

AdaAugment: A Tuning-Free and Adaptive Approach to Enhance Data Augmentation

Suorong Yang, Peijia Li[✉], Xin Xiong, Furao Shen[✉], *Member, IEEE*, and Jian Zhao[✉], *Senior Member, IEEE*

Abstract—Data augmentation (DA) is widely employed to improve the generalization performance of deep models. However, most existing DA methods employ augmentation operations with fixed or random magnitudes throughout the training process. While this fosters data diversity, it can also inevitably introduce uncontrolled variability in augmented data, which could potentially cause misalignment with the evolving training status of the target models. Both theoretical and empirical findings suggest that this misalignment increases the risks of both underfitting and overfitting. To address these limitations, we propose AdaAugment, an innovative and tuning-free adaptive augmentation method that leverages reinforcement learning to dynamically and adaptively adjust augmentation magnitudes for individual training samples based on real-time feedback from the target network. Specifically, AdaAugment features a dual-model architecture consisting of a policy network and a target network, which are jointly optimized to adapt augmentation magnitudes in accordance with the model’s training progress effectively. The policy network optimizes the variability within the augmented data, while the target network utilizes the adaptively augmented samples for training. These two networks are jointly optimized and mutually reinforce each other. Extensive experiments across benchmark datasets and deep architectures demonstrate that AdaAugment consistently outperforms other state-of-the-art DA methods in effectiveness while maintaining remarkable efficiency. Code is available at <https://github.com/Jackbrocp/AdaAugment>.

Index Terms—Data augmentation, generalization, deep learning, computer vision, automated data augmentation.

I. INTRODUCTION

RECENTLY, deep neural networks (DNNs) have witnessed remarkable advancements. Despite these advancements, achieving satisfactory performance often necessitates larger network architectures and extensive training data. This is particularly challenging as labeled data are sometimes unavailable and expensive to collect [1], [2], [3]. Thus, the ability of

deep learning models to generalize beyond the finite training set remains one of the fundamental problems in deep learning [4], [5]. In this context, data augmentation (DA) emerges as a critical and highly effective technique for enhancing DNN generalization performance [6], [7], [8], leading to the development of more generalized and robust models [9], [10], [11], [12]. Indeed, training large deep models to achieve state-of-the-art (SOTA) performance typically employs strong DA techniques.

However, it is important to note that existing DA methods primarily rely on the utilization of totally random or manually devised augmentation magnitudes during training, potentially leading to suboptimal training scenarios [13]. For instance, DA methods based on information deletion or fusion [14], [15], [16], [17], [18], [19], [20] typically erase or mix random sub-regions in images to generate augmented samples, resulting in random augmentation strengths. Automatic DA approaches [21], [22], [23], [24], [25] often incur significant computational overhead to optimize augmentation strategies for each specific dataset prior to target model training. During training, the magnitudes of the applied augmentations are predetermined without any adaptation throughout the entire training process. Moreover, most adaptive DA methods [25], [26], [27], [28] select augmentation operations with random or predefined magnitudes during online training, primarily aiming to alleviate overfitting risks. Consequently, existing DA methods may fail to capture the unique characteristics of different datasets and the entire evolving state of network training, potentially limiting their effectiveness.

Using random augmentation magnitudes or random operations with fixed magnitudes introduces inherent randomness, which can increase data diversity yet introduce uncontrolled variability into the augmented data. Such randomness in data variability may not optimally align with the evolving training status of deep models, thereby introducing severe side effects to the training process. For instance, during the initial training stages, when models generally exhibit weaker generalization capabilities, substantial data variability can induce noises and distributional shifts, potentially leading to underfitting phenomena [29]. In contrast, in the later training stages, limited data variability may elevate the risk of overfitting [16]. Thus, this misalignment in current DA methodologies increases the risks of both underfitting and overfitting, which can ultimately have adverse effects on the generalization performance of models [9], [30]. *This is a challenge often overlooked in the field.*

Received 2 June 2024; revised 4 December 2024 and 4 June 2025; accepted 12 July 2025. Date of publication 30 July 2025; date of current version 4 August 2025. This work was supported in part by Science and Technology Innovation (STI) 2030-Major Projects of China under Grant 2021ZD0201300, in part by the Fundamental Research Funds for the Central Universities under Grant 2024300394, and in part by the National Natural Science Foundation of China under Grant 62276127. The associate editor coordinating the review of this article and approving it for publication was Dr. Jacinto C. Nascimento. (*Corresponding authors: Suorong Yang; Furao Shen.*)

Suorong Yang is with the State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China (e-mail: sryang@smail.nju.edu.cn).

Peijia Li, Xin Xiong, and Furao Shen are with the State Key Laboratory for Novel Software Technology and the School of Artificial Intelligence, Nanjing University, Nanjing 210023, China (e-mail: lipj@smail.nju.edu.cn; xiongxin@smail.nju.edu.cn; frshen@nju.edu.cn).

Jian Zhao is with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: jianzhao@nju.edu.cn).

Digital Object Identifier 10.1109/TIP.2025.3592538

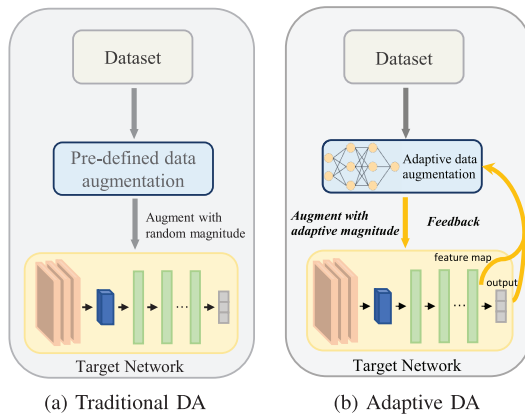


Fig. 1. Comparison between traditional DA methods and our adaptive DA mechanism. (a) The typical methods employ augmentation operations with random magnitudes and may not align with the training status of deep models. In contrast, in (b), our AdaAugment adaptively determines augmentation magnitudes for each training sample based on the model's real-time feedback.

To tackle these challenges, it is crucial to adopt an adaptive modulation of augmented data variability based on real-time feedback from the target models. As illustrated in Figure 1, we present a comparison between traditional DA and adaptive DA mechanisms. Unlike conventional approaches that rely on random or predefined augmentation magnitudes, this adaptive DA mechanism dynamically adjusts the magnitudes of DA operations based on real-time feedback from the target network. Moreover, different from most existing adaptive DA methods, this adaptive DA mechanism **emphasizes optimizing augmentation strengths rather than the specifics of the operations themselves**. The adaptive strategy significantly reduces the search space and offers an effective countermeasure to directly adjust the variability of the augmented data regardless of the specifics of the augmentation policies. This brings several notable advantages: effectively mitigating the aforementioned challenges, unleashing the full potential of DA, and enhancing the generalization capabilities of deep models. A straightforward approach to achieving adaptive DA is to create a formal measure that reflects the real-time learning status of each training sample. This measure can then be used to determine the appropriate level of augmentation strengths, thereby adapting model training status. Nevertheless, theoretical analyses have demonstrated that determining the model status, e.g., underfitting or overfitting risks of a learning algorithm, is undecidable [31], [32], [33]. Consequently, formulating a definitive measure of learning status for training samples remains a formidable challenge.

To address the aforementioned limitations and challenges, in this work, we propose **AdaAugment**, a novel and tuning-free **Adaptive data Augmentation** method for image classification. Instead of relying on any hand-crafted measures, AdaAugment leverages a reinforcement learning algorithm to adaptively determine specific augmentation magnitudes for each training sample. Central to AdaAugment is a dual-model architecture: the policy network and the target network. The policy network learns the policy that determines the magnitudes of augmentation operations for each training sample based on the real-time feedback from the target network during

training. The target network simultaneously utilizes these adaptively augmented samples for training. Both networks are jointly optimized, eliminating the need for separate re-training of the target network and enhancing the practical applicability of our approach across diverse datasets. The learned policy adaptively adjusts the variability within augmented data to align with the learning status of the target model, thereby optimizing the introduced variability into the augmented data. Specifically, in the training of the policy network, we estimate the risks of underfitting and overfitting by deriving losses from fully augmented and non-augmented data, respectively. These two losses are then compared with those derived from data adaptively augmented by AdaAugment, serving as reward signals. In this way, **AdaAugment is capable of adjusting the emphasis to align with model training status, effectively achieving model- and data-adaptive augmentation without requiring any prior knowledge about specific datasets**. Extensive experimental results across various benchmark datasets, including coarse-grained classification datasets CIFAR-10/100 [34], Tiny-ImageNet [35], and ImageNet-1k [36], long-tail classification datasets ImageNet-LT and Places-LT [37], and several fine-grained datasets [38], [39], [40], [41] demonstrate the superior effectiveness of AdaAugment in enhancing model generalization performance. Furthermore, complexity analyses of AdaAugment verify that it merely imposes minimal parameters and computational overhead, highlighting its competitive efficiency. Thus, AdaAugment achieves a commendable balance between effectiveness and efficiency, yielding high performance without introducing excessive computational complexity. For instance, AdaAugment can achieve more than 1% improvements over other state-of-the-art DA methods when training on Tiny-ImageNet without any architectural modifications or additional regularization.

In summary, we highlight our contributions as follows:

- 1) We propose AdaAugment, an innovative and tuning-free adaptive DA approach that leverages reinforcement learning techniques to dynamically adapt augmentation magnitudes for individual training samples based on real-time feedback from the target network.
- 2) AdaAugment features a dual model architecture that jointly optimizes a policy network and a target network. This facilitates policy formulation and enhances task performance concurrently and effectively.
- 3) AdaAugment employs both model- and data-adaptive DA operations to dynamically adapt model training progress.
- 4) Extensive experiments conducted on multiple benchmark datasets and deep architectures demonstrate that AdaAugment outperforms existing SOTA DA methods with competitive training efficiency. This creates a strong baseline of data augmentation for future research.

II. RELATED WORK

A. Data Augmentation

DA has played a crucial role in enhancing the generalization capabilities of deep neural networks. Existing DA methods

could be broadly divided into two categories: methods based on information deletion or fusion [14], [15], [16], [17], [18], [19], [42], [43] and automatic DA methods [21], [22], [23], [24], [25], [26], [27], [28], [44]. Typically, these methods have primarily relied on augmentations characterized by random or predefined magnitudes to introduce diversity into the training data.

1) *Methods Based on Information Deletion or Fusion:*

Among these methods, Cutout [14] is one of the most widely used techniques that randomly masks out one or more square regions within images. Random Erasing [43] randomly selects a rectangular region within an image and erases its pixels with random values. Similarly, Hide-and-Seek (HaS) [45] randomly hides patches in a training image, improving object localization accuracy and the generalization ability of deep models. GridMask [15] employs structured dropping regions within the input images. Since these methods may easily introduce noise and ambiguity into the augmented data, AdvMask [16] identifies classification-discriminative information in images and structurally drops some sub-regions containing critical points for augmentation. Additionally, Mixup [17] and CutMix [18] synthesize augmented data by mixing random information from two or more images. However, these DA approaches primarily focus on data transformations and often overlook the training status of models. This oversight makes it challenging to alleviate overfitting risks accordingly through online adjustments to augmentation strengths.

2) *Automatic DA Methods:* This group of studies can be broadly divided into fixed augmentation policies and online augmentation policies. Methods based on fixed augmentation policies, such as AutoAugment [21], Fast-AutoAugment [46], Faster-AutoAugment [47], and RandAugment [22] leverage reinforcement learning or grid search to search existing policies for the optimal combination of DA operations on different image datasets in an offline manner. Similarly, Adversarial AutoAugment [48] utilizes a fixed augmentation space and rewards the policies that yield the lowest accuracy, causing the policy distribution to shift towards progressively stronger augmentations throughout training. TrivialAugment [23] employs the same augmentation space obtained by these automatic DA methods and applies a single augmentation operation to each image during training. MetaAugment [49] leverages a static augmentation space, and its augmentation policy network outputs weights for augmented data loss. However, the magnitudes of augmentation policies employed in these methods remain fixed or are randomly sampled during online training, leading to the uncontrollable extent of data transformations [50].

On the other hand, methods based on online policies focus on adaptively selecting augmentation policies during training through various optimization processes [26]. Meanwhile, they primarily aim to alleviate overfitting risks. For instance, the work [26] proposes an adaptive DA method that focuses on identifying small transformations that yield maximal classification loss on the augmented data. SelectAugment [24] employs hierarchical reinforcement learning to acquire online policies that determine the portion of the augmented data

and whether each individual sample should be augmented. Since the specific augmentation transformation applied to each sample is AutoAugment, Mixup, or CutMix, the DA strengths employed are still uncontrollable. The work [51] leverages impact modeling via the influence function to learn differentiable DA transformation. The work [52] randomly determines the types and magnitudes of DA operations for batch-level data and updates DA's parameters along the gradient direction of the target network loss. The work [53] proposes AdDA for contrastive learning, which allows the network to adjust the augmentation compositions and achieve more generalizable representations. KeepAugment [30] detects and preserves the salient regions of the images during augmentation. MADAug [50] trains a model-adaptive policy through a bi-level optimization scheme, minimizing a validation set loss of a model trained using the policy-produced data augmentations. AdaAug [27] learns class-dependent and potentially instance-dependent augmentations via a differentiable workflow. Meanwhile, MADAug and AdaAug search not only the applied magnitudes but also the probability of augmentation and specific augmentation policies. TeachAugment [25] transforms data so that they are adversarial to the target model. More recently, the work [9] evaluates DA's efficacy using similarity and diversity measures, revealing their varying importance across datasets and suggesting the merit of adaptively adjusting augmentation strengths for an optimal balance. Different from existing DA approaches, our proposed AdaAugment merely optimizes the real-value augmentation magnitudes regardless of specific augmentation policies. This brings **several notable advantages**: the introduced data diversity can be directly optimized regardless of specific augmentation operations; optimizing real-value magnitudes significantly enhances optimization efficiency; and AdaAugment is capable of adapting DA operations with model training status.

B. Reinforcement Learning

Reinforcement learning (RL) entails learning a sequence of actions within an interactive environment by trial and error that maximizes the expected reward [54], [55]. RL obtains a wide range of applications, such as autonomous driving [56] and recommender systems [57], etc. In the realm of RL algorithms, there are two fundamental categories: value optimization and policy optimization methods. Value optimization methods primarily focus on the estimation of an optimal value function, which subsequently serves as a basis for deriving the optimal policy [58]. Conversely, policy optimization methods estimate the optimal behavior policy without directly estimating the value functions [59]. Moreover, the actor-critic framework, widely adopted in RL, combines the strengths of both value-based and policy-based RL methods [60], [61], [62]. This framework comprises two components: the actor, responsible for learning the policy function, and the critic, tasked with evaluating the actor's chosen actions by estimating the value function. This dual mechanism ensures more stable and efficient learning [63], such as Advantage Actor-Critic (A2C) [64].

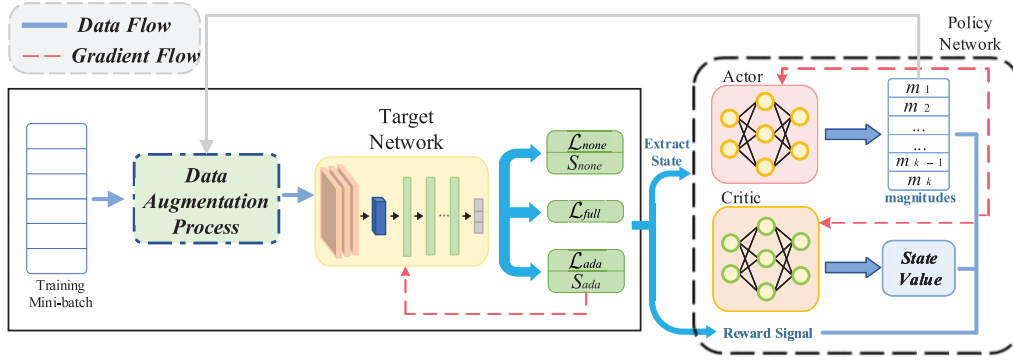


Fig. 2. The dual-model framework of the proposed AdaAugment.

III. ADAUGMENT

A. Overview

The primary objective of AdaAugment is to mitigate both underfitting and overfitting risks through adaptively adjusting DA strengths during the training phase. Specifically, it accelerates model convergence in early training stages and alleviates overfitting risks in later training stages. This adaptive adjustment can be formalized as a decision-making problem on a per-sample basis, thereby eliminating the need for manually crafted metrics. Figure 2 illustrates the dual-model framework of AdaAugment: While training the target network, a policy network is simultaneously introduced to dynamically optimize the magnitudes of DA operations within the augmentation process. This dual-model framework facilitates the joint optimization of both networks, eliminating the need for separate re-training of the target network and enabling real-time adjustments of augmentation strengths based on the training progress of the target network. Specifically, the policy network optimizes magnitudes at the sample level in the current training epoch. Subsequently, in the next epoch, the data augmentation process applies augmentation operations to the training samples with these corresponding magnitudes. The adaptively augmented samples are then employed to optimize the target network parameters.

B. Preliminary

An RL task can be formalized as a Markov Decision Process (MDP), consisting of the following components: a state space \mathcal{S} , an action space \mathcal{A} , a transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ representing the probability of transitioning from one state to another by taking actions, a reward function \mathcal{R} , a discount factor $\gamma \in [0, 1]$, and a time step T . Given a state $s \in \mathcal{S}$, an RL agent determines an action $a \in \mathcal{A}$ with the policy $\pi(a | s)$. With these elements, the objective of an RL task is to find an optimal policy, denoted as π^* , that maximizes the expected cumulative reward within the given MDP framework.

Suppose that the training dataset \mathcal{D} consists of N training samples, each of the form $(x, y) \in \mathcal{D}$. Here, x denotes the original data, and y is a k -dimensional vector of zeros and ones indicating the true label of x , where k is the total number of classes. The augmentation operation is defined as $e(m, x)$, where e is randomly drawn from an augmentation

TABLE I
THE AUGMENTATION SPACE \mathcal{E}

Transformation	S_{Max}	Symmetric
identity	-	-
auto contrast	-	-
equalize	-	-
color	1.9	-
contrast	1.9	-
brightness	1.9	-
sharpness	1.9	-
rotation	30°	±
translate _x	10	±
translate _y	10	±
shear _x	0.3	±
shear _y	0.3	±
solarize	256	-
posterize	4	-

space \mathcal{E} and m represents the corresponding magnitude of e . Although e is randomly selected from \mathcal{E} , the applied magnitudes are adaptively determined by AdaAugment. This effectively adjusts the augmented data variability, achieving a superior balance between the diversity and consistency of the training data, leading to improved model generalization.

We provide the specific details of \mathcal{E} in Table I, which contains widely used image operations and their corresponding maximal magnitudes. Notably, the augmentation space \mathcal{E} closely follows the setting from [21], [22], [23], and [46]. Unlike previous works, our approach adaptively determines the magnitude in the augmentation space \mathcal{E} rather than assigning a predefined value before training. In this way, the applied magnitudes are determined by $S_{Max} \times m$, where S_{Max} is the maximum allowable strength of the corresponding transformations. Moreover, for symmetric transformations, the symmetric direction is randomly selected.

C. State Design

Since the objective of RL is to dynamically determine the appropriate magnitudes for each sample, the state $s \in \mathcal{S}$ should consider three factors: the inherent difficulty associated with each sample, the current training status of the models (e.g., feature extraction capabilities), and the intensity of augmentation operations w.r.t. the former two factors. This

multifaceted consideration of state variables is essential for effective RL. Notably, the feature map plays a vital role in this process by providing feedback from the model [65], [66], encapsulating both the inherent difficulty of the sample and the model’s real-time feature extraction capability. To illustrate this, as shown in Figure 2, the state vector s in AdaAugment encodes the feature maps of both the non-augmented sample \mathbf{x} and adaptively augmented sample $\tilde{\mathbf{x}}$, denoted as s_{none} and s_{ada} , respectively. The feature maps are obtained from the output of the final linear layer before the classification head in the target network. Here, s_{none} captures the inherent difficulty of sample \mathbf{x} w.r.t. the current target network training status. Meanwhile, s_{ada} encodes the augmentation operation strength applied to $\tilde{\mathbf{x}}$ w.r.t. both the real-time target network status and the inherent sample difficulty. By utilizing this information, AdaAugment effectively aligns data variability with the evolving training status of the target networks.

D. Action Design

The policy determines the augmentation magnitudes \mathbf{m} for the augmented data. Despite the randomness in the composition of each mini-batch during training, the magnitudes \mathbf{m} operate on a per-sample basis, corresponding to each training sample. For simplicity, we denote the magnitudes of the current mini-batch data as \mathbf{m} , where the dimension of \mathbf{m} equals the batch size, and each $m \in \mathbf{m}$ is strictly bounded within the interval $[0, 1]$. When $m = 0$, no augmentation is applied, while $m = 1$ indicates the maximum magnitudes for the corresponding augmentation operations. Thus, magnitudes closer to 0 yield samples that are more similar to the original data, while magnitudes closer to 1 produce more diverse data. The adaptively augmented sample $\tilde{\mathbf{x}}$ can be obtained by:

$$\tilde{\mathbf{x}} = e(\mathbf{m}, \mathbf{x}), \quad (1)$$

where e is a random augmentation operation and m is determined based on the action policy. Eq. (1) enables us to optimize the similarity-diversity preference of the augmented data. Thus, the policy network does not know the augmentation types and directions but just optimizes the magnitudes. More importantly, this adjustment on the similarity-diversity preference reflects AdaAugment’s effectiveness in accelerating model convergence in the early training stages and alleviating overfitting risks at later training stages.

E. Reward Function

Consider a target classification model f_θ , parameterized by θ , an input sample $\mathbf{x} \in \mathbb{R}^n$, and $f_\theta(\mathbf{x})$ denotes the network output. Let \mathcal{L} denote the cross-entropy loss, and $\mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y})$ represent the loss item for the original sample (\mathbf{x}, \mathbf{y}) . Our proposed method aims to promote model fitting in the early training stages and alleviate overfitting at later training stages by controlling the magnitudes of data augmentation operations based on feedback from the target model. To this end, we define three loss terms based on our augmentation strategy. Firstly, $\mathcal{L}_{full}(f_\theta(\mathbf{x}^+), \mathbf{y})$ represents the loss for the sample \mathbf{x} with the maximal augmentation magnitude, i.e., $\mathbf{x}^+ = e(\mathbf{m} = 1, \mathbf{x})$. Secondly, $\mathcal{L}_{none}(f_\theta(\mathbf{x}^-), \mathbf{y})$ denotes the

TABLE II
NETWORK ARCHITECTURE OF A2C

	index	Layer	Dimension
Actor	1	Linear	(512, 512)
	2	Linear	(512, 256)
	3	Linear	(256, 1)
Critic	1	Linear	(512, 512)
	2	Linear	(512, 256)
	3	Linear	(256, 1)

loss for the non-augmented sample, i.e., $\mathbf{x}^- = e(\mathbf{m} = 0, \mathbf{x}) = \mathbf{x}$. Lastly, based on Eq. (1), the loss for the adaptively augmented data is denoted as $\mathcal{L}_{ada}(f_\theta(\tilde{\mathbf{x}}, \mathbf{y}))$, with the augmentation magnitude determined by the actor network. Inspired by the curriculum learning [29], we formulate the reward function as follows:

$$r = \lambda(\mathcal{L}_{full} - \mathcal{L}_{ada}) + (1 - \lambda)(\mathcal{L}_{ada} - \mathcal{L}_{none}), \quad (2)$$

where $\lambda \in [0, 1]$ is an adjusting factor, which is initialized at 1 and gradually decreases to 0 during the training process. Through this adaptive mechanism, we encourage the RL module to apply weaker DA at the beginning of training to accelerate model learning and progressively increase the DA magnitude in later stages to enhance sample diversity and mitigate the risk of overfitting.

F. Policy Learning

The policy aims to determine the instance-level magnitudes for augmentation operations. For policy learning, we employ the widely-used A2C algorithm [64] due to its lightweight architecture. The details of the A2C architecture are presented in Table II, which consists of an actor network θ_a and a critic network θ_c , which only contains several linear layers. The actor network learns the policy, i.e., the probability distribution over actions given a particular state, $\pi(a | s)$. Meanwhile, the critic network estimates the value associated with a particular state, denoted as $V_{\theta_c}(s)$.

To update the actor and critic networks, we reformulate the loss functions tailored to our specific problem scenario. Specifically, the loss function for updating θ_a is defined as:

$$\mathcal{L}_{actor} = -\log \pi_{\theta_a}(a | s_{none})(r + \gamma V_{\theta_c}(s_{ada}) - V_{\theta_c}(s_{none})). \quad (3)$$

Meanwhile, the loss function for updating θ_c is defined as:

$$\mathcal{L}_{critic} = \mathbb{E}[(r + \gamma V_{\theta_c}(s_{ada}) - V_{\theta_c}(s_{none}))^2]. \quad (4)$$

To provide a comprehensive insight into AdaAugment, Algorithm 1 offers a detailed algorithmic procedure.

G. Theoretical Analysis

In the realm of data augmentation, the augmentation magnitude is approximately proportional to the loss values of the augmented samples, i.e., $m \propto \mathcal{L}$. Therefore, the following inequality approximately holds: $\mathcal{L}_{none} \leq \mathcal{L}_{ada} \leq \mathcal{L}_{full}$, which suggests that \mathcal{L}_{none} and \mathcal{L}_{full} can serve as real-time indicators of potential overfitting and underfitting status associated with

Algorithm 1 The General Workflow of AdaAugment

Input: dataset \mathcal{D} , batch size B , an augmentation space \mathcal{E} , cross-entropy loss \mathcal{L} , total number of epochs T .

- 1: Randomly initialize f_θ , θ_a , and θ_c
- 2: $\mathbf{m} \leftarrow \mathbf{0}$
- 3: **for** $t = 0: T - 1$ **do**
- 4: Sample a mini-batch $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^B$ from \mathcal{D}
- 5: Sample a random augmentation operation e from \mathcal{E}
- 6: Generate the adaptively augmented sample $\tilde{\mathbf{x}}_i$ with corresponding magnitude $m \in \mathbf{m}$ and fully augmented sample \mathbf{x}_i^+ according to Eq. (1)
- 7: Compute the loss items $\mathcal{L}_{none}(f_\theta(\mathbf{x}_i), \mathbf{y}_i)$, $\mathcal{L}_{full}(f_\theta(\mathbf{x}_i^+), \mathbf{y}_i)$, and $\mathcal{L}_{ada}(f_\theta(\tilde{\mathbf{x}}_i), \mathbf{y}_i)$
- 8: Determine the corresponding adaptive magnitudes \mathbf{m} using θ_a
- 9: Calculate reward value according to Eq. (2)
- 10: Update θ_a and θ_c according to Eq. (3) and Eq. (4)
- 11: Update f_θ based on $\mathcal{L}_{ada}(f_\theta(\tilde{\mathbf{x}}_i), \mathbf{y}_i)$
- 12: **end for**

Output: f_θ

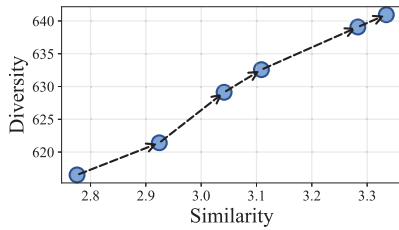


Fig. 3. Evaluation of the adaptively augmented datasets throughout the training process with CIFAR-10 using ResNet-50.

augmented samples, respectively. Meanwhile, these loss values continuously evolve as training progresses.

Regarding the reward function in Eq. (2), the term $\mathcal{L}_{full} - \mathcal{L}_{ada}$ holds significant importance during the initial training phases. Consequently, the policy network tends to employ relatively modest augmentation magnitudes, aiming to maximize the disparity between \mathcal{L}_{full} and \mathcal{L}_{ada} . Therefore, more similar augmented samples are encouraged initially, enabling models to capture broader features or patterns and facilitating rapid convergence during early training stages [50], [67]. As training advances, the focus shifts towards $\mathcal{L}_{ada} - \mathcal{L}_{none}$, prompting the policy network to apply greater augmentation magnitudes, trying to increase the difference between \mathcal{L}_{ada} and \mathcal{L}_{none} . This yields more diverse augmented samples, thereby mitigating overfitting risks. Thus, the methodology aligns with the curriculum learning principle [29], [68], wherein models initially learn patterns from similar samples and subsequently benefit from diverse samples in later training stages.

To further investigate the similarity-diversity trade-off of AdaAugment throughout the training process, we employ the similarity and diversity measures [9] for the augmented data at various training stages. Specifically, we calculate the average magnitude values for all samples at the intervals of 50 epochs. As depicted in Figure 3, it can be observed that the diversity measure increases and the similarity measure decreases

gradually as training progresses, which aligns with our theoretical analysis. This highlights AdaAugment’s inherent adaptability in adjusting DA strengths to harmonize with the evolving training dynamics of the target models.

IV. EXPERIMENTS

Experimental Settings: Following previous work [16], [23], [28], we evaluate the effectiveness of our method across several benchmark datasets, including the coarse-grained classification datasets CIFAR-10/100 [69], Tiny-ImageNet [35], and ImageNet-1k [36]. We then employ transfer learning to evaluate the efficacy of our method in enhancing model generalization performance and transferability. Moreover, we demonstrate the efficacy of AdaAugment on long-tail classification datasets, including ImageNet-LT and Places-LT [37], as well as on several fine-grained datasets, including Oxford Flowers [39], Oxford-IIIT Pets [40], FGVC-Aircrafts [41], and Stanford Cars [38]. In addition, we conduct effect visualization and perform a convergence analysis of AdaAugment on CIFAR-10 throughout the training process. We also provide a complexity analysis of AdaAugment’s computational efficiency and resource utilization. Ablation studies are conducted to show how the components affect performance.

For baseline methods, AutoAugment and Fast-AutoAugment employ 16 operations, including Shear X/Y, Rotate, AutoContrast, Invert, Equalize, Solarize, Posterize, Contrast, Color, Brightness, Sharpness, Cutout, and Sample pairing. It randomly chooses from 25 augmentation policies on CIFAR-10/100 and 24 on ImageNet-1k. Each policy contains 2 augmentation operations. RandAugment contains 14 operations, which is the same as Table I, and it randomly selects N operations from the augmentation space with fixed magnitudes of M . TeachAugment trains augmentation models to represent geometric augmentation and color augmentation, which can represent most operations employed in AutoAugment. TrivialAugment employs 21 augmentation operations. Besides the ones used in Table I, it also uses Cutout, Invert, Flip-X/Y, Sample pairing, Blur, and Smooth. Moreover, the probability of CutMix is 0.5 on CIFAR10/100 and 1.0 on ImageNet-1k. The probability of Cutout and AdvMask is 1.0.

Comparison with State-of-the-arts: We compare our method with 13 most representative and commonly-used data augmentation methods, including HaS [45], Fast-AutoAugment (FAA) [46], DADA [28], Cutout [14], CutMix [18], MADAug [50], AdvMask [16], GridMask [15], AutoAugment [21], RandAugment [22], TeachAugment [25], TrivialAugment [23], and RandomErasing [43].

Implementation Details: The discount factor γ in Eq. (3) and Eq. (4) is set to 0.99 in our method, following [70], [71], and [72]. We closely follow previous works [16], [23] in our setup. Specifically, all images are preprocessed by dividing each pixel value by 255 and normalized by the dataset statistics. For CIFAR-10 and CIFAR-100, we utilize ResNet-18/50 [73], Wide-ResNet-28-10 (WRN-28-10) [74], and Shake-Shake-26 \times 32 (ShakeShake) [75]. We train 1800 epochs with cosine learning rate decay for Shake-Shake using SGD with

TABLE III
TEST ACCURACY (%) ON CIFAR-10/100 (AVERAGE ± STD). * MEANS RESULTS REPORTED IN PREVIOUS PAPERS

Method	CIFAR-10				CIFAR-100			
	ResNet-18	ResNet-50	WRN-28-10	ShakeShake	ResNet-18	ResNet-50	WRN-28-10	ShakeShake
baseline	95.28±0.14*	95.66±0.08*	95.52±0.11*	94.90±0.07*	77.54±0.19*	77.41±0.27*	78.96±0.25*	76.65±0.14*
HaS [45]	96.10±0.14*	95.60±0.15	96.94±0.08	96.89±0.10*	78.19±0.23	78.76±0.24	80.22±0.16	76.89±0.33
FAA [46]	95.99±0.13	96.69±0.16	97.30±0.24	96.42±0.12	79.11±0.09	79.08±0.12	79.95±0.12	81.39±0.16
DADA [28]	95.58±0.06	95.61±0.14	97.30±0.13*	97.30±0.14*	78.28±0.22	80.25±0.28	82.50±0.26*	80.98±0.15
Cutout [14]	96.01±0.18*	95.81±0.17	96.92±0.09	96.96±0.09*	78.04±0.10*	78.62±0.25	79.84±0.14	77.37±0.28
CutMix [18]	96.64±0.62*	96.81±0.10*	96.93±0.10*	96.47±0.07	79.45±0.17	81.24±0.14	82.67±0.22	79.57±0.10
MADAug [50]	96.49±0.10	97.12±0.11	97.48±0.12	97.37±0.11	79.39±0.19	81.40±0.10	83.01±0.11	81.67±0.18
AdvMask [16]	96.44±0.15*	96.69±0.10*	97.02±0.05*	97.03±0.12*	78.43±0.18*	78.99±0.31*	80.70±0.25*	79.96±0.27*
GridMask [15]	96.38±0.17	96.15±0.19	97.23±0.09	96.91±0.12	75.23±0.21	78.38±0.22	80.40±0.20	77.28±0.38
AutoAugment [21]	96.51±0.10*	96.59±0.04*	96.99±0.06	97.30±0.11	79.38±0.20	81.34±0.29	82.21±0.17	82.19±0.19
RandAugment [22]	96.47±0.32	96.25±0.06	96.94±0.13*	97.05±0.15	78.30±0.15	80.95±0.22	82.90±0.29*	80.00±0.29
TeachAugment [25]	96.47±0.09	96.40±0.10	97.50±0.12	97.29±0.14	79.27±0.17	80.54±0.23	82.81±0.20	81.30±0.20
TrivialAugment [23]	96.28±0.10	97.07±0.08	97.18±0.11	97.30±0.10	78.67±0.19	81.34±0.18	82.75±0.26	82.14±0.16
RandomErasing [43]	95.69±0.10	95.82±0.17	96.92±0.09	96.46±0.13*	75.97±0.11*	77.79±0.32	80.57±0.15	77.30±0.18
AdaAugment	96.75±0.06	97.34±0.13	97.66±0.07	97.41±0.06	79.84±0.27	81.46±0.12	83.23±0.23	82.82±0.25

Nesterov Momentum and a learning rate of 0.01, a batch size of 256, $1e^{-3}$ weight decay. We train 300 epochs for all other networks using SGD with Nesterov Momentum and a learning rate of 0.1, a batch size of 128, a $5e^{-4}$ weight decay, and cosine learning rate decay. For Tiny-ImageNet, we resize the images into 64×64 , initialize models with ImageNet pre-trained weights, and then fine-tune models using various augmentations. In each experiment, we perform three independent random trials. For ImageNet-1k, following [21] and [23], ResNet-50 is used as the target network. We resize images into 224×224 , use the learning rate of 0.1, a batch size of 256, the Nesterov Momentum with a momentum parameter of 0.9, and a weight decay of 0.4. Note that because of the huge computational cost, the experiment in each case is performed once. The baseline approach involves solely padding and horizontal flipping. For a fair comparison, all methods are implemented with the same training configurations. If not specified, all experiments are conducted on an 8-NVIDIA-2080TI-GPU server.

A. Results on CIFAR-10 and CIFAR-100

In Table III, we evaluate the efficacy of AdaAugment on CIFAR-10 and CIFAR-100, employing various widely-used deep networks, such as ResNet-18/50 [73], Wide-ResNet-28-10 [74], and ShakeShake-26-32 [75]. It can be observed that AdaAugment consistently exhibits superior performance in improving the accuracy of these networks when compared to existing SOTA DA methods. Notably, our method demonstrates significant improvements for both datasets. For example, AdaAugment achieves improvements on the baseline model’s performance of ResNet-18, ResNet-50, Wide-ResNet-28-10, and ShakeShake-26-32 by 1.47%, 1.66%, 2.14%, and 2.51%, respectively. While on smaller models, the performance differences among different methods are marginal, using larger models can further highlight the effectiveness of AdaAugment. For instance, AdaAugment outperforms CutMix by nearly 1% on CIFAR-10 and 3.3% on CIFAR-100 with ShakeShake. This superior performance of AdaAugment can

TABLE IV
IMAGE CLASSIFICATION ACCURACY (%) ON TINY-IMAGENET DATASET (AVERAGE ± STD)

Method	ResNet-18	ResNet-50	WRN-50-2	ResNext-50
baseline	61.38±0.99	73.61±0.43	81.55±1.24	79.76±1.89
HaS [45]	63.51±0.58	75.32±0.59	81.77±1.16	80.52±1.88
FAA [46]	68.15±0.70	75.11±2.70	82.90±0.92	81.04±1.92
DADA [28]	70.03±0.10	78.61±0.34	83.03±0.18	81.15±0.34
Cutout [14]	68.67±1.06	77.45±0.42	82.27±1.55	81.16±0.78
CutMix [18]	64.09±0.30	76.41±0.27	82.32±0.46	81.31±1.00
MADAug [50]	70.16±0.76	78.62±0.32	82.38±0.42	81.41±1.26
AdvMask [16]	65.29±0.20	78.84±0.28	82.87±0.55	81.38±1.54
GridMask [15]	62.72±0.91	77.88±2.50	82.25±1.47	81.05±1.33
AutoAugment [21]	67.28±1.40	75.29±2.40	79.99±2.20	81.28±0.33
RandAugment [22]	65.67±1.10	75.87±1.76	82.25±1.02	80.36±0.62
TeachAugment [25]	70.05±0.57	70.56±0.44	82.95±0.13	81.39±0.97
TrivialAugment [23]	69.97±0.96	78.41±0.39	82.16±0.32	80.91±2.26
RandomErasing [43]	64.00±0.37	75.33±1.58	81.89±1.40	81.52±1.68
AdaAugment	71.25±0.64	79.11±1.51	83.07±0.78	81.92±0.29

TABLE V
IMAGE CLASSIFICATION ACCURACY (%) ON TINY-IMAGENET DATASET WITH RANDOMLY INITIALIZED RESNET-50

baseline	HaS	FAA	DADA	Cutout	CutMix	GridMask
49.73±.37	52.05±.12	52.83±.02	53.07±.28	50.89±.52	52.31±.20	50.37±.11
MADAug	AA	RA	TeachAug	TA	RE	AdaAugment
54.95±.24	53.37±.13	50.63±.53	53.97±.23	53.88±.03	50.74±.27	55.98±.12

be attributed to its adaptive adjustment of augmentation magnitudes based on real-time feedback from the target network during training. This adjustment effectively mitigates the risks of overfitting, resulting in improved generalization capabilities. Additionally, larger models do not necessarily lead to better performance. The baseline performance of ResNet-50 underperforms that of ResNet-18 by applying simple augmentations. This is likely due to higher overfitting risks for the larger models. However, when more advanced augmentation methods are employed, ResNet-50 consistently outperforms ResNet-18, further validating the effectiveness of DA.

B. Results on Tiny-ImageNet

In this section, we evaluate the efficacy of AdaAugment on the Tiny-ImageNet dataset using pre-trained ResNet-18/50,

TABLE VI

TOP-1 ACCURACY (%) ON IMAGENET-1K DATASET WITH RESNET-50. SOME RESULTS ARE ADAPTED FROM [23], [25], AND [28]

baseline	HaS [45]	FAA [46]	DADA [28]	Cutout [14]	CutMix [18]	MADAUG [50]	GridMask [15]	AA [21]	RA [22]	TeachAug [25]	TA [23]	RE [43]	AdaAugment
77.1	77.2	77.6	77.5	77.1	77.2	78.3	77.9	77.6	77.8	78.0	77.9	77.3	78.3

TABLE VII

PERFORMANCE (%) ON IMAGENET-1K WITH ViT-BASE, ViT-LARGE, AND SWIN-TRANSFORMER USING A 4-V100-GPU SERVER

Model	baseline	AdaAugment
ViT-B	81.46	82.55
ViT-L	83.50	84.67
Swin-Transformer	85.00	85.65

Wide-ResNet-50-2 (WRN-50-2), and ResNext50 [76], with the results summarized in Table IV. AdaAugment achieves significant improvements in classification accuracy across various architectures, outperforming other widely used DA methods by a substantial margin. Particularly, AdaAugment elevates the accuracy by 9.87%, 5.5%, 1.52%, and 1.16% on ResNet-18, ResNet-50, WRN-50-2, and ResNext-50, respectively. Furthermore, Table V presents the results of training ResNet-50 on Tiny-ImageNet from scratch. Even in this scenario, AdaAugment delivers superior performance, achieving an improvement of over 1%. These results highlight the effectiveness of our adaptive augmentation mechanism in enhancing both pre-trained and from-scratch training setups. Thus, we demonstrate the superior efficacy and potential of AdaAugment for enhancing model performance.

C. Results on ImageNet-1k

We evaluate AdaAugment on the more challenging large-scale ImageNet-1k dataset [36]. Note that the method AdvMask is not compared on ImageNet-1k due to its huge time costs of the sparse adversarial attack module [16]. As shown in Table VI and Table VII with more advanced baseline models, on ImageNet-1k, all DA methods' average accuracy significantly outperforms the baseline. Moreover, AdaAugment achieves the highest improvements in accuracy using ResNet-50. Notably, AdaAugment achieves 1.1% improvements compared to the baseline without any architecture modifications and noticeable additional training costs. Although MADAUG obtains similar results to ours on ImageNet-1k, it involves a bi-level optimization scheme that introduces much higher training costs than AdaAugment. Moreover, in contrast to previous works, which employed fixed augmentation operations for the entire dataset, our method applies model- and sample-adaptive operations for each image. This significantly enhances the model's generalization.

D. Transfer Learning

Transfer learning plays a pivotal role in assessing the models' transferability [77], a common practice in data augmentation [9], [16], [30]. To quantitatively assess the efficacy of various DA methods, we assess the transferred accuracy

of models using different DA methods. Specifically, we pre-train ResNet-50 models on CIFAR-100 and Tiny-ImageNet utilizing different augmentations, followed by fine-tuning these models on the CIFAR-10 dataset. Consequently, superior DA techniques result in higher transferred test accuracy.

In Table VIII, we present the transferred accuracy achieved by various DA methods. While the discrepancies in transferred accuracy may appear modest, it is noteworthy that AdaAugment consistently outperforms others. This consistent and effective performance underscores our method's capability to enhance model transferability performance. Thus, AdaAugment indeed enhances the model generalization performance.

E. Results on Long-Tail Datasets

While most DA methods have not been tested on large-scale long-tail biased datasets, we strengthen the generality and effectiveness of our method by applying it to the ImageNet-LT and Places-LT datasets [37]. Specifically, we utilize the codebase provided by OLTR [37] and rigorously follow all the hyperparameters and training strategies.

As shown in Table IX, AdaAugment brings substantial improvements across all the many/medium/few-shot classes and the open classes in both long-tail datasets. Notably, AdaAugment achieves **over 3% accuracy improvements in the closed-set setting**, highlighting the effectiveness of our method in improving model robustness and generalization. Therefore, AdaAugment can also be effectively employed in more challenging long-tail datasets as well.

F. Results on Fine-Grained Datasets

To comprehensively assess the effectiveness of our proposed methods, we apply AdaAugment to various fine-grained datasets, including Oxford Flowers [39], Oxford-IIIT Pets [40], FGVC-Aircraft [41], and Stanford Cars [38]. Specifically, for all the fine-grained datasets, we employ the ResNet-50 model [73] pre-trained on ImageNet, followed by fine-tuning these models using our proposed methods. To ensure fairness, experiments on the same dataset utilize the same experimental settings.

Table X summarizes the test accuracy of our methods compared to the baseline. It can be observed that AdaAugment brings notable accuracy improvements across all fine-grained datasets. Specifically, AdaAugment improves the model performance on Oxford Flowers by 7.70%, Oxford-IIIT Pets by 2.22%, FGVC-Aircraft by 13.67%, and Stanford Cars by 2.63%, respectively. Hence, our proposed methods can also be utilized to substantially enhance the model performance on fine-grained datasets.

TABLE VIII
TRANSFERRED TEST ACCURACY (%) ON CIFAR-10 OF VARIOUS DA METHODS. THE PRETRAINED RESNET-50 MODEL IS TRAINED ON BOTH CIFAR-100 (UPPER ROW) AND TINY-IMAGENET (BOTTOM ROW)

baseline	HaS [45]	FAA [46]	DADA [28]	Cutout [14]	CutMix [18]	MADAUG [50]	GridMask [15]	AA [21]	RA [22]	TeachAug [25]	TA [23]	RE [43]	AdaAugment
91.53±0.03	92.51±0.24	92.28±0.13	92.58±0.09	92.42±0.20	92.81±0.47	92.84±0.10	91.49±0.10	92.82±0.04	92.78±0.23	92.83±0.18	92.80±0.16	92.55±0.05	93.06±0.25
64.02±0.05	66.84±0.06	70.32±0.63	69.04±0.43	65.54±0.75	69.29±0.09	72.82±0.32	64.88±0.43	69.53±0.53	64.68±0.97	69.98±0.17	71.53±0.35	64.56±0.27	76.86±0.12

TABLE IX
TOP-1 CLASSIFICATION ACCURACY (%) ON IMAGENET-LT AND PLACES-LT. * MEANS RESULTS REPORTED IN THE ORIGINAL PAPER

Dataset	Methods	closed-set setting				open-set setting			
		Many-shot	Medium-shot	Few-shot	Overall	Many-shot	Medium-shot	Few-shot	F-measure
ImageNet-LT	OLTR	43.2±0.1*	35.1±0.2*	18.5±0.1*	35.6±0.1*	41.9±0.1*	33.9±0.1*	17.4±0.2*	44.6±0.2*
	OLTR+AdaAugment	45.9±0.1	38.3±0.1	22.0±0.2	39.0±0.1	44.1±0.1	36.8±0.1	20.8±0.2	45.8±0.1
Places-LT	OLTR	44.7±0.1*	37.0±0.2*	25.3±0.1*	35.9±0.1*	44.6±0.1*	36.8±0.1*	25.2±0.2*	46.4±0.1*
	OLTR+AdaAugment	43.7±0.1	41.1±0.1	29.5±0.2	39.6±0.1	43.9±0.1	40.8±0.1	28.9±0.1	50.4±0.1

TABLE X
TEST ACCURACY (%) OF ADAUGMENT ON FINE-GRAINED DATASETS WITH RESNET-50. THE PERFORMANCE IS AVERAGE±STD

Dataset	baseline	AdaAugment
Oxford Flowers [39]	89.47±0.08	97.17±0.14
Oxford-IIIT Pets [40]	89.73±0.18	91.95±0.24
FGVC-Aircraft [41]	77.25±0.09	90.92±0.05
Stanford Cars [38]	82.13±0.03	84.76±0.20

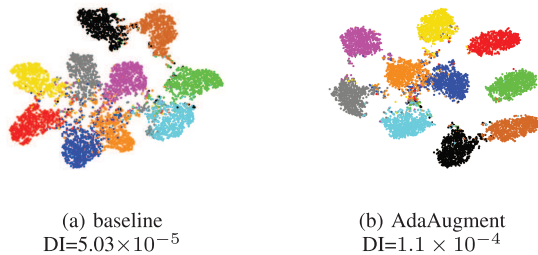


Fig. 4. Visualization of effectiveness using t-SNE algorithm with CIFAR-10 dataset. The embedding model is ResNet-50. DI: Dunn index.

G. Analytical Results

1) *Visualization of Effectiveness*: Since the primary objective of DA is to enhance the generalization performance of models, in this section, we conduct a comparative analysis of effectiveness visualization between models utilizing AdaAugment and those without. Specifically, we leverage the t-SNE algorithm [78] to visualize the feature embedding of the CIFAR-10 test set, utilizing feature maps derived from different trained models. Theoretically, models that showcase more robust generalization capabilities tend to excel in feature extraction ability.

Figure 4 presents the t-SNE visualization results for the baseline model and the model trained with AdaAugment. Compared to Figure 4(a), it can be seen that Figure 4(b) exhibits an optimized geometric structure characterized by **enhanced inter-cluster separation and intra-cluster compactness**. Furthermore, to furnish a quantitative evaluation of these embedding results, we employ the Dunn Index (DI) [79]

as an evaluative metric. The DI is mathematically defined as follows:

$$DI = \frac{\min_{1 \leq i \neq j \leq m} \delta(C_i, C_j)}{\max_{1 \leq j \leq m} \Delta_j}, \tag{5}$$

where separation $\delta(C_i, C_j)$ denotes the inter-cluster distance metric between clusters C_i and C_j , and compactness Δ_j calculates the mean distance between all pairs within each cluster. A higher DI value signifies superior clustering performance. The DI values for the baseline in Figure 4(a) and AdaAugment in Figure 4(b) are 5.03×10^{-5} and 1.05×10^{-4} , respectively. Consequently, AdaAugment achieves a DI value that is **108.7% higher** than that of the baseline, demonstrating its superior efficacy in enhancing model performance. Through both quantitative and qualitative analyses, we validate the effectiveness of AdaAugment in improving model performance.

2) *Complexity Analysis*: Most existing DA methods typically rely on predefined random magnitude values during training. While this mechanism incurs minimal additional computational overhead in online training, it may not be optimally aligned with the dynamic evolution of deep models, potentially introducing undesirable side effects. To address this, AdaAugment incorporates an auxiliary policy network into online training to dynamically determine the magnitudes of DA operations. Although this adjustment inevitably introduces slightly additional parameters and computational complexity, it is a deliberate trade-off to achieve more effective DA.

In this section, following the methodology employed in prior studies [21], [24], [46], we thoroughly analyze the parameter and time complexity of AdaAugment to assess its efficiency. Since the policy network’s parameter complexity is related to the feature space of the target network, we report the parameter complexity on ResNet-18/50 and WRN-28-10 in Table XI. Notably, it can be observed that **the policy network’s parameter complexity results in a mere 1.3% increase for ResNet-18** (with 11.7M parameters), **a 2.4% increase for ResNet-50** (with 25.5M parameters), and **a 0.52% increase for WRN-28-10** (with 36.5M parameters). Furthermore, the overall training costs experience a slight increase for these

TABLE XI

THE ADDITIONAL ARCHITECTURE PARAMETERS AND TRAINING TIME OF THE AUXILIARY POLICY NETWORK ON CIFAR-10. THE DEVICE USED IS 2 NVIDIA RTX2080TI GPUS AND AN INTEL(R) XEON(R) CPU E5-2678 @ 2.50GHZ

Models	FLOPs	Parameters	GPU hours	Improved Acc.
ResNet-18	1.82G	+0.15M	+0.41h±0.03	+1.47%±0.06
ResNet-50	4.14G	+0.60M	+0.49h±0.03	+1.68%±0.13
WRN-28-10	5.25G	+0.19M	+0.43h±0.03	+2.04%±0.07

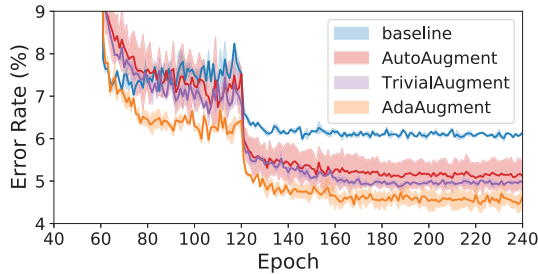


Fig. 5. Convergence analysis of training process with CIFAR-10 using ResNet-50.

networks. Despite the marginal increases in training costs, AdaAugment achieves significant accuracy improvements, with a notable enhancement of 1.47%, 1.68%, and 2.04% for ResNet-18, ResNet-50, and WRN-28-10, respectively. This improvement underscores AdaAugment’s remarkable effectiveness-efficiency trade-off, demonstrating its capacity to enhance model performance while incurring minimal training costs.

3) *Convergence Analysis*: In this section, we conduct a convergence analysis of AdaAugment throughout the entire training process, comparing it against two of the most representative DA methods, AutoAugment and TrivialAugment. Specifically, to illustrate the dynamic evolution of test errors during training, we train ResNet-18 models on CIFAR-10 using a multi-step learning rate decay schedule. The learning rate is initialized as 0.1 and multiplied by 0.2 at epochs 60, 120, 160, 220, and 280. From Figure 5, it can be observed that AdaAugment significantly enhances model performance, particularly after the second learning rate drop. Moreover, even after the initial learning rate drop, AdaAugment consistently maintains lower error rates than other methods. These empirical findings not only underscore the superior efficacy of AdaAugment in terms of model performance enhancement but also emphasize its capacity to expedite the convergence of the model during training. Therefore, AdaAugment contributes to decreasing the training difficulty of deep networks.

4) *Adaptively Evolving Magnitude*: To present the evolving progress of the adaptive magnitudes, we train ResNet-50 with CIFAR-100. The learning rate is initialized as 0.1 and multiplied by 0.2 at epochs 60, 120, 160, 220, and 280, along with a 5-epoch scheduler warmup. In Figure 6, it can be seen that the magnitude generally increases through model training, providing progressively more diverse augmented samples to mitigate the risk of overfitting. Moreover, this dynamic alignment aligns with the model’s training status. For instance, at different training stages, such as the warmup phase at

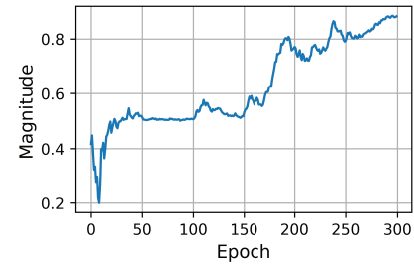
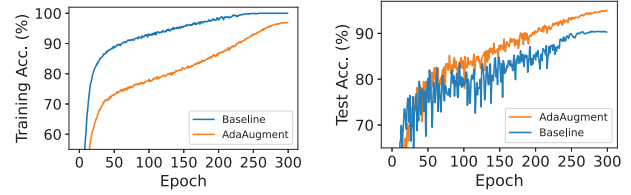


Fig. 6. Evolving magnitudes throughout the training process with CIFAR-100 using ResNet-50.



(a) Training Accuracy.

(b) Test Accuracy.

Fig. 7. Effectiveness of AdaAugment in mitigating overfitting risks with reduced CIFAR-10 using ResNet-50.

the beginning of training or during learning rate changes, the magnitudes adaptively modify values, aligning with the model’s evolving status.

5) *Effectiveness of Mitigating Overfitting*: Although modern deep models can fit the distribution of training datasets through training, this capability often faces overfitting risks, which degrades models’ generalization performance, particularly in scenarios with limited training data. Data augmentation is primarily designed to address this issue by increasing the diversity of training data. To evaluate the effectiveness of AdaAugment in mitigating overfitting risks, we simulate a high-overfitting-risk scenario by training a large ResNet-50 model on the relatively simple CIFAR-10 dataset. To further amplify the overfitting risk, we randomly reduce the sizes of the training set by 50% while keeping the test set unchanged.

As can be seen in Figure 7, without AdaAugment, models achieve very high training accuracy, nearing 100%, but the test accuracy rises slowly and exhibits significant fluctuations, indicating poor generalization and significant overfitting. With AdaAugment, models achieve slower training accuracy growth and consistently lower training accuracy. Despite this, the test accuracy is significantly higher and more stable, demonstrating AdaAugment’s ability to mitigate overfitting and enhance generalization performance.

H. Ablation Study

1) *Effect of the Augmentation Space*: Although the augmentation space used in AdaAugment is not larger than other methods [21], [22], [23], [46], [50], we further conduct analytical experiments to evaluate the impact of base augmentations. Specifically, we analyze the performance by changing the number of augmentation transformations in our augmentation space. We present the results in Table XII. Notably, the number of augmentation types used in Table XII is consistently lower

TABLE XII
IMPACT OF THE AUGMENTATION SPACE WITH CIFAR-10 USING RESNET-18

# of transformations	2	4	6	8	10	12	14
Accuracy (%)	95.99±0.02	96.18±0.04	96.38±0.05	96.54±0.10	96.61±0.11	96.67±0.09	96.75±0.06

TABLE XIII
IMPACT OF ADAPTIVE MAGNITUDE m . COMPARATIVE ANALYSIS OF ADAUGMENT COMPARED TO DIFFERENT SETTINGS OF m WITH CIFAR-100 USING RESNET-18/50

Model	$m = 0.5$	random m	linear m	sine m	AdaAugment
R-18	78.62±0.32	77.08±0.30	78.38±0.27	78.58±0.32	79.84±0.27
R-50	80.23±0.29	80.61±0.19	80.28±0.31	80.65±0.15	81.46±0.12

TABLE XIV
EFFECT OF DIFFERENT RL MODULES WITH RESNET-18 ON CIFAR-10/100

Dataset	DDPG [62]	SAC [61]	Ours
CIFAR-10	96.73±0.06	96.82±0.04	96.75±0.06
CIFAR-100	79.27±0.19	79.80±0.29	79.84±0.27

than other baseline approaches. It can be observed that while the accuracy decreases as fewer augmentations are utilized, it drops very slowly. When $N = 2$, only two types of augmentations will be employed, which limits the diversity of augmented data. Even under this case, the accuracy drops less than 0.8%. This highlights the effectiveness of AdaAugment in adaptively adjusting the augmentation magnitudes, effectively enhancing the model generalization.

2) *Effect of the RL Module*: By utilizing the RL module to adaptively adjust the augmentation magnitude of DA operations, AdaAugment eliminates the need for manual parameter design and adjustment. In this section, we conduct experiments to assess the effect of the RL module. By using the same augmentation space, we employ different experimental settings on m : fixed and random values of m , as well as increasing m using linear and sine functions. The results are summarized in Table XIII. It can be seen that the RL module brings substantial improvements in model performance. By estimating the real-time model training status through the RL module, AdaAugment can optimize the augmentation magnitudes during training. Thus, we demonstrate that the superior efficacy is brought by the adaptive adjustment of augmentation magnitude rather than the base augmentation space.

Additionally, in AdaAugment, we utilize lightweight A2C architectures. Here, we also employ two other more recent RL architectures, including Soft Actor-Critic (SAC) [61] and Deep Deterministic Policy Gradient (DDPG) [62]. As can be seen in Table XIV, the performance differences among different RL modules are relatively marginal.

3) *Effect of the Discount Factor γ* : Following previous works [70], [71], [72], we set γ as a constant 0.99. Here, we investigate the effect of different γ values. The results are presented in Table XV. It can be seen that AdaAugment performs best under our setting.

4) *Effect of Parameter λ in Eq. (2)*: We conduct analytical experiments to evaluate the effect of parameter λ in Eq. (2).

TABLE XV
EFFECT OF γ WITH CIFAR-100 USING RESNET-18/50

γ	0.2	0.4	0.6	0.8	0.99
R-18	78.94±0.33	78.95±0.20	78.99±0.29	79.24±0.25	79.84±0.27
R-50	80.61±0.15	80.44±0.20	80.68±0.17	80.90±0.19	81.46±0.12

TABLE XVI
EFFECT OF λ WITH CIFAR-100 USING RESNET-18/50

λ	0	1	Ours
ResNet-18	78.53±0.36	78.70±0.30	79.84±0.27
ResNet-50	80.21±0.19	80.74±0.29	81.46±0.12

Specifically, instead of dynamically adjusting λ , λ is fixed as 0 and 1. The results are shown in Table XVI, validating the stability of parameter λ .

V. DISCUSSION AND FUTURE WORK

In this work, we propose AdaAugment, a tuning-free adaptive data augmentation method. Instead of relying on any hand-crafted criteria, AdaAugment incorporates an RL module to adaptively adjust the augmentation magnitudes for individual training samples. This adaptive adjustment aligns with the model training progress. Experiment results demonstrate that AdaAugment achieves enhanced model performance across various benchmark datasets without introducing noticeable training overhead. Besides strengths, the limitations of our method should also be discussed.

While the proposed method estimates the risks of underfitting and overfitting by deriving losses from fully augmented and nonaugmented data, this additional loss calculation requires two extra forward passes compared to vanilla training. To enhance the overall efficiency, our policy network is designed to be consistent across various target models. Thus, the increase in GPU hours is relatively modest and not directly correlated with the complexity of the task model. Future work should consider leveraging other surrogate indicators to estimate underfitting and overfitting risks.

In our augmentation space, we leverage image transformations whose applied strengths can be adjusted with a real-valued parameter, i.e., $m \in [0, 1]$. However, most existing DA methods, such as Cutout [14], AdvMask [16], and AutoAugment [21], do not support explicit adjustments of augmentation magnitudes. Therefore, future work should consider exploring an adjustable version of existing advanced DA methods, facilitating explicit adjustment of augmentation strength with continuous real values. Consequently, our augmentation space may be further expanded. This expansion holds promise for further performance enhancements.

VI. CONCLUSION

In this work, we propose AdaAugment, an innovative and tuning-free method for adaptive data augmentation. By adaptively adjusting the magnitudes of DA operations based on the real-time feedback obtained during the training of the target model, AdaAugment optimizes the data variability. This dynamic adjustment ultimately mitigates the risks associated with both underfitting and overfitting, thereby optimizing model training and enhancing model generalization capabilities. Through extensive empirical evaluations, we demonstrate that our adaptive augmentation mechanism helps significantly improve performance across various benchmark datasets with competitive efficiency.

REFERENCES

- [1] X. Ren, W. Lin, X. Yang, X. Yu, and H. Gao, "Data augmentation in defect detection of sanitary ceramics in small and non-i.i.d datasets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8669–8678, Nov. 2023.
- [2] Y. Tang, B. Li, M. Liu, B. Chen, Y. Wang, and W. Ouyang, "AutoPedestrian: An automatic data augmentation and loss function search scheme for pedestrian detection," *IEEE Trans. Image Process.*, vol. 30, pp. 8483–8496, 2021.
- [3] H. Zhang, Z. Xu, X. Han, and W. Sun, "Data augmentation using bit-plane information recombination model," *IEEE Trans. Image Process.*, vol. 31, pp. 3713–3725, 2022.
- [4] T.-H. Cheung and D.-Y. Yeung, "A survey of automated data augmentation for image classification: Learning to compose, mix, and generate," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13185–13205, Oct. 2024.
- [5] M. Wang et al., "Smooth-guided implicit data augmentation for domain generalization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, pp. 4984–4995, Mar. 2024.
- [6] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109347.
- [7] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Jul. 2019.
- [8] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," 2022, *arXiv:2204.08610*.
- [9] S. Yang, S. Guo, J. Zhao, and F. Shen, "Investigating the effectiveness of data augmentation from similarity and diversity: An empirical study," *Pattern Recognit.*, vol. 148, Apr. 2024, Art. no. 110204.
- [10] J. Gao, Y. Hua, G. Hu, C. Wang, and N. M. Robertson, "Discrepancy-guided domain-adaptive data augmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 5064–5075, Aug. 2023.
- [11] Y. Pu, Y. Han, Y. Wang, J. Feng, C. Deng, and G. Huang, "Fine-grained recognition with learnable semantic data augmentation," *IEEE Trans. Image Process.*, vol. 33, pp. 3130–3144, 2024.
- [12] S. Yang, P. Ye, F. Shen, and D. Zhou, "When dynamic data selection meets data augmentation," 2025, *arXiv:2505.03809*.
- [13] F. Meng, H. Liu, Y. Liang, J. Tu, and M. Liu, "Sample fusion network: An end-to-end data augmentation network for skeleton-based human action recognition," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5281–5295, Nov. 2019.
- [14] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.
- [15] P. Chen, S. Liu, H. Zhao, X. Wang, and J. Jia, "GridMask data augmentation," 2020, *arXiv:2001.04086*.
- [16] S. Yang, J. Li, T. Zhang, J. Zhao, and F. Shen, "AdvMask: A sparse adversarial attack-based data augmentation method for image classification," *Pattern Recognit.*, vol. 144, Dec. 2023, Art. no. 109847.
- [17] H. Zhang, M. Cissé, Y. Dauphin, and D. López-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–13. [Online]. Available: <https://openreview.net/forum?id=r1Ddp1-Rb>
- [18] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6023–6032.
- [19] F. Dornaika and D. Sun, "LGOAMix: Local and global context-and-object-part-aware superpixel-based data augmentation for deep visual recognition," *IEEE Trans. Image Process.*, vol. 33, pp. 205–215, 2024.
- [20] Z. Mai, G. Hu, D. Chen, F. Shen, and H. T. Shen, "MetaMixUp: Learning adaptive interpolation policy of MixUp with metalearning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 3050–3064, Jul. 2022.
- [21] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 113–123.
- [22] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 18613–18624.
- [23] S. G. Müller and F. Hutter, "TrivialAugment: Tuning-free yet state-of-the-art data augmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 774–782.
- [24] S. Lin, Z. Zhang, X. Li, and Z. Chen, "SelectAugment: Hierarchical deterministic sample selection for data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 2, pp. 1604–1612.
- [25] T. Suzuki, "TeachAugment: Data augmentation optimization using teacher knowledge," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10904–10914.
- [26] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, "Adaptive data augmentation for image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3688–3692.
- [27] T.-H. Cheung and D.-Y. Yeung, "AdaAug: Learning class-and instance-adaptive data augmentation policies," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16.
- [28] Y. Li, G. Hu, Y. Wang, T. M. Hospedales, N. M. Robertson, and Y. Yang, "DADA: Differentiable automatic data augmentation," 2020, *arXiv:2003.03780*.
- [29] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," *Int. J. Comput. Vis.*, vol. 130, no. 6, pp. 1526–1565, 2022.
- [30] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, "KeepAugment: A simple information-preserving data augmentation approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1055–1064.
- [31] D. Bashir, G. D. Montañez, S. Sehra, P. S. Segura, and J. Lauw, "An information-theoretic perspective on overfitting and underfitting," in *Proc. Australas. Joint Conf. Artif. Intell.*, 2020, pp. 347–358.
- [32] S. Sehra, D. Flores, and G. D. Montañez, "Undecidability of underfitting in learning algorithms," in *Proc. 2nd Int. Conf. Comput. Data Sci. (CDS)*, Jan. 2021, pp. 591–594.
- [33] H. Li, G. K. Rajbahadur, D. Lin, C.-P. Bezemer, Z. Ming, and Jiang, "Keeping deep learning models in check: A history-based approach to mitigate overfitting," *IEEE Access*, vol. 12, pp. 70676–70689, 2024.
- [34] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, Canada, Tech. Rep., 2009. [Online]. Available: <https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf>
- [35] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A downsampled variant of ImageNet as an alternative to the CIFAR datasets," 2017, *arXiv:1707.08819*.
- [36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [37] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2532–2541.
- [38] J. Krause et al., "Collecting a large-scale dataset of fine-grained cars," Stanford AI Lab, Tech. Rep., 2013. [Online]. Available: <https://ai.stanford.edu/~jkrause/papers/fgvc13.pdf>
- [39] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process.*, Dec. 2008, pp. 722–729.
- [40] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3498–3505.
- [41] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013, *arXiv:1306.5151*.
- [42] M. Kang and S. Kim, "GuidedMixup: An efficient mixup strategy guided by saliency maps," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 1, pp. 1096–1104.
- [43] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 13001–13008.
- [44] S. Yang, F. Shen, and J. Zhao, "EntAugment: Entropy-driven adaptive data augmentation framework for image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 197–214.

- [45] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3544–3553.
- [46] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast AutoAugment," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 6665–6675.
- [47] R. Hataya, J. Zdenek, K. Yoshizoe, and H. Nakayama, "Faster AutoAugment: Learning augmentation strategies using backpropagation," in *Proc. Eur. Conf. Comput. Vis. Glasgow, U.K.: Springer*, 2020, pp. 1–16.
- [48] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong, "Adversarial AutoAugment," 2019, *arXiv:1912.11188*.
- [49] J. Rajendran, A. Irpan, and E. Jang, "Meta-learning requires meta-augmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 5705–5715.
- [50] C. Hou, J. Zhang, and T. Zhou, "When to learn what: Model-adaptive data augmentation curriculum," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 1717–1728.
- [51] D. Lee, H. Park, T. Pham, and C. D. Yoo, "Learning augmentation network via influence functions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10961–10970.
- [52] X. Xu and H. Zhao, "Universal adaptive data augmentation," 2022, *arXiv:2207.06658*.
- [53] Y. Zhang, H. Zhu, and S. Yu, "Adaptive data augmentation for contrastive learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.
- [54] N. Le, V. S. Rathour, K. Yamazaki, K. Luu, and M. Savvides, "Deep reinforcement learning in computer vision: A comprehensive survey," *Artif. Intell. Rev.*, vol. 55, no. 4, pp. 1–87, Apr. 2022.
- [55] S. E. Li, "Deep reinforcement learning," in *Reinforcement Learning for Sequential Decision and Optimal Control*. Cham, Switzerland: Springer, 2023, pp. 365–402.
- [56] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.
- [57] Y. Lin et al., "A survey on reinforcement learning for recommender systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13164–13184, Oct. 2024.
- [58] H. Byeon, "Advances in value-based, policy-based, and deep learning-based reinforcement learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, pp. 348–354, no. 8, 2023.
- [59] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 1999, pp. 1057–1063.
- [60] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Syst. Appl.*, vol. 231, Nov. 2023, Art. no. 120495.
- [61] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
- [62] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
- [63] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern., C Appl. Rev.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.
- [64] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.
- [65] D. Huang, Q. Bu, Y. Qing, Y. Fu, and H. Cui, "Feature map testing for deep neural networks," 2023, *arXiv:2307.11563*.
- [66] A. Zhang, X. Wang, C. Fang, J. Shi, T.-S. Chua, and Z. Chen, "A-FMI: Learning attributions from deep networks via feature map importance," 2021, *arXiv:2104.05527*.
- [67] T. Y. Liu and B. Mirzasoleiman, "Data-efficient augmentation for training neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 5124–5136.
- [68] A. S. Azad, İ. Gür, A. Faust, P. Abbeel, and I. Stoica, "CLUTR: Curriculum learning via unsupervised task representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 1361–1395.
- [69] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished Manuscript*, vol. 40, no. 7, pp. 1–9, 2010.
- [70] V. Mnih, "Asynchronous methods for deep reinforcement learning," 2016, *arXiv:1602.01783*.
- [71] M. Shen and J. P. How, "Robust opponent modeling via adversarial ensemble reinforcement learning in asymmetric imperfect-information games," 2019, *arXiv:1909.08735*.
- [72] J. Humpalik, A. Galashov, L. Hasenclever, P. A. Ortega, Y. Whye Teh, and N. Heess, "Meta reinforcement learning as task inference," 2019, *arXiv:1905.06424*.
- [73] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [74] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2016, pp. 87.1–87.12, doi: 10.5244/c.30.87.
- [75] X. Gastaldi, "Shake-shake regularization," 2017, *arXiv:1705.07485*.
- [76] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.
- [77] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.
- [78] L. Van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
- [79] C.-E. Ben Ncir, A. Hamza, and W. Bouaguel, "Parallel and scalable Dunn index for the validation of big data clusters," *Parallel Comput.*, vol. 102, May 2021, Art. no. 102751.



Suorong Yang received the B.S. degree from the Department of Computer Science and Technology, Nanjing University, in 2019. He is currently pursuing the Ph.D. degree in computer science and technology with Nanjing University, under the supervision of Prof. Furao Shen. His research interests include computer vision, data augmentation, and multimodal learning.



Peijia Li received the B.S. degree from the School of Artificial Intelligence, Nanjing University, in 2023, where he is currently pursuing the master's degree, under the supervision of Prof. Furao Shen. His research interests include reinforcement learning and data augmentation.



Xin Xiong received the master's degree from the School of Artificial Intelligence, Nanjing University, in 2024. Her research interests include data augmentation and graph learning.



Furao Shen (Member, IEEE) received the B.Sc. and M.Sc. degrees in mathematics from Nanjing University, Nanjing, China, in 1995 and 1998, respectively, and the Ph.D. degree from Tokyo Institute of Technology, Tokyo, Japan, in 2006. He is currently a Full Professor of computer science and technology with Nanjing University. His current research interests include neural computing and robotic intelligence.



Jian Zhao (Senior Member, IEEE) received the B.S. degree from Nanjing University, Nanjing, China, the M.Sc. degree from the Hamburg University of Technology, Hamburg, Germany, and the Dr.Sc. degree in electrical engineering from the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland. From 2010 to 2015, he was a Research Scientist with the Institute for Infocomm Research, A*STAR, Singapore. He is currently an Associate Professor with the School of Electronic Science and Engineering, Nanjing University. His research interests include deep neural networks, mathematical optimization, and wireless communication networks. He was honored with the Dengfeng Scholars Program of Nanjing University in 2015, the IEEE Globecom 2008 Best Paper Award, and the 2009 Chinese Government Award for Outstanding Self-Financed Students Abroad.