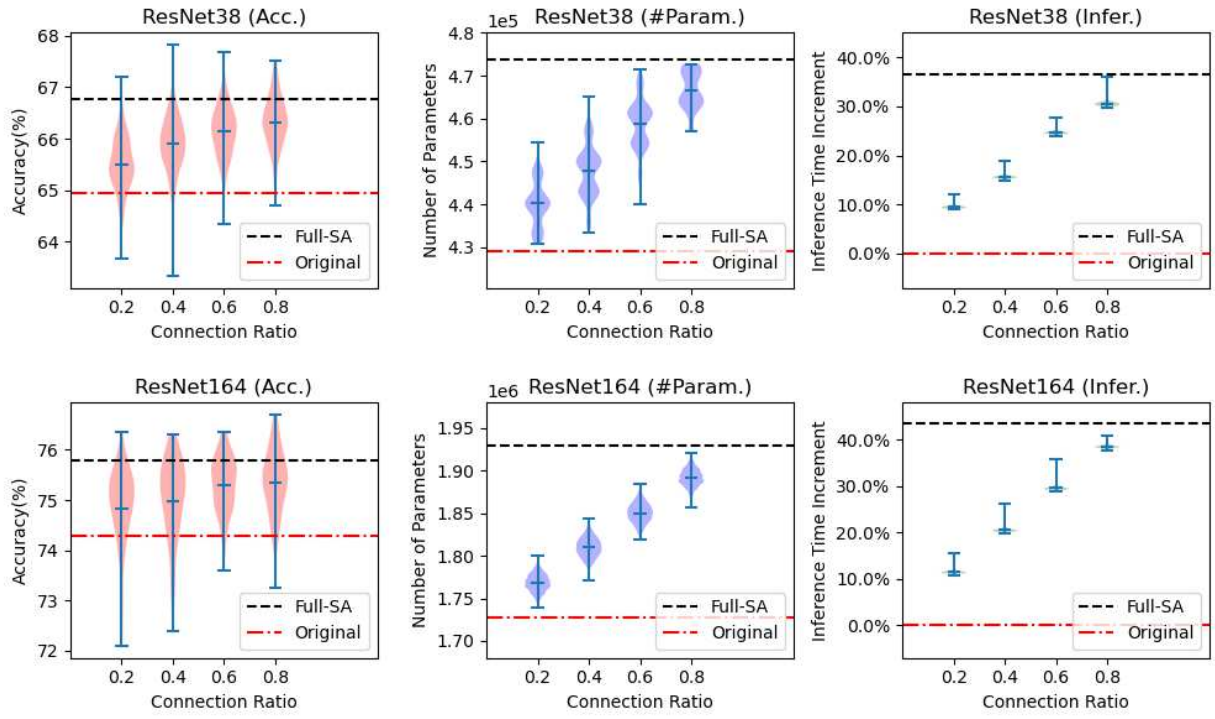

Fig. 2. Violin plot of accuracy, number of parameters, and inference time increment under different connection ratios. Three bars of each vertical line from the top to the bottom represent maximum, mean, and minimum, respectively. The light color shows the distribution. The black dotted line is the performance of the Full-SA network while the red line is the performance of the original CNN. The network with accuracy higher than the black dotted line is a ticket.

to present the empirical results given that they are sufficient to cover different sparsity levels. Fig. 2 displays the distribution of accuracy, the number of parameters, and inference time increment under different connection ratios. From Fig. 2, we observe that:

1) For different SAM connection ratios, there exist self-attention subnetworks with higher accuracy than the full self-attention network, even when the connections are very sparse, *e.g.*, the connection ratio is 0.2.
2) For different SAM connection ratios, there exist self-attention subnetworks with lower accuracy than the original CNN, which means some connection schemes are harmful.
3) Under the same connection ratio, the parameter and inference time of different connection schemes vary considerably.

Observation 1 shows the existence of the subnetwork with sparse connections yet good accuracy, which empirically illustrates the potential of finding some connection schemes that satisfy LTH4SA. Moreover, Observation 2 and Observation 3 reveal that even though there exist some connection schemes that satisfy the three conditions of LTH4SA, it is still necessary to carefully design methods to find tickets as the sparse connection scheme may harm the accuracy of the network and incur larger parameters and computational cost.

B. Theoretical Evidence of LTH4SA Existence

In Section III-A, the empirical evidence of the existence of LTH4SA has been demonstrated. Now we provide some theoretical evidence of the existence as well. The hidden neuron can be considered as a SAM as they apply internal information from the previous layer to the following network outputs. Hence, we can consider a more general scenario that given a network, there exists a subnetwork that approximates the original network and can be obtained by removing the hidden neurons from the original network.

We first consider a 1-hidden-layer feed-forward network and the network follows the initialization as [16].

Theorem 1. *A 1-hidden-layer feed-forward NN is defined as* $NN(x) = W^2\sigma(W^1x)$*, where input* $x \in R^d$ *with* $\|x\|_2 \leq 1$*,* W^1 *is of size* $m \times d$*,* W^2 *is of size* $1 \times m$ *and* σ *is ReLU activation.* $W^1_{i,j}$ *is initialized i.i.d. by the Gaussian distribution* $\mathcal{N}(0, (\frac{1}{\sqrt{m}})^2)$*, and* $W^2_{1,j}$ *is initialized by the uniform distribution* $Uniform\{1, -1\}$*. Let* $\mathcal{P}(d-1, \epsilon)$ *be* $\mathbb{P}\{\chi^2(d-1) \geq \epsilon^2\}$*, where* $\chi^2(d-1)$ *is a chi-square variable with* $d-1$ *degree of freedom. Then for any* $\epsilon, \delta > 0$*, when the number of hidden neurons* $m > \frac{\ln(\delta)}{\ln(\mathcal{P}(d-1,\epsilon))}$*, then there exists the row* j *of* W^1 *such that when we set the row* j *to be zero, i.e.,* B_jW^1 *with* $B_j = diag\{1, \cdots, 1, 0, 1, \cdots, 1\}$ *(the* j^{th} *entry is 0), we have*

$$\|W^2\sigma(W^1x) - W^2\sigma(B_jW^1x)\|_2 < \epsilon,$$

with probability higher than $1 - \delta$*.*

The proof for Thm. 1 is in Appendix. Thm. 1 shows that when the width of the network is sufficiently large, we can find a subnetwork that approximates the original network with high probability. Next, we consider a more general and modern network structure, *i.e.*, the ResNet [8] with ReLU.

Theorem 2. *Let $T(x)$ be a Lipschitz continuous and Lebesgue integrable function in d-dimensional compact set K. And $R_{full}(x, \theta_{full})$ is a ReLU ResNet structure with parameters θ_{full}. Let $\epsilon_0 > 0$ be a constant. Suppose that there exists θ^0_{full} such that $\int_K |R_{full}(x, \theta^0_{full}) - T|dx \leq \frac{\epsilon_0}{2}$. If the width of each layer in $R_{full}(x, \theta_{full})$ is larger than d and the depth of $R_{full}(x, \theta_{full})$ is larger than a constant that depends on ϵ_0, then for any $\epsilon \in (\epsilon_0, 1)$, there exists a subnetwork $R_{sub}(x)$ of $R_{full}(x, \theta_{full})$ such that*

$$\int_K |R_{full}(x, \theta^0_{full}) - R_{sub}(x)|dx \leq \epsilon. \tag{4}$$

The proof for Thm. 2 is in Appendix. In industrial applications, it is not necessary for the discrepancy between the output of two networks to be arbitrarily small if they have comparable performance. In practice, the discrepancy is acceptable if it reaches some certain levels, such as $\epsilon_0 = 10^{-5}$ or 10^{-10}. When a sufficiently small discrepancy ϵ_0 is given, Thm. 2 can guarantee that a large-size network contains a subnetwork that has similar performance.

C. Which block should we connect the SAMs to?

By studying the statistical characteristics of tickets and some harmful connection schemes, in this part, we investigate which block we should connect the SAMs to such that the corresponding connection scheme can achieve good accuracy.

We consider a statistics called connection score which characterizes the frequency of the connections of a scheme set. Given a network with m blocks and a set of N connection schemes with each scheme $\mathbf{a}_i = (a_{i1}, a_{i2}, ..., a_{im}), a_{ij} \in \{0, 1\}, i = 1, ..., N, j = 1, ..., m$, we define the connection score of a scheme set as follows,

$$\left(\frac{1}{N}\sum_{i=1}^{N} a_{i1}, \frac{1}{N}\sum_{i=1}^{N} a_{i2}, ..., \frac{1}{N}\sum_{i=1}^{N} a_{im}\right). \tag{5}$$

This statistic can characterize the importance of each block based on their frequency of connecting with the SAM. If the connection score of a block is large, the connection between this block and the SAM appears in large portion among N connection schemes.

We consider two sets of connection schemes, *i.e.*, the ticket set and the bad scheme set, for ResNet38 as presented in Fig. 3. The ticket set stands for the set of schemes that satisfy the LTH4SA, while the bad scheme set stands for a set of schemes whose accuracy is lower than the original network.

From Fig. 3, we can observe that the connection score of each block is almost the same for both the ticket set and bad scheme set. Besides, for each set, we use univariate linear regression to fit these scores, and use slope to characterize their trends. We can see that the slopes of two scheme set are close to zero. These observations indicate no specific block of the network will dominate the accuracy, and each block can be connected to the SAMs with almost equal frequency in a ticket.

Since no specific block dominates the accuracy, it is not easy to define a metric to identify the importance for the connection of each block as the network pruning algorithms do [17], [18]. Hence, it is necessary to design an effective search method to find the tickets from the thousands of possible connection schemes.

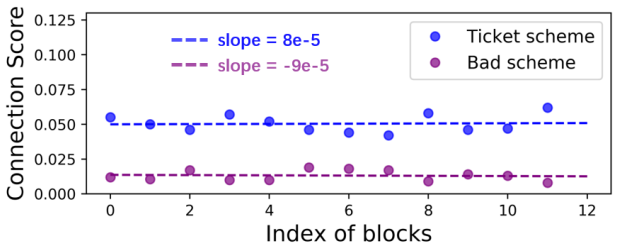

Fig. 3. The comparison of the connection scores of the ticket set or the bad scheme set with ResNet38.

Algorithm 1 Searching a ticket of LTH4SA

Input: Training set D_{train}; validation set D_{val}; a Full-SA attention network $\Omega(\mathbf{x}|\mathbf{1})$; the pre-training step K; the searching step T; the probability of retaining connection β.
Output: The trained controller $\chi_\theta(q_0)$.

▷ Pre-train the supernet
for t from 1 to K **do**
  $\mathbf{a} \sim [Bernoulli(\beta)]^m$
  train $\Omega(\mathbf{x}|\mathbf{a})$ with D_{train}
end for
▷ Policy-gradient-based search
for t from 1 to T **do**
  $\mathbf{p}_\theta \leftarrow \chi_\theta(q_0)$
  $\mathbf{a} \sim \mathbf{p}_\theta$
  Calculate the rewards g_{spa},g_{val},g_{rnd} (see Appendix)
  Update the trainable parameters.
end for
return $\chi_\theta(q_0)$

IV. Proposed Baseline Method

In this section, we introduce the proposed method which consists of two parts. First, we pre-train a supernet as the search space. The supernet assembles different candidate network architectures into a single network by weight sharing [19]. Each candidate architecture corresponds to a subnetwork and in our problem, each connection scheme corresponds to a specific subnetwork sampled from the supernet. Second, we use a policy-gradient-based method to search for an optimal connection scheme from the supernet. The basic workflow of our method is shown in Alg.1.

A. Problem Description

According to LTH4SA, our goal is (1) to find a connection scheme $\mathbf{a}$, which is sparse enough for less computational cost and parameters, from 2^m possibilities; (2) to ensure that the subnetwork specified by the scheme can maintain the accuracy as the Full-SA network.

To determine the optimal architecture from the pool of candidates, it is costly to evaluate all the candidates' performances after training from scratch, since even training a

candidate individually from scratch will require a large number of computation times (e.g., tens of hours), not to mention traversing such an extensive pool. In many related works on Neural Architecture Search (NAS), the validation accuracy of the candidates sampled from a supernet can be served as a satisfactory performance proxy [15], [20], [21] to approximately estimate those candidates' stand-alone[1] performance, which can effectively reduce the extensive computational cost correspondingly. Thus similarly, to efficiently obtain the optimal connection scheme, we propose to train the supernet as the search space. We follow the DropAct [22] training strategy to train the supernet. Then we consider the validation performance of the sampled subnetworks from the supernet as the proxy for their stand-alone performance. We consider a supernet $\Omega(\mathbf{x}|\mathbf{a})$ with m blocks and input $\mathbf{x}$. $\Omega(\mathbf{x}|\mathbf{a})$ has the same components as a Full-SA network, but its connections between blocks and SAMs are specified by $\mathbf{a}$.

B. Pre-training the Supernet

Given a dataset, we split all training samples into the training set D_{train} and the validation set D_{val}. To train the supernet, we activate or deactivate the SAM in each block of it randomly during optimization. Specifically, we first initialize a supernet $\Omega(\mathbf{x}|\mathbf{a}^{(0)})$, where $\mathbf{a}^{(0)} = (1, \cdots, 1)$. At the iteration t, we randomly draw a connection scheme $\mathbf{a}^{(t)} = (a_1^t, \cdots, a_m^t)$, where a_i^t is sampled from a Bernoulli distribution $Bernoulli(\beta)$. Then we train subnetwork $\Omega(\mathbf{x}|\mathbf{a}^{(t)})$ with the scheme $\mathbf{a}^{(t)}$ from the supernet on D_{train} via weight sharing. More detail of training the supernet is provided in Appendix.

C. Training Controller with Policy Gradient

We introduce the steps to search for the optimal connection scheme. Concretely, we use a controller to generate connection schemes and update the controller by policy gradient.

We use a fully connected network as the controller $\chi_\theta(q_0)$ to produce the connection schemes, where θ are the learnable parameters, and q_0 is a constant vector $\mathbf{0}$. The output of $\chi_\theta(q_0)$ is $\mathbf{p}_\theta$, where $\mathbf{p}_\theta = (p_\theta^1, p_\theta^2, ..., p_\theta^m)$ and p_θ^i represents the probability of connecting the SAM to the i^{th} block. A realization of $\mathbf{a}$ is sampled from the controller output, *i.e.*, $\mathbf{a} \sim \mathbf{p}_\theta$. The probability associated with the scheme $\mathbf{a}$ is $\hat{\mathbf{p}}_\theta = (\hat{p}_\theta^1, \hat{p}_\theta^2, ..., \hat{p}_\theta^m)$, where $\hat{p}_\theta^i = (1 - a_i)(1 - p_\theta^i) + a_i p_\theta^i$.

We denote $G(\mathbf{a})$ as a reward for $\mathbf{a}$. The parameter set θ within the controller can be updated via policy gradient with learning rate η, *i.e.*,

$$R_\theta = G(\mathbf{a}) \cdot \sum_{i=1}^{m} \log \hat{p}_\theta^i, \quad \theta \leftarrow \theta + \eta \cdot \nabla R_\theta. \tag{6}$$

In this way, the controller tends to output the probability that results in a large reward G. Therefore, designing a reasonable G can help us search for a good structure.

To find a ticket, we should incorporate the accuracy and connection ratio into the reward G. We use the validation accuracy g_{val} of the subnetwork $\Omega(\mathbf{x}|\mathbf{a})$ sampled from the supernet as a reward, which depicts the performance of its structure. Besides, we complement a sparsity reward g_{spa} to encourage the controller to generate the schemes with fewer connections between SAMs and backbone. Finally, to encourage the controller to explore more potentially useful connection schemes, we add the Random Network Distillation (RND) curiosity bonus g_{rnd} in our reward [23]. Therefore, $G(\mathbf{a}) = \lambda_1 \cdot g_{\text{spa}} + \lambda_2 \cdot g_{\text{val}} + \lambda_3 \cdot g_{\text{rnd}}$, where $\lambda_1, \lambda_2, \lambda_3$ are the coefficient for each bonus. The detailed definition of $g_{\text{spa}}, g_{\text{val}}$, and g_{rnd} can be found in Appendix.

[1]Train the subnetworks from scratch

V. Experiments

In this section, we demonstrate the effectiveness of our method in finding the ticket. First, we show that our method can outperform some popular NAS and pruning algorithms. Next, to further reduce the number of parameters for various types of SAMs, we search the ticket from another self-attention framework proposed in [5]. Finally, we conduct a comprehensive comparison with various searching methods.

A. Datasets and Settings

On CIFAR100 [24] and ImageNet2012 [25] datasets, we conduct classification using ResNet [8] backbone with different SAMs, including Squeeze-and-Excitation (SE) [2], Spatial Group-wise Enhance (SGE) [4] and Dense-Implicit-Attention (DIA) [5] modules. The description of these SAMs is in Appendix. Since the networks with SAMs have extra computational cost compared with the original backbone inevitably, we formulate the relative inference time increment to represent the relative speed of different self-attention networks, *i.e.*,

$$\frac{I_t(\text{CNN with SAMs}) - I_t(\text{Original CNN})}{I_t(\text{Original CNN})} \times 100\%, \tag{7}$$

where $I_t(\cdot)$ denotes the inference time of the network. The inference time is measured by forwarding the data of batch size 50 for 1000 times.

CIFAR100. CIFAR100 consists of 50k training images and 10k test images of size 32 by 32. In our implementation, we choose 10k images from the training images as a validation set (100 images for each class, 100 classes in total), and the remainder images as a sub-training set. Regarding the experimental settings of ResNet164 [8] backbone with different SAMs, the supernet is trained for 150 epochs, and the search step T is set to be 1000.

ImageNet2012. ImageNet2012 comprises 1.28 million training images. We split 100k images (100 from each class and 1000 classes in total) as the validation set and the remainder as the sub-training set. The testing set includes 50k images. Besides, the random cropping of 224 by 224 is used. Regarding the experimental settings of ResNet50 [8] backbone with different SAMs, the supernet is trained for 40 epochs, and the search step T is set to be 300.

B. Searching Tickets from the Full-SA Network

In this part, we compare our search method with some popular NAS and pruning algorithms, *e.g.*, Genetic Algorithm (GA) [14], ENAS [26] and DARTS [27], ℓ_1 pruning, and

TABLE I

COMPARISON OF RELATIVE INFERENCE TIME INCREMENT (DENOTED BY INFERE. (%) AS IN EQ.(7)), THE NUMBER OF PARAMETERS (#P (M)), AND TEST ACCURACY (ACC.) ON CIFAR100. HERE THE CONNECTION SCHEMES ARE SEARCHED WITH SE MODULE ON RESNET38 AND RESNET164, USING DIFFERENT SEARCHING METHODS. "TICKET?" PRESENTS WHETHER THE FOUND CONNECTION SCHEME IS A TICKET OR BELONGS TO TOP 5% HIGH ACCURACY. WE TRAIN A SUPERNET WITH THE PROBABILITY β OF RETAINING CONNECTION AS IN ALG. 1.

ResNet38						
β	Method	Acc.	#P (M)	Infere. (%)	Top5%?	Ticket?
0.2	ENAS	64.94	0.43	0.00	✗	✗
	DARTS	66.07	0.44	13.04	✗	✗
	GA	65.25	0.47	32.88	✗	✗
	ℓ_1	66.06	0.45	6.37	✗	✗
	GM	66.39	0.45	6.45	✗	✗
	EAN	**66.78**	0.45	23.21	✗	✓
0.5	ENAS	65.09	0.45	26.55	✗	✗
	DARTS	66.06	0.44	20.02	✗	✗
	GA	65.57	0.44	19.82	✗	✗
	ℓ_1	65.86	0.47	20.00	✗	✗
	GM	66.27	0.46	19.50	✗	✗
	EAN	**66.90**	0.45	26.52	✓	✓
0.8	ENAS	66.77	0.47	38.98	✗	✗
	DARTS	66.67	0.46	29.55	✗	✗
	GA	66.21	0.46	33.61	✗	✗
	ℓ_1	66.32	0.47	29.60	✗	✗
	GM	66.61	0.47	29.60	✗	✗
	EAN	**67.06**	0.46	19.98	✓	✓
	Original CNN	64.94	0.43	0.00	-	-

ResNet164					
β	Method	Acc.	#P (M)	Infere. (%)	Ticket?
0.2	ENAS	74.29	1.73	0.00	✗
	DARTS	74.42	1.74	8.43	✗
	GA	76.07	1.80	15.64	✓
	ℓ_1	74.50	1.81	8.92	✗
	GM	75.09	1.81	8.24	✗
	EAN	**76.53**	1.85	25.84	✓
0.5	ENAS	75.33	1.81	18.13	✗
	DARTS	73.44	1.85	21.97	✗
	GA	75.75	1.76	15.21	✗
	ℓ_1	73.79	1.90	23.23	✗
	GM	75.37	1.90	22.04	✗
	EAN	**76.21**	1.82	22.41	✓
0.8	ENAS	75.80	1.93	43.56	✗
	DARTS	75.42	1.90	34.42	✗
	GA	75.10	1.78	13.91	✗
	ℓ_1	**76.27**	1.92	34.66	✓
	GM	75.55	1.92	34.29	✗
	EAN	75.80	1.82	20.64	✓
	Original CNN	74.29	1.73	0.00	-

Geometric Median (GM) pruning [18]. Table I displays the experiments conducted on CIFAR100 with Full-SA networks including ResNet38 and ResNet164 using SE for different searching methods under different supernets.

From Table I, our method outperforms the heuristic method GA. Different from DARTS that searches schemes by minimizing the validation loss, the RL-based method (e.g., ENAS and our baseline method) can directly consider the validation accuracy as a reward although the accuracy or sparsity constraint is not differentiable. However, ENAS does not learn an effective scheme from the reward because of its controller architecture as discussed in Section V-D.

Since the pruning algorithms (ℓ_1 and GM) are based on intuitive design [28] to measure the importance of the connection individually and ignore the combination of the connections of a network as mentioned in Section III-C, they may fail to find the reasonable connection scheme and obtain unstable results in some cases.

C. Searching Tickets from Full-Share Network

We demonstrate that our search method can also be applied to other self-attention frameworks, such as sharing mechanism [5], which shares a SAM with the same set of parameters to different blocks in the same stage. We call a network as Full-Share network if each block in the same stage of this network connects to a shared SAM. The shared SAM significantly reduces the trainable parameters compared with the Full-SA network.

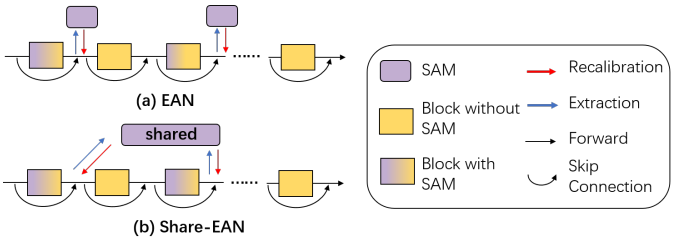

Fig. 4. The network structures of (a) EAN and (b) Share-EAN.

The connection scheme found by our method from a Full-Share network is called Share-EAN, as illustrated in Fig. 4 (b). We show the test accuracy, the number of parameters, and relative inference time increment on CIFAR100 and ImageNet2012 of Share-EAN in Table II, searching from the supernet trained with retaining ratio $\beta = 0.5$.

From Table II, we can observe that most results of Share-EAN satisfy the three conditions of LTH4SA. Since both Share-EAN and the Full-Share network use sharing mechanism [5] to implement the SAM over the original ResNet, they both have fewer parameters increment than the Full-SA network. Besides, Share-EANs achieve faster inference speed among the Full-Share network and the Full-SA network. Furthermore, the accuracy of Share-EANs is on par or even surpass that of the Full-Share networks.

TABLE II
COMPARISON OF THE RELATIVE INFERENCE TIME INCREMENT (SEE EQ.(7)), THE NUMBER OF PARAMETERS, AND TEST ACCURACY BETWEEN VARIOUS SAMS ON CIFAR100 AND IMAGENET2012. "ORIGINAL CNN" STANDS FOR RESNET164 BACKBONE IN CIFAR100 AND RESNET50 BACKBONE IN IMAGENET. THE SAM IN DIA IS A RNN SO DIA NETWORK DOES NOT HAVE FULL-SA STRUCTURE.

Dataset	Model	Test Accuracy (%)			Parameters (M)			Relative Inference Time Increment (%)		
		Full-SA	Full-Share	Share-EAN	Full-SA	Full-Share	Share-EAN	Full-SA	Full-Share	Share-EAN
CIFAR100	Original CNN	74.29	-	-	1.727	-	-	0.00	-	-
	SE [2]	75.80	76.09	**76.93**	1.929	1.739	**1.739**	43.56	41.66	**18.81** (↓ 22.85)
	SGE [4]	75.75	76.17	**76.36**	1.728	1.727	**1.727**	93.60	93.41	**50.49** (↓ 42.92)
	DIA [5]	-	**77.26**	77.12	-	1.946	**1.946**	-	121.11	**65.46** (↓ 55.65)
ImageNet	Original CNN	76.01	-	-	25.584	-	-	0.00	-	-
	SE [2]	77.01	77.35	**77.40**	28.115	26.284	**26.284**	25.94	25.92	**10.35** (↓ 15.57)
	SGE [4]	77.20	77.51	**77.62**	25.586	25.584	**25.584**	40.60	40.50	**19.66** (↓ 20.84)
	DIA [5]	-	77.24	**77.56**	-	28.385	**28.385**	-	27.26	**16.58** (↓ 10.68)

TABLE III
COMPARISON OF THE ACCURACY AND RELATIVE INFERENCE TIME INCREMENT OF THE SEARCHED NETWORK FOR DIFFERENT METHODS.

Method	Acc.	Time Increment (%)	Method	Acc.	Time Increment (%)
Share-EAN	**76.93**	18.81 (↓22.85)	HSP (101010..)	75.02	21.29 (↓20.37)
DARTS	75.41	28.02 (↓13.64)	HSP (010101..)	74.87	21.29 (↓20.37)
GA	76.09	20.87 (↓20.79)	HSP (100100..)	75.29	14.50 (↓27.16)
ENAS	76.08	28.05 (↓13.61)	HSP (010010..)	74.01	14.50 (↓27.16)

D. Comprehensive Comparison with Searching Methods

In this part, we compare our method (EAN) with heuristic selection policy (HSP), Genetic Algorithm (GA) [14], ENAS [26] and DARTS [27] for the Full-Share network. HSP is a heuristic policy that makes SAM connection every N layers. For example, when $N = 2$, the schemes can be $10101\cdots$ or $01010\cdots$. Table III displays the experiments conducted on CIFAR100 with ResNet164 and SE module for different searching methods. From Table III, our method achieve better results of the Full-Share network compared with other methods, which is consistent with the results of the Full-SA network in Section V-B. Besides, the heuristic design of the connection scheme like HSP does not give a scheme for good accuracy, and hence it is necessary for a careful search algorithm as mentioned in III-C.

The controller of ENAS tends to converge to some periodic-alike schemes at a fast speed. In this case, it will conduct much less exploration of the potential efficient structures. We show the list of connection schemes by ENAS (an example) in Table V. The majority of the schemes searched by ENAS are "111...111" (Full-Share network) or "000...000" (Original network), which shows that it can not strike the balance between the performance and inference time. In Table IV, the minority of the periodic-alike schemes searched by ENAS are shown, e.g., "001" in ENAS (a). Such schemes may result from the input mode of ENAS, *i.e.*, for a connection scheme $\mathbf{a} = (a_1, a_2, ..., a_m)$, the value of component a_l depends on $a_{l-1}, a_{l-2}, ..., a_1$. This strong sequential correlations let the sequential information dominate in the RNN controller instead of the policy rewards. Compared with the periodic-alike connection schemes searched by ENAS, Share-EAN demonstrates better performance.

Besides, our experiment indicates that ENAS explores a much smaller number of candidate schemes. We quantify the convergence of the controller using $\bar{\mathbf{p}} = \frac{1}{m}\sum_{i=1}^{m}\hat{p}_\theta^i$, which is the mean of the probability $\hat{\mathbf{p}}$ associated with the scheme. When $\bar{\mathbf{p}}$ is close to 1, the controller tends to generate a deterministic scheme. Fig. 5 and Table V show the curve of $\bar{\mathbf{p}}$ with the growth of searching iterations, where $\bar{\mathbf{p}}$ of ENAS shows the significant tendency for convergence in 20 iterations and converges very fast within 100 iterations. Generally speaking, methods in NAS [26], [29] require hundreds or thousands of iterations for convergence.

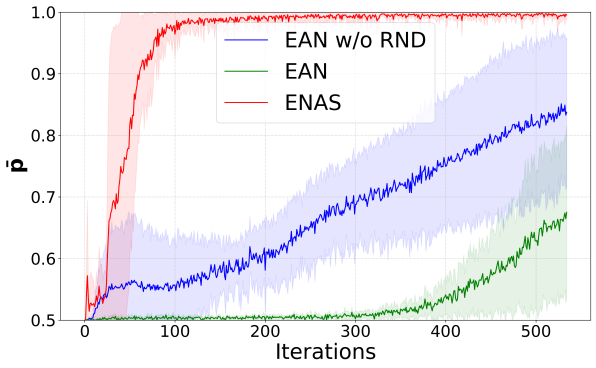

Fig. 5. Comparison of the convergence speed between ENAS and EAN. The controller tends to generate a deterministic scheme when $\bar{\mathbf{p}}$ is close to 1.

VI. ANALYSIS

In this section, we demonstrate that the found self-attention subnetwork has the capacity of capturing the discriminative features as the full network and transferring to the downstream tasks. Besides, we compare the training time and the searching time to show that our search time is acceptable.

A. Capturing Discriminative Features

To study the ability of Share-EAN in capturing and exploiting features of a given target, we apply Grad-CAM [30]

TABLE IV
THE CONNECTION SCHEMES SEARCHED BY ENAS [26] OR OUR METHOD. THE EXPERIMENT IS CONDUCTED ON CIFAR100 WITH SE MODULE AND RESNET164 BACKBONE.

Method	Stage1	Stage2	Stage3	Test Accuracy (%)
ENAS (a)	001001001001001001	001001001001001001	001001001001001001	75.80
ENAS (b)	100100101100100100	101101100100101101	100100101101100100	75.11
ENAS (c)	110110110110110110	110110110110110110	110110110110110110	76.08
Our method (a)	001100100101110101	001100000111001111	101100000111110001	**76.93**
Our method (b)	001100000001010111	011100001000010111	101000100110000000	**76.71**

TABLE V
THE CONNECTION SCHEME SEARCHED BY ENAS. $\bar{\mathbf{p}}$ IS THE AVERAGE OF THE PROBABILITY ASSOCIATED WITH THE SCHEME. THE CONTROLLER TENDS TO GENERATE A DETERMINISTIC SCHEME IF $\bar{\mathbf{p}}$ IS CLOSE TO 1. THE EXPERIMENT IS CONDUCTED ON CIFAR100 WITH RESNET164 AND SE MODULES.

Iteration	Connection Scheme	Sparse	$\bar{\mathbf{p}}$
0	000110000001111101110010000001000110111110110110010011	0.52	0.50
5	100100111101010011110001110101011111011100110001000011	0.44	0.51
10	100001001011101100010001100110111011101111011011000011	0.44	0.50
15	111001000110011101110001110010110111110110111101110001	0.37	0.57
20	111111001111111111000111001111001011011001111111110111	0.26	0.67
25	101111111111111110011111111000001000110111111100111111	0.26	0.64
30	011110011111111111111110001111111111111001111111101111	0.17	0.85
35	111111100011111111110111111111111111111111111111111111	0.07	0.91
40	101111111111111111111111111111111111111111111111111111	0.02	0.96
45	111111111111111101111111111111111111111111111111111111	0.02	0.98
50	011111111111111111110001111111111111111111111111111111	0.07	0.98
55	111111111100111111111111111111111111111111111111111111	0.04	0.98
60	111111111111111111111111111111111111111111111111111111	0.00	0.98
65	111111111111111111111111111111111111111111111111111111	0.00	0.99
70	111111111111111001111111111111111111111111111111111111	0.04	0.96
75	011111111111111111111111111111111111111111111111111111	0.02	0.99
80	111111111111111111111111111111111111111111111111111111	0.00	1.00
85	111111111111111111111111111111111111111111111111111111	0.00	1.00
90	111111111111111111111111111111111111111111111111111111	0.00	1.00
95	111111111111111111111111111111111111111111111111111111	0.00	0.98
100	111111111111111111111111111111111111111111111111111111	0.00	0.99
105	111111111111111111111111111111111111111111111111111111	0.00	1.00
110	111111111111111111111111111111111111111111111111111111	0.00	1.00
115	111111111111111111111111111111111111111111111111111111	0.00	1.00
120	111111111111111111111111111111111111111111111111111111	0.00	1.00
125	111111111111111111111111111111111111111111111111111111	0.00	1.00
130	111111111111111111111111111111111111111111111111111111	0.00	1.00
135	111111111111111111111111111111111111111111111111111111	0.00	0.99

to compare the regions where different models localize with respect to their target prediction. Grad-CAM is a technique to generate the heatmap highlighting network attention by the gradient related to the given target. Fig. 6 shows the visualization results and the softmax scores for the target with original ResNet50, Full-Share, and Share-EAN on the validation set of ImageNet2012. SE module is used in this part. The red region indicates an essential place for a network to obtain a target score while the blue region is the opposite. The results show that Share-EAN can extract similar features as Full-Share, and in some cases, Share-EAN can even capture much more details of the target associating with higher confidence for its prediction. This implies that the searched connection scheme may have a more vital ability to emphasize the more discriminative features for each class than the two baselines (original ResNet and Full-Share). Therefore it is reasonable to bring additional improvement on the final classification performance with Share-EAN in that the discrimination is crucial for the classification task, which is also validated from ImageNet test results in Table II.

B. Comparison of Training Time and Search Time

For NAS, we not only need to care about whether the search method can find a neural network structure that satisfies certain conditions but also need to focus on its computational cost. Taking the experiments ($\beta = 0.5$) in Table I as example, we measure the time of these experiments in Table VI. The time for training ResNet38 and ResNet164 from scratch is 1.61h, 4.46h on a single GPU 1080Ti while our RL-based method requires only 25.4%, 21.3% of the train time for searching from the supernet, respectively. The search time is acceptable and worthwhile as the found ticket may be applied to the downstream tasks that will be discussed in the next part.

TABLE VI
THE TRAINING TIME OF OUR BASELINE METHOD. "TRAIN TIME" DENOTES THE TIME OF TRAINING A NEURAL NETWORK FROM SCRATCH. "SEARCH TIME" DENOTES THE TIME OF OUR RL SEARCH. "SEARCH/TRAIN" REPRESENTS THE RATIO OF SEARCH TIME TO THE TRAIN TIME. ALL EXPERIMENTS ARE CONDUCTED ON A SINGLE GPU.

GPU	Model	Train Time	Search Time	Search/Train
1080Ti	ResNet38	1.61 hrs	0.41 hrs	25.40%
1080Ti	ResNet164	4.46 hrs	0.94 hrs	21.30%

C. Transferring Connection Schemes

In this part, we study the transferability of the network architecture searched by our baseline method. Specifically, we conduct experiments on transferring the optimal architecture from image classification to crowd counting task [31]–[34] and segmentation [35]. The model trained with classification is typically used to initialize the model for downstream tasks [36], [37]. If the found network have the transferability, we will have the advantages as follows: (I) We do not need to spend extra time searching for a ticket for the new task; (II) The model for the new task inherits a good representation ability of the pretrained model; (III) The model with fewer SAMs has less forward and back-propagation cost compared with Full-SAM. In this case, the computational cost of our RL-based search shown in Table VI is acceptable.

Crowd counting. Crowd counting aims to estimate the density map and predict the total number of people for a given image, whose efficiency is also crucial for many real-world applications, *e.g.*, video surveillance and crowd analysis. However, most state-of-the-art works still rely on the heavy pre-trained backbone networks [38] for obtaining satisfactory performance on such dense regression problems.

TABLE VII
COMPARISON OF PERFORMANCE BETWEEN DIFFERENT PRE-TRAINED MODELS ON CROWD COUNTING. SMALLER MAE/MSE IS BETTER.

Dataset	Model	MAE/MSE (↓)			Relative Inference Time Increment (%)		
		Full-SA	Full-Share	Share-EAN	Full-SA	Full-Share	Share-EAN
SHHB	SE [2]	9.5/15.93	8.9/14.6	8.6/14.7	19.19	19.19	**6.16** (↓ 13.03)
	DIA [5]	-	9.1/14.9	8.2/13.9	-	16.93	**8.71** (↓ 8.22)
SHHA	SGE [4]	93.9/144.5	91.6/143.1	88.4/140.0	58.98	58.85	**30.55** (↓ 28.30)
	SE [2]	89.9/140.2	89.9/140.2	79.4/127.7	49.50	49.00	**21.07** (↓ 27.93)
	DIA [5]	-	92.5/130.4	90.3/141.6	-	51.75	**29.43** (↓ 22.32)

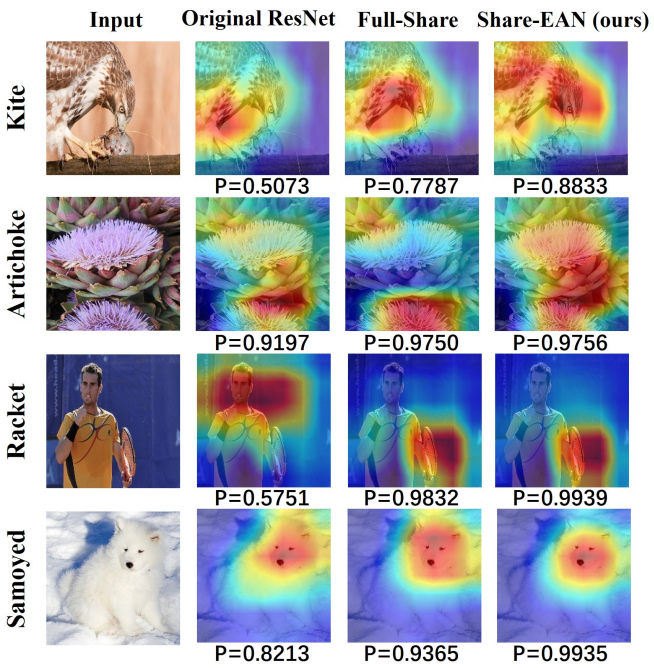

Fig. 6. Grad-CAM visualization of different networks. The red region indicates an essential place for a network to obtain a target score (**P**) while the blue one is the opposite.

The experiments show that the Share-EAN trained on ImageNet serves as an efficient backbone network and can extract the representative features for crowd counting. We evaluate the transferring performance on the commonly-used Shanghai Tech dataset [31], which includes two parts. Shanghai Tech part A (SHHA) has 482 images with 241,677 people counting, and Shanghai Tech part B (SHHB) contains 716 images with 88,488 people counting. Following the previous works, SHHA and SHHB are split into train/validation/test set with 270/30/182 and 360/40/316 images, respectively. The performance on the test set is reported using the standard Mean Square Error (MSE) and Mean Absolute Error (MAE), as shown in Table VII. Our Share-EANs outperform the baseline (Full-SA and Full-Share) while reducing the inference time increment by up to 28% compared with the baseline.

Semantic segmentation. We verify the transferability of the Share-EAN on semantic segmentation task in Pascal VOC 2012 [39] dataset. Table VIII shows the performance comparison of the backbone with different types of SAM, e.g., DIA and SE. Again, our results indicate that the Share-EAN can maintain the performance of the Full-Share network and significantly reduce the time increment compared with the Full-Share network, which shows Share-EAN has the capacity of transferring to segmentation.

TABLE VIII
PERFORMANCE AND RELATIVE INFERENCE TIME INCREMENT COMPARISON ON PASCAL VOC 2012 VALIDATION SET.

Model	mIoU/mAcc/allAcc (%)	Time Increment (%)
Original ResNet	69.39 / 78.87 / 92.97	-
Full-Share-SE	73.03 / 82.13 / 93.74	48.16
Share-EAN-SE	73.68 / 83.08 / 93.79	**16.43** (↓ 31.73)
Full-Share-DIA	74.02 / 83.11 / 93.92	64.86
Share-EAN-DIA	73.91 / 82.93 / 93.92	**7.68** (↓ 57.18)

VII. RELATED WORKS

Lottery Tickets Hypothesis (LTH). The Lottery Ticket Hypothesis [40] conjectures that: every random initialized and dense NN contains a subnetwork that can be trained in isolation with the original initialization to achieve comparable performance to the original NN. This original LTH attracts many researchers to rethink the training of the overparameterized model, leading to variants of LTH under different learning paradigms and machine learning fields [41]–[43]

On the other hand, Malach et al. [44] try to prove a LTH variants [41] by showing that given a target NN of depth l and width d, any random initialized network with depth $2l$ and width $O\left(d^5 l^2/\epsilon^2\right)$ contains subnetworks that can approximate the target network with ϵ error. Following works [45], [46] further reduce the width to $O(d\log(dl/\epsilon))$ for the random initialized network.

Neural Architecture Search (NAS). Designing a satisfactory neural architecture automatically, also known as neural architecture search, is of significant interest for academics and industries. Such a problem may always be formulated as searching for the optimal combination of different network granularities. The early NAS works require expensive computational costs for scratch-training a massive number of architecture candidates [29], [47]. To alleviate the searching cost, the recent advances of one-shot approaches for NAS bring up the concept of supernet based on the weight-sharing heuristic. Supernet serves as the search space embodiment of the candidate architectures, and it is trained by optimizing different sub-networks from the sampling paths, *e.g.*, SPOS [15], GreedyNAS [20].

Self-Attention Mechanism. The self-attention mechanism is widely used in CNNs for computer vision [1], [2], [4]–[7]. Squeeze-Excitation (SE) module [2] leverages global average pooling to extract the channel-wise statistics and learns the non-mutually-exclusive relationship between channels. Spatial Group-wise Enhance (SGE) module [4] learns to recalibrate features by saliency factors learned from different groups of the feature maps. Dense-Implicit-Attention (DIA) module [5] captures the layer-wise feature interrelation with a recurrent neural network (RNN).

VIII. Conclusion

Lottery Ticket Hypothesis for Self-attention Networks is proposed in this paper, which is supported by numerical and theoretical evidence. Then, to find a ticket, we propose an effective connection scheme searching method based on policy gradient as a baseline to find a ticket. The self-attention network found by our method can maintain accuracy, reduce parameters and accelerate the inference speed. Besides, we illustrate that the found network has the capacity of capturing the informative features and transferring to other computer vision tasks.

Appendix
Proof of Theorem 1

Theorem 1. A 1-hidden-layer feed-forward NN is defined as $NN(x) = W^2\sigma(W^1x)$, where input $x \in \mathbb{R}^{d\times 1}$ with $\|x\|_2 \leq 1$, W^1 is of size $m \times d$, W^2 is of size $1 \times m$ and σ is ReLU activation. $W^1_{i,j}$ is initialized i.i.d. by the Gaussian distribution $\mathcal{N}(0, (\frac{1}{\sqrt{m}})^2)$, and $W^2_{1,j}$ is initialized by the uniform distribution $Uniform\{1, -1\}$. Let $\mathcal{P}(d-1, \epsilon)$ be $\mathbb{P}\{\chi^2(d-1) \geq \epsilon^2\}$, where $\chi^2(d-1)$ is a chi-square variable with $d-1$ degree of freedom. Then for any $\epsilon, \delta > 0$, when the number of hidden neurons $m > \frac{\ln(\delta)}{\ln(\mathcal{P}(d-1,\epsilon))}$, then there exists the row j of W^1 such that when we set the row j to be zero, i.e., B_jW^1 with $B_j = diag\{1, \cdots, 1, 0, 1, \cdots, 1\}$ (the j^{th} entry is 0), we have

$$\|W^2\sigma(W^1x) - W^2\sigma(B_jW^1x)\| < \epsilon,$$

with probability higher than $1-\delta$.

Proof. Since $W^2_{1,j} \sim Uniform\{-1, 1\}$, we have $\|W_2\|_2 = \sqrt{m}$. We denote the row s of W^1 as $W^1_{s:}$. Let's consider the following probability,

$$\begin{aligned}\mathbb{P}\{\nexists s : \|W^1_{s:}\|_2 < \frac{\epsilon}{\sqrt{m}}\} &= \mathbb{P}\{\cup_{s=1}^m \|W^1_{s:}\|_2 \geq \frac{\epsilon}{\sqrt{m}}\} \\ &= \prod_{s=1}^m \mathbb{P}\{\|W^1_{s:}\|_2 \geq \frac{\epsilon}{\sqrt{m}}\} \\ &= \prod_{s=1}^m \mathbb{P}\{\frac{1}{m}\chi^2(d-1) \geq \frac{\epsilon^2}{m}\} \\ &= \mathcal{P}(d-1, \epsilon)^m.\end{aligned}$$

Let $\mathcal{P}(d-1,\epsilon)^m < \epsilon$, and then we have $m > \frac{\ln(\delta)}{\ln(\mathcal{P}(d-1,\epsilon))}$. Therefore, when $m > \frac{\ln(\delta)}{\ln(\mathcal{P}(d-1,\epsilon))}$, with probability greater than $1-\delta$, there exists j such that $\|W^1_{j:}\|_2 < \frac{\epsilon}{\sqrt{m}}$.

Let $G = diag\{W^1x \geq 0\}$, where the s-th diagonal component of G is 1 if $W^1_{s:}x \geq 0$ else 0. Then $\sigma(W^1x) = GW^1x$ and $\sigma(B_jW^1x) = GB_jW^1x$. Finally, we obtain

$$\begin{aligned}&\|W^2\sigma(W^1x) - W^2\sigma(B_jW^1x)\|_2 \\ =&\|W^2GW^1x - W^2GB_jW^1x\|_2 \\ \leq&\|W^2\|_2\|G\|_2\|W^1 - B_jW^1\|_2\|x\|_2 \\ \leq&\sqrt{m} \times 1 \times \|W^1_{j:}\|_2 \times 1 \leq \epsilon,\end{aligned}$$

with probability greater than $1-\delta$. □

Appendix
Proof of Theorem 2

Theorem 2. Let $T(x)$ be a Lipschitz continuous and Lebesgue integrable function in d-dimensional compact set K. And $R_{\text{full}}(x, \theta_{\text{full}})$ is a ReLU ResNet structure with parameters θ_{full}. Let $\epsilon_0 > 0$ be a fixed constant. Suppose that there exists θ^0_{full} such that $\int_K |R_{\text{full}}(x, \theta^0_{\text{full}}) - T|dx \leq \frac{\epsilon_0}{2}$. If the width of each layer in $R_{\text{full}}(x, \theta_{\text{full}})$ is larger than d and the depth of $R_{\text{full}}(x, \theta_{\text{full}})$ is larger than a constant that depends on ϵ_0, then for any $\epsilon \in (\epsilon_0, 1)$, there exists a subnetwork $R_{\text{sub}}(x)$ of $R_{\text{full}}(x, \theta_{\text{full}})$ such that

$$\int_K |R_{\text{full}}(x, \theta^0_{\text{full}}) - R_{\text{sub}}(x)|dx \leq \epsilon. \tag{8}$$

Lemma 1. *[48] For any $d \in \mathbb{N}$, the family of ResNet with one-neuron hidden layers and ReLU activation function can universally approximate any Lebesgue integrable function f. In other words, for any $\epsilon > 0$, there is a ResNet R with finitely many layers and width not larger than d such that*

$$\int_{\mathbb{R}^d} |f(x) - R(x)|dx \leq \epsilon. \tag{9}$$

Proof. See [48]. □

We use the notation $\text{dep}(\cdot)$ to denote the depth of a network.

Lemma 2. *(Extension strategy) Let T be a Lebesgue integrable d-dimensional function. For an $\epsilon > 0$, a ReLU ResNet $g(x, \theta^0_g)$ with parameters θ^0_g satisfies $\int_{\mathbb{R}^d} \left|g\left(x, \theta^0_g\right) - T\right| dx \leq \epsilon$, then there exists $f(x, \theta^0_f)$ with $\text{dep}(f) > \text{dep}(g)$ such that*

$$\int_{\mathbb{R}^d} \left|f\left(x, \theta^0_f\right) - T\right| dx \leq \epsilon. \tag{10}$$

Here $f(x, \theta^0_f)$ is obtained by adding some layers to the last layer of $g(x, \theta^0_g)$.

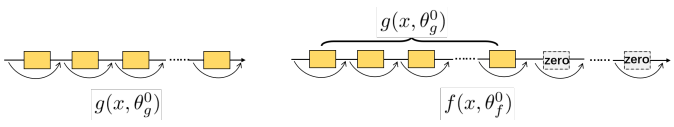

Fig. 7. The structure $f(x, \theta^0_f)$ and $g(x, \theta^0_g)$.

Proof. As shown in Fig. 7, we can expand g by adding some skip connection layers to its last layer. And then we set all the values of the parameters of the extra layers to zeros.

Through the skip connections, we have $g(x,\theta_g^0) = f(x,\theta_f^0)$ and $\text{dep}(g) < \text{dep}(f)$. Therefore,

$$\int_{\mathbb{R}^d} \left| f\left(x,\theta_f^0\right) - T \right| dx = \int_{\mathbb{R}^d} \left| g\left(x,\theta_g^0\right) - T \right| dx \leq \epsilon. \quad (11)$$

□

Lemma 3. *Let f be the Lebesgue integrable function defined on d-dimensional compact set $K \subset \mathbb{R}^d$. $\forall \epsilon > 0$, according to Lemma 1, there is a ResNet $R(x)$ with finitely many layers and width not larger than d such that $\int_K |f(x) - R(x)| dx \leq \epsilon$. Then the depth of $R(x)$ is $O(1/r^d)$, where r satisfies $\omega_K(r) \leq \epsilon/Vol(K)$ with $\omega_K(r)$ defined by*

$$\omega_K(r) = \max_{x,y \in K, ||x-y|| \leq r} |f(x) - f(y)|. \quad (12)$$

Proof. Refer to [48]. □

Now, we prove **Theorem 2** through the lemmas above.

Proof. First, $T(x)$ is a Lipschitz continuous function, so

$$|T(x) - T(y)| \leq L|x - y|, \quad (13)$$

for $x, y \in K$, where L is a constant. Then we have

$$\begin{aligned} \omega_K(r) &= \max_{x,y \in K, ||x-y|| \leq r} |T(x) - T(y)| \\ &\leq \max_{x,y \in K, ||x-y|| \leq r} L|x - y| \qquad \text{Since Eq.(13)} \\ &\leq Lr. \end{aligned}$$

Let $\omega_K(r) \leq Lr = \epsilon/\text{Vol}(K)$, and then we have

$$r = \frac{\epsilon}{\text{Vol}(K) \cdot L}. \quad (14)$$

When $r = \epsilon/(\text{Vol}(K) \cdot L)$, then for any $\epsilon \in (\epsilon_0, 1)$,

$$O(1/r^d) = O((\frac{L}{\epsilon})^d) < O((\frac{L}{\epsilon_0})^d) = C(\frac{L}{\epsilon_0})^d, \quad (15)$$

where C is a constant. Therefore, according to Lemma 3, $\forall \epsilon \in (\epsilon_0, 1)$, there exist a ResNet $R_{\text{short}}(x)$ with width not greater than d and the depth of at most $C(\frac{L}{\epsilon_0})^d$ such that $\int_K |T(x) - R_{\text{short}}(x)| dx \leq \epsilon/2$. When the depth of $R_{\text{full}}(x, \theta_{\text{full}})$ is greater than $C(\frac{L}{\epsilon_0})^d$, we can use Lemma 2 (extension strategy) to construct a function $R_{\text{long}}(x)$ such that

$$\text{dep}(R_{\text{long}}) = \text{dep}(R_{\text{full}}), \quad (16)$$

and the width of R_{long} is not greater than d. Also, for any $x \in K$, $R_{\text{long}}(x) = R_{\text{short}}(x)$. So we have

$$\int_K |T(x) - R_{\text{long}}(x)| dx = \int_K |T(x) - R_{\text{short}}(x)| dx \leq \epsilon/2. \quad (17)$$

Then

$$\int_K |R_{\text{full}}(x, \theta_{\text{full}}^0) - R_{\text{long}}(x)| dx \leq \int_K |T(x) - R_{\text{long}}(x)| dx \quad (18)$$

$$+ \int_K |R_{\text{full}}(x, \theta_{\text{full}}^0) - T(x)| dx \quad (19)$$

$$\leq \epsilon/2 + \epsilon_0/2 \leq \epsilon \quad (20)$$

Note that $\text{dep}(R_{\text{long}}) = \text{dep}(R_{\text{full}})$. Also, R_{long} is a ResNet with width not larger than d while the width of R_{full} is greater than d. Therefore, R_{long} is a subnetwork of R_{full} and satisfies the inequality (20).

□

APPENDIX
DIFFERENT TYPES OF SAMS

In this part, we review the SAMs used in our paper, *i.e.*, SE [2], SGE [4] and DIA [5]. We follow some notations of Section II. Let x_ℓ be the input of the ℓ^{th} block, $f_\ell(\cdot)$ be the residual mapping, and $M(\cdot; W_\ell)$ be the SAM in the ℓ^{th} block with the parameters W_ℓ. The attention is formulated as $M(f_\ell(x_\ell); W_\ell)$. We denote $f_\ell(x_\ell)$ as $X^{(\ell)}$ of size $C \times H \times W$, where C, H and W denote channel, height and width, respectively. For simplicity, we denote $X_{chw}^\ell = X^\ell[c, h, w]$ as the value of pixel (h, w) at the channel c and $X_c^\ell = X^\ell[c, :, :]$ as the tensor at the channel c.

SE Module. SE module utilizes average pooling to extract the features and processes the extracted features by a one-hidden-layer fully connected network.

First, the SE module squeezes the information of channels by the average pooling,

$$m_c^\ell = \text{AVG}(X_c^\ell) = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} X_{chw}^\ell, \quad (21)$$

where $c = 1, \cdots, C$. Then, an one-hidden-layer fully connected network $\text{FC}(\cdot; W_\ell)$ with ReLU activation is used to fuse the information of all the channels and here W_ℓ is the parameter. The hidden layer node size is $C//r$, where "//" is exact division and "r" denotes reduction rate. The reduction rate is 16 in our experiments. Finally, a sigmoid function (*i.e.*, $\text{sig}(z) = 1/(1+e^{-z})$) is applied to the processed features and we get the attention as follows,

$$[\delta_1; \cdots; \delta_C] = \text{sig}(\text{FC}([m_1^\ell; \cdots; m_C^\ell]; W_\ell)). \quad (22)$$

DIA Module. DIA module integrates the block-wise information by an LSTM (Long Short-Term Memory). Let m_c^ℓ be the output of average pooling as Eq.21. Then m_c^ℓ is passed to LSTM along with a hidden state vector $h_{\ell-1}$ and a cell state vector $c_{\ell-1}$, where h_0 and c_0 are initialized as zero vectors. The LSTM generates h_ℓ and c_ℓ at the ℓ^{th} block, *i.e.,*

$$(h_\ell, c_\ell) = \text{LSTM}([m_1^\ell; \cdots; m_C^\ell], h_{\ell-1}, c_{\ell-1}; W), \quad (23)$$

where W is the trainable parameter of the LSTM. The hidden state vector h_t is used as attention to recalibrate feature maps. The reduction ratio within LSTM introduced in [5] is 4 for CIFAR100 or 20 for ImageNet2012.

SGE Module. SGE divides the feature maps into different groups and then utilizes the global information from the group to recalibrate its features. Let G be the number of groups and then each group has $C//G$ feature maps. Denote Y^ℓ of size $(C//G) \times H \times W$ as a group of feature maps within X^ℓ. The extracted feature for the group Y^ℓ is

$$g_c^\ell = \text{AVG}(Y_c^\ell) = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} Y_{chw}^\ell. \quad (24)$$

Let g be $[g_1^\ell; \cdots; g_{C//G}^\ell]$. The importance coefficient for each pixel (h, w) is defined as

$$p_{hw} = g \cdot Y[:, h, w], \tag{25}$$

where $\cdot$ is dot product. Then p_{hw} is normalized by

$$\hat{p}_{hw} = \frac{p_{hw} - \mu}{\sigma + \epsilon}, \tag{26}$$

where the mean μ and variance σ^2 are defined by

$$\mu = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} p_{hw}, \quad \sigma^2 = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(p_{hw} - \mu)^2. \tag{27}$$

An additional pair of parameters (γ, β) are introduced for the group Y^ℓ to rescale and shift the normalized features, and SGE modules get the attention for $Y[:, h, w]$ as follows,

$$\text{sig}(\gamma \hat{p}_{hw} + \beta). \tag{28}$$

G is 4 for CIFAR100 and 64 for ImageNet2012 experiments.

APPENDIX
TRAINING DETAILS FOR SUPERNET

In Alg.1, we have presented the training strategy for supernet briefly. We show this process in an intuitive way in Fig. 8. First, for each step t, we can sample one connection scheme from a $[Bernoulli(\beta)]^m$ distribution. Next, based on this sampled connection scheme, we can obtain a subnetwork from supernet. Then, we train this subnetwork on the training set D_{train}.

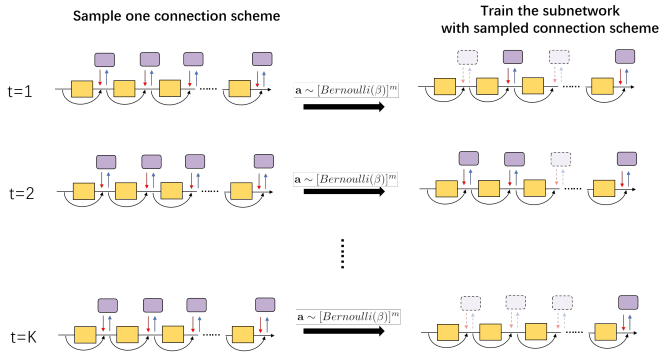

Fig. 8. Procedure of training a supernet.

APPENDIX
TRAINING DETAILS FOR CONTROLLER

In this part, we provide the training details for the controller. The training process is shown in Fig. 9. The reward function of the connection scheme consists of three parts, *i.e.*, sparsity reward, validation reward, and curiosity bonus. Besides, we supplement some details of Alg. 1.

Sparsity Reward g_{spa}. One of our goals is to accelerate the inference of the Full-SA network. To achieve it, we complement a sparsity reward g_{spa} to encourage the controller to generate the schemes with fewer connections between SAMs and backbone. We define g_{spa} by

$$g_{\text{spa}} = 1 - \frac{\|\mathbf{a}\|_0}{m}, \tag{29}$$

where $\|\cdot\|_0$ is a zero norm that counts the number of non-zero entities, and m is the number of blocks.

Validation Reward g_{val}. Another goal is to find the schemes with which the networks can maintain the original accuracy. Hence, we use the validation accuracy of the subnetwork $\Omega(\mathbf{x}|\mathbf{a})$ sampled from the supernet as a reward, which depicts the performance of its structure. The accuracy of $\Omega(\mathbf{x}|\mathbf{a})$ on D_{val} is denoted as g_{val}. In fact, it is popular to use validation accuracy of a candidate network as a reward signal in NAS [15], [20], [26], [29], [47]. Furthermore, it has been empirically proven that the validation performance of the subnetworks sampled from a supernet can be positively correlated to their stand-alone performance [49]. We evaluate the correlation between the validation accuracy of subnetworks sampled from a supernet and their stand-alone performance on CIFAR100 with ResNet and SE module over 42 samples and obtain the Pearson coefficient is 0.71, which again confirms the strong correlation as shown in the previous works.

Curiosity Bonus g_{rnd}. To encourage the controller to explore more potentially useful connection schemes, we add the Random Network Distillation (RND) curiosity bonus [23] in our reward. Two extra networks with input $\mathbf{a}$ are involved in the RND process, including a target network $\sigma_1(\cdot)$ and a predictor network $\sigma_2(\cdot;\phi)$, where ϕ is the parameter set. The parameters of $\sigma_1(\cdot)$ are randomly initialized and fixed after initialization, while $\sigma_2(\cdot;\phi)$ is trained with the connection schemes collected by the controller.

The basic idea of RND is to minimize the difference between the outputs of these two networks, which is denoted by term $\sigma_\phi(\cdot) = \|\sigma_1(\cdot) - \sigma_2(\cdot;\phi)\|_2^2$, over the seen connection schemes. If the controller generates a new scheme $\mathbf{a}$, $\sigma_\phi(\mathbf{a})$ is expected to be larger because the predictor $\sigma_2(\cdot;\phi)$ never trains on scheme $\mathbf{a}$. Then, we denote the term $\|\sigma_1(\mathbf{a}) - \sigma_2(\mathbf{a};\phi)\|_2^2$ as g_{rnd}, which is used as curiosity bonus to reward the controller for exploring a new scheme. Besides, in Fig. 5, we empirically show that RND bonus mitigates the fast convergence of early training iterations, leading to exploration for more schemes.

To sum up, our reward $G(\mathbf{a})$ becomes

$$G(\mathbf{a}) = \lambda_1 \cdot g_{\text{spa}} + \lambda_2 \cdot g_{\text{val}} + \lambda_3 \cdot g_{\text{rnd}}, \tag{30}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the coefficients for each reward.

Data Reuse. To improve the utilization efficiency of sampled connection schemes and speed up the training of the controller, we incorporate Proximal Policy Optimization (PPO) [50] in our method. As shown in Alg. 1, after the update of parameter θ and ϕ, we put the tuple $(\mathbf{p}_\theta,\ \mathbf{a},\ G(\mathbf{a}))$ into a buffer. At the later step, we retrieve some used connection schemes and update θ as follows:

$$\begin{aligned} \kappa &= \mathbb{E}_{\mathbf{a}\sim\mathbf{p}_{\theta_{old}}}\left[G(\mathbf{a})\sum_{i=1}^{m}\frac{\hat{p}_\theta^i}{\hat{p}_{\theta_{old}}^i}\nabla_\theta \log \hat{p}_\theta^i\right], \\ \theta &\leftarrow \theta + \eta \cdot \kappa, \end{aligned} \tag{31}$$

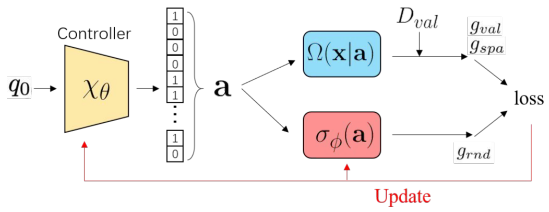

Fig. 9. The illustration of our policy-gradient-based method to search an optimal scheme.

where η is the earning rate and the θ_{old} denotes the θ sampled from buffer. Note that we do not update with PPO until the controller is updated for h times.

Finally, we give hyper-parameter settings for training a controller on different datasets.

CIFAR100. We optimize the controller for 1000 iterations with momentum SGD. The learning rate is set to be 5×10^{-2}. The time step h to apply PPO is 10.

ImageNet2012. We optimize the controller for 300 iterations with momentum SGD. The learning rate is set to be 5×10^{-2}. The time step h to apply PPO is 10.

APPENDIX
TRAINING DETAILS FOR STAND-ALONE PERFORMANCE

In this part, we introduce the parameter setting for the model trained from scratch. In our experiments, we use cross-entropy loss and optimize the model by SGD with momentum 0.9 and initial learning rate 0.1. The weight decay is set to be 10^{-4}. The results for all search methods reported are the best out of three candidates with the highest reward (lowest validation loss for DARTS) in one search.

CIFAR100. When ResNet164 is used, the model is trained for 164 epochs with the learning rate dropped by 0.1 at 81, 122 epochs. When ResNet38 is used, the model is trained for 100 epochs with the learning rate following cosine learning rate decay. In order to mitigate the over-fitting problems faced by the deep networks, ResNet164 is trained with random flipping and cropping. ResNet38 is trained with random flipping.

ImageNet2012. We use the ResNet50 backbone for ImageNet experiments. The network is trained for 120 epochs with the learning rate dropped by 0.1 at every 30 epochs.

REFERENCES

[1] S. Liang, Z. Huang, M. Liang, and H. Yang, "Instance enhancement batch normalization: An adaptive regulator of batch noise." in *AAAI*, 2020, pp. 4819–4827.

[2] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[3] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[4] X. Li, X. Hu, and J. Yang, "Spatial group-wise enhance: Improving semantic feature learning in convolutional networks," *arXiv preprint arXiv:1905.09646*, 2019.

[5] Z. Huang, S. Liang, M. Liang, and H. Yang, "Dianet: Dense-and-implicit attention network." in *AAAI*, 2020, pp. 4206–4214.

[6] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[7] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[8] 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.

[9] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[10] H. Lee, H.-E. Kim, and H. Nam, "Srm: A style-based recalibration module for convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1854–1862.

[11] W. He, M. Wu, M. Liang, and S.-K. Lam, "Cap: Context-aware pruning for semantic segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 960–969.

[12] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," *arXiv preprint arXiv:1810.05270*, 2018.

[13] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.

[14] P. Vidnerová and R. Neruda, *Multi-objective Evolution for Deep Neural Network Architecture Search*, 2020.

[15] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, "Single path one-shot neural architecture search with uniform sampling," in *European Conference on Computer Vision*. Springer, 2020, pp. 544–560.

[16] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*, 2018.

[17] Z. Huang, W. Shao, X. Wang, L. Lin, and P. Luo, "Rethinking the pruning criteria for convolutional neural network," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=HL_4vjPTdtp

[18] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4340–4349.

[19] D. Wang, C. Gong, M. Li, Q. Liu, and V. Chandra, "Alphanet: Improved training of supernets with alpha-divergence," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 760–10 771. [Online]. Available: https://proceedings.mlr.press/v139/wang21i.html

[20] S. You, T. Huang, M. Yang, F. Wang, C. Qian, and C. Zhang, "Greedynas: Towards fast one-shot nas with greedy supernet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1999–2008.

[21] X. Chu, B. Zhang, R. Xu, and J. Li, "Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search," *arXiv preprint arXiv:1907.01845*, 2019.

[22] S. Liang, Y. Khoo, and H. Yang, "Drop-activation: Implicit parameter reduction and harmonious regularization," *Communications on Applied Mathematics and Computation*, vol. 3, no. 2, pp. 293–311, 2021.

[23] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=H1lJJnR5Ym

[24] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[26] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *International Conference on Machine Learning*, 2018, pp. 4095–4104.

[27] L. et al., "Darts: Differentiable architecture search," 2018.

[28] Z. Huang, W. Shao, X. Wang, and P. Luo, "Convolution-weight-distribution assumption: Rethinking the criteria of channel pruning," *arXiv preprint arXiv:2004.11627*, 2020.

[29] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[30] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via

gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[31] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 589–597.

[32] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[33] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091–1100.

[34] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, "Crowd counting using scale-aware attention networks," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1280–1288.

[35] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[36] B. Ma, Y. Zhao, Y. Yang, X. Zhang, X. Dong, D. Zeng, S. Ma, and S. Li, "Mri image synthesis with dual discriminator adversarial learning and difficulty-aware attention mechanism for hippocampal subfields segmentation," *Computerized Medical Imaging and Graphics*, vol. 86, p. 101800, 2020.

[37] Z. Fan, X. Zhang, J. A. Gasienica, J. Potts, S. Ruan, W. Thorstad, H. Gay, X. Wang, and H. Li, "A novel adversarial learning strategy for medical image classification," *arXiv preprint arXiv:2206.11501*, 2022.

[38] L. Liu, J. Chen, H. Wu, T. Chen, G. Li, and L. Lin, "Efficient crowd counting via structured knowledge transfer," in *ACM International Conference on Multimedia*, 2020.

[39] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[40] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2019.

[41] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, and M. Rastegari, "What's hidden in a randomly weighted neural network?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[42] T. Chen, J. Frankle, S. Chang, S. Liu, Y. Zhang, Z. Wang, and M. Carbin, "The lottery ticket hypothesis for pre-trained bert networks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 15 834–15 846. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/b6af2c9703f203a2794be03d443af2e3-Paper.pdf

[43] X. Chen, Y. Cheng, S. Wang, Z. Gan, Z. Wang, and J. Liu, "Early{bert}: Efficient {bert} training via early-bird lottery tickets," 2021. [Online]. Available: https://openreview.net/forum?id=I-VfjSBzi36

[44] E. Malach, G. Yehudai, S. Shalev-Schwartz, and O. Shamir, "Proving the lottery ticket hypothesis: Pruning is all you need," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 6682–6691. [Online]. Available: https://proceedings.mlr.press/v119/malach20a.html

[45] L. Orseau, M. Hutter, and O. Rivasplata, "Logarithmic pruning is all you need," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 2925–2934. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/1e9491470749d5b0e361ce4f0b24d037-Paper.pdf

[46] A. Pensia, S. Rajput, A. Nagle, H. Vishwakarma, and D. Papailiopoulos, "Optimal lottery tickets via subsetsum: Logarithmic over-parameterization is sufficient," *Advances in neural information processing systems*, 2020.

[47] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[48] H. Lin and S. Jegelka, "Resnet with one-neuron hidden layers is a universal approximator," *Advances in Neural Information Processing Systems*, vol. 31, pp. 6169–6178, 2018.

[49] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *International Conference on Machine Learning*, 2018, pp. 550–559.

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