SelectAugment: Hierarchical Deterministic Sample Selection for Data Augmentation

Shiqi Lin[1*], **Zhizheng Zhang**[2*], **Xin Li**[1], **Zhibo Chen**[1†]

[1] University of Science and Technology of China
[2] Microsoft Research Asia
{linsq047, lixin666}@mail.ustc.edu.cn, zhizzhang@microsoft.com, chenzhibo@ustc.edu.cn

Abstract

Data augmentation (DA) has been extensively studied to facilitate model optimization in many tasks. Prior DA works focus on designing augmentation operations themselves, while leaving selecting suitable samples for augmentation out of consideration. This might incur visual ambiguities and further induce training biases. In this paper, we propose an effective approach, dubbed SelectAugment, to select samples for augmentation in a deterministic and online manner based on the sample contents and the network training status. To facilitate the policy learning, in each batch, we exploit the hierarchy of this task by first determining the augmentation ratio and then deciding whether to augment each training sample under this ratio. We model this process as two-step decision-making and adopt Hierarchical Reinforcement Learning (HRL) to learn the selection policy. In this way, the negative effects of the randomness in selecting samples to augment can be effectively alleviated and the effectiveness of DA is improved. Extensive experiments demonstrate that our proposed SelectAugment significantly improves various off-the-shelf DA methods on image classification and fine-grained image recognition.

Introduction

Data augmentation (DA) is an effective technique to foster model optimization by improving the amount and diversity of training data, which has been widely used in various tasks, such as image classification (Perez and Wang 2017; Mikołajczyk and Grochowski 2018; Fawzi et al. 2016; Lin et al. 2022), segmentation (Zhao et al. 2019; Ronneberger, Fischer, and Brox 2015), object detection (Montserrat et al. 2017; Zhong et al. 2020), *etc*. There is a series of works that augment samples with label-invariant transforms, *e.g*. random rotation, flipping, erasing(Zhong et al. 2020; DeVries and Taylor 2017), *etc*. Besides, some automatic DA approaches (Cubuk et al. 2018; Lim et al. 2019; Zhang et al. 2020; Ho et al. 2019; Lin et al. 2021) propose to search for more effective augmentation policies from a set of predesigned DA operations.

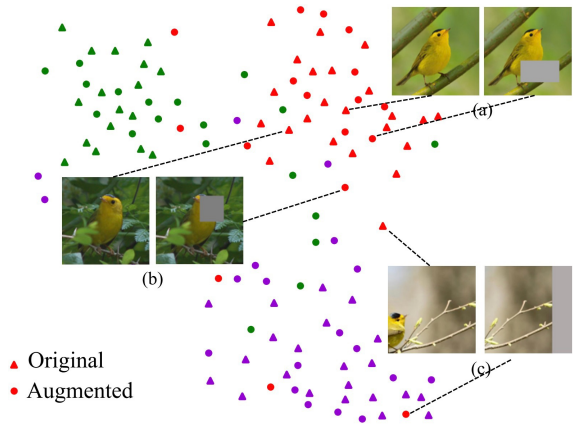

Figure 1: Visualization of feature embeddings of original images and images augmented by AutoAugment (Cubuk et al. 2018), where colors denote classes. It indicates that some samples are unsuitable for augmentation. For example, augmenting (c) pushes it away from the corresponding class center and causes noise and visual ambiguity, leading to the side effects.

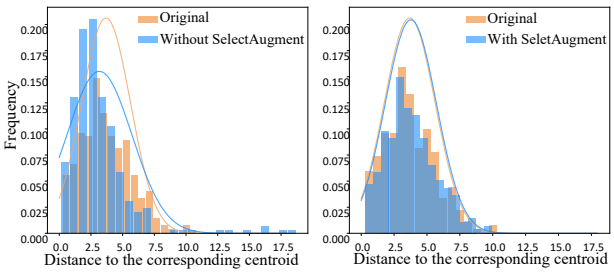

Figure 2: Illustration of the existence of the distribution shift when using DA and the role of our proposed SelectAugment in alleviating this. Here, we compare the histograms of distances to the corresponding centroids between the augmented data without (left) and with (right) SelectAugment.

Prior DA works mentioned above only focus on designing augmentation strategies/operations themselves but ignore the selection of samples suitable to be augmented. As a common practice, each training sample is randomly aug-

*Equal contribution.
†Corresponding Author.

mented with a probability that is usually heuristically set or automatically searched. Fig. 1 illustrates some examples unsuitable for augmentation. For example, augmenting Fig. 1 (c) pushes it away from the corresponding class center and causes excessive noise and visual ambiguity, leading to the distribution shift. We further showcase this by comparing the feature distribution of original data to the augmented data with a representative DA method (*i.e.*, AutoAugment) in the left sub-figure of Fig. 2. This uncontrollable shift on the training data may limit the benefits of DA, even lead to performance deterioration on the test data. Thus, it's of great importance to design a deterministic sample selection strategy for DA, which is still under-explored.

Actually, the problem of data selection for executing DA operations is a complicated one because its optimal policy depends on various factors including the dynamic training status of task-specific networks as well as the contents of samples and various DA operations. It is quite difficult to set a simple and unified criterion to ascertain how likely the side-effects will happen since training images and DA methods are diverse. One straightforward solution choice is to decide whether to augment an image or not in each batch in a deterministic way, which is in line with the decision-making problem in reinforcement learning (RL). However, directly applying RL will suffer from difficult optimization due to the large action space (*i.e.*, 2^b where b denotes the size of mini-batch). To facilitate policy learning, we propose a novel approach named SelectAugment, which exploits the hierarchy of data selection for DA by modeling this task into a two-step decision process. We adopt Hierarchical Reinforcement Learning (HRL) to online learn a parent policy and a child policy for batch-level sample selection and instance-wise sample selection respectively in a deterministic way. Specifically, the parent policy aims to first choose the augmentation ratio for each batch from a ratio pool P according to sample information and the training status of the target network. Then, under this ratio, the child policy performs the sample-level selection for executing DA based on the contents and semantics of samples. Considering the hierarchy, the action space sizes of the parent policy and the child policy are $|P|$ and $C_b^{\lfloor b\times p \rfloor}$, respectively, where $p \in P$ and $C_b^{\lfloor b\times p \rfloor}$ represents the number of combinations of $\lfloor b\times p \rfloor$ selected from b. Easy to find that action spaces are effectively reduced compared to 2^b. With a smaller action space, it's easier for the RL algorithm to find an effective strategy. The parent policy and the child policy in our proposed approach are jointly optimized with the target network (*i.e.*, the mainstream task-specific model) together, which leaves our proposed approach as an online one, obviating the need for re-training the target network after learning the sample selection strategy. The selected images processed by off-the-shelf DA methods are combined together with the remaining images staying original to form a new batch that is fed into the target network for task training. We use the feedback of the target network as the reward signal to train our policy network. As illustrated in the right sub-figure of Fig. 2, our proposed method effectively prevents uncontrollable distribution shift when adopting data augmentation. Besides, extensive experiments demonstrate the compatibility and effectiveness of our sample selection policy which significantly enhances numerous off-the-shelf DA methods, including Mixup (Zhang et al. 2018), CutMix (Yun et al. 2019), AutoAugment (Cubuk et al. 2018) and RandAugment (Cubuk et al. 2020).

In summary, our contributions lie in three aspects:

- We are the first to pinpoint that deterministic sample selection matters in bringing different DA methods into their full play by reducing the side effects caused by the randomness of selecting samples, which is overlooked in prior works;
- We model the sample selection for DA as a two-step decision-making problem delicately and learn the policy via HRL, where we first determine the augmentation ratio at the batch level, then perform a deterministic allocation for executing augmentation operations at the instance level.
- We conduct extensive experiments to demonstrate our proposed method can be generally applicable for various existing DA methods and substantially improve them. Note that this work is in fact complementary to those works devoted to designing DA operations themselves.

Related Works

Data Augmentation

Data augmentation (DA) plays a critical role in deep learning, which can effectively alleviate network overfitting problems. Previous commonly used methods perform simple transformations, such as random rotation and translation (Simard et al. 2003). CutOut (DeVries and Taylor 2017) and its variant (Takahashi, Matsubara, and Uehara 2019; Zhong et al. 2020) which randomly crop regions of an image to form stronger perturbations. Furthermore, recent progress in automated machine learning has begun to study (Cubuk et al. 2018; Lim et al. 2019; Lin et al. 2021; Ho et al. 2019; Li et al. 2020) automatically searching for the optimal transformation policy to relieve human expertise. The pioneer work AutoAugment (Cubuk et al. 2018) first automates the data augmentation design, where DA policies are searched under reinforcement learning. However, AutoAugment repeatedly trains the light-weight proxy network for the evaluation of every candidate DA policies and is subsequently applied to the target large models, which is computationally expensive. To boost the efficiency, PBA (Ho et al. 2019) and Fast AutoAugment (Lim et al. 2019) introduce an efficient population based optimization and a bayesian optimization respectively. DADA (Li et al. 2020) relaxes the optimization problem to be differentiable and uses gradient based optimization to achieve effective DA policy search. RandAugment (Cubuk et al. 2020) reduces the search space and takes the simple grid search to find DA policy to remove the separate search phase. In addition, there are label-perturbing DA methods. Mixup (Zhang et al. 2018) interpolates two training images in both pixel and label space. CutMix (Yun et al. 2019) randomly crops a region of one image and pastes it into another image, mixing labels with the proportion of

two images. SuperMix (Dabouei et al. 2021) further models the problem of mixing augmentation as a supervised task to improve the efficiency of previous DA methods merging images. Co-Mixup (Kim et al. 2021) simultaneously mixes different regions from multiple input data ensuring diversity among the generated mixup examples.

Although these DA methods have achieved impressive results on many tasks, they do not consider whether the samples are suitable for augmentation. Specifically, in these methods, a probability is given for each image to roughly decide whether to augment or not, where the probability is commonly manually designed. For example, the value of probability is a fixed scalar in Mixup. In addition, the probability is a scalar which is increased linearly in CutMix. In particular, AutoAugment automatically searches for optimal augmentation policies including this aforementioned probability. To be specific, AutoAugment generates a tuple (operation, magnitude, probability) and evaluates it with the performance of a small proxy network that is trained from scratch. When retraining the target network, one of the offline policies is randomly selected and performed on the training data to train the target network. Therefore, they all ignore the image contents when using offline policies for DA, and decide whether to augment or not in a stochastic way. Recently, a series of online automatic DA methods (Zhang et al. 2020; Lin et al. 2019, 2021) learns to choose the DA operations based on the sample contents, but still executes them relying on a manually set probability.

In our paper, we propose a deterministic data selection approach to select the samples to execute the DA operations according to their contents and the network training status, which is generally applicable to different tasks to enlarge the benefits of DA. Note that our proposed method targets a different problem (*i.e.*, data selection for executing off-the-shelf DA operations) instead of designing DA operations as prior DA works.

Hierarchical Reinforcement Learning

Hierarchical RL (HRL) decomposes a complex task into several sub-ones with a hierarchical topology to speed up the learning process. Specifically, goal-conditioned HRL (Levy et al. 2017; Nachum et al. 2018; Kulkarni et al. 2016) divides the corresponding task into two steps, which are respectively completed by two graded policies (*i.e.*, the parent and child policies) that are learned simultaneously. Commonly, the parent policy outputs preliminary goals as the guidance for its subordinate child policy in a top-down view. Some early works require manual design of goals (Peng et al. 2017; Cuayáhuitl et al. 2010), while some methods (Florensa et al. 2018; Tang et al. 2018) (including our proposed method) automatically generate goals through the interaction with the environment. According to goals, second-step policy executes actions at a more granular level.

Proposed Method

In this section, we first outline the framework of SelectAugment as well as its core idea, then elaborate on the hierarchical deterministic sample selection policy learning in SelectAugment. Furthermore, we discuss the superiority and an important expansion of our proposed method.

Basics of Reinforcement Learning

We introduce the basics of Reinforcement Learning (RL) in this part. RL commonly model the policy learning problem as a Markov decision process (MDP) represented with $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, T)$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state space and the action space respectively. The RL agent observes the environment state $s \in \mathcal{S}$ and takes an action $a \in \mathcal{A}$ with the policy $\pi(a\,|s) \ : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. Then, the RL agent receives a step-wise reward $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. Accordingly, the environment moves to next state with a transition function denoted as $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$. $\gamma \in (0, 1]$ is a discount factor and T is a time horizon. The objective of policy learning is to learn an optimal policy π^* which can maximize the accumulative reward R over different steps in each episode.

Hierarchical Reinforcement Learning (HRL) aims to decompose a complex task into a hierarchy of several subtasks. Specifically, a two-level HRL commonly learns a parent policy $\pi^P(a^P\,|s^P)$ and a child policy $\pi^C(a^C\,|s^C, a^P)$ corresponding to $MDP^P = (\mathcal{S}^P, \mathcal{A}^P, \mathcal{P}^P, R^P, \gamma, T)$ and $MDP^C = (\mathcal{S}^C, \mathcal{A}^C, \mathcal{P}^C, R^C, \gamma, T)$, respectively. The $\mathcal{A}^P$ denotes the action space of parent policy while the $\mathcal{A}^C$ denotes the child action space. The parent policy outputs a parent action $a^P \in \mathcal{A}^P$, which is taken as the condition for the following decision by the child policy. In this paper, we employ HRL to decide an augmentation ratio at the batch level and select specific samples to be augmented at the instance level.

Hierarchical Deterministic Sample Selection Policy

In our proposed SelectAugment, we formulate the task of data selection as a two-step decision-making problem and adopt Hierarchical Reinforcement Learning (HRL) to search for the optimal policy. We detail our design below.

Parent Policy Modeling The parent policy aims to determine the proportion of the augmented images in each batch. As illustrated in Fig. 3, we adopt RL to learn the policy towards this objective, and model the *state* and *action* for the policy learning of this level as below.

Parent state. As illustrated in Fig. 3, we take the current batch and the status variables of the target network together as the parent state vector $\boldsymbol{s^P}$. Specifically, $\boldsymbol{s^P}$ is required to represent not only the characteristics of the training batch, the training status of target network, but also the responses of the target network on the current batch.

To achieve this, the parent state vector $\boldsymbol{s^P}$ encodes three different categories of information in our design: 1) Encoding the information about the data itself, *i.e.*, labels (denoted by $\boldsymbol{y}$) of data, which represents the coarse semantics of the current batch. 2) Encoding the information about the target network, including the training iteration number and the recorded average historical training loss of target network. Here, we choose such variables to reflect the training status following (Fan et al. 2018; Kumar, Packer, and Koller 2010; Jiang et al. 2014) where the effectiveness of such design has been experimentally demonstrated. 3) Encoding the

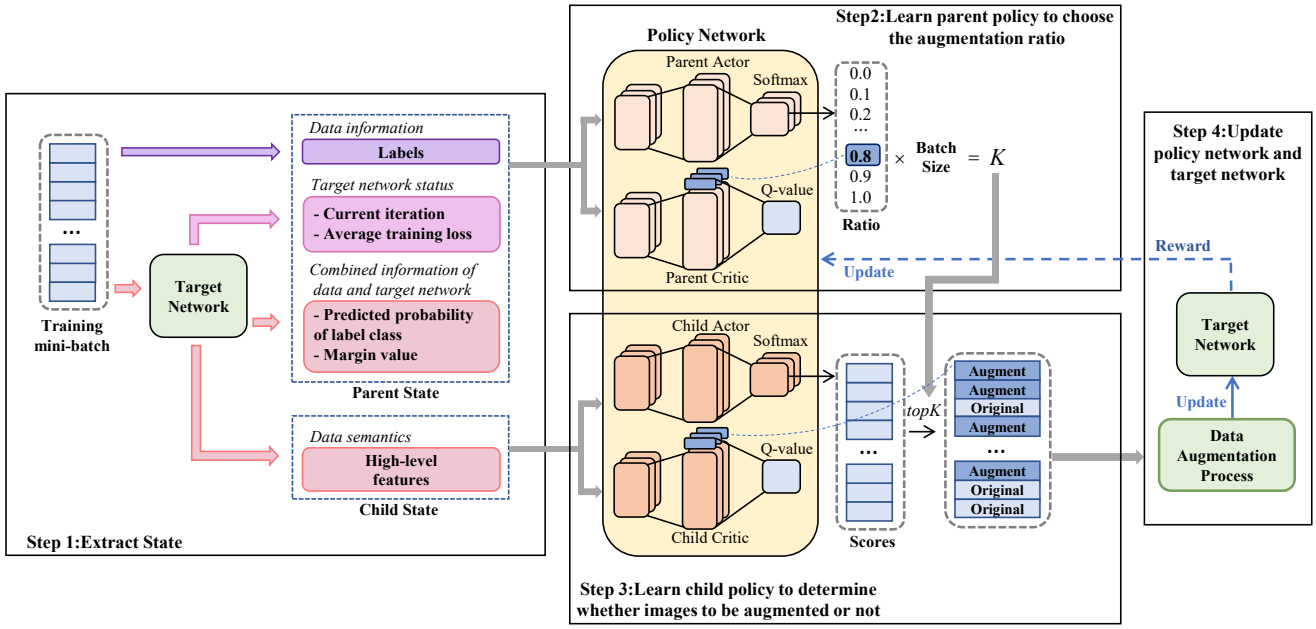

Figure 3: Pipeline of our proposed SelectAugment. To alleviate the side effects caused by randomly choosing images for augmentation, we aim to provide a deterministic sample selection for existing DA methods according to the content of images as well as the training status of the target network, based on reinforcement learning algorithm.

feedback of the target network on the current batch, including the predicted probability of label class as well as a margin value proposed in (Cortes, Mohri, and Rostamizadeh 2013) with the definition of $P(\boldsymbol{y}|\boldsymbol{x}) - \max_{\boldsymbol{y}' \neq \boldsymbol{y}} P(\boldsymbol{y}'|\boldsymbol{x})$.

Parent action. The parent policy determines what proportion of samples in the current batch to be augmented by choosing the augmentation ratio from the ratio pool. We set up the augmentation ratio pool (*i.e.*, the action space of parent agent) as $\mathcal{A}^P = \{0.0, 0.1, 0.2, ..., 0.9, 1.0\}$. Here, $a^P = 0$ represents *non-augmentation* in the sense that all images in the batch are original images, while $a^P = 1$ denotes *fully-augmentation* where all images in the batch are augmented.

Child Policy Modeling The child policy aims to determine whether each image in a batch is augmented or not under the learned ratio by parent policy, as shown in Fig. 3. This means that we make an instance-level decision on sample selection based on their contents with the child policy.

Child state. The child policy needs to perceive the sample contents for finding the most suitable samples to be augmented under the augmentation ratio inferred by parent policy. Therefore, we propose to utilize the deep features of images extracted by the target network as child state $\boldsymbol{s^C}$.

Child action. Different from the parent policy that determines the augmentation ratio in a coarse way, the child policy aims to make a precise decision on whether each image is augmented. Here, we define child action $\boldsymbol{a^C}$ as a vector whose dimension equals to the batch-size b, which represents the scores of images suitable for augmentation in each batch. Given an augmentation ratio a^P inferred by the parent policy, the number of augmented images K can be determined as $K = \lfloor a^P \times b \rfloor$. Then, child policy selects the samples corresponding to the K-highest scores $topK(\cdot)$ to execute augmentation operations $\psi(\cdot)$ as Eq.(1).

$$\boldsymbol{x_{aug}} = \{\psi(x_i) | a_i^C \in topK(\boldsymbol{a^C}), x_i \in \boldsymbol{x})\}, \quad (1)$$

$$\boldsymbol{x_{ori}} = \{x_j | a_j^C \notin topK(\boldsymbol{a^C}), x_j \in \boldsymbol{x}\}, \quad (2)$$

where $\boldsymbol{x}$ denotes the original training mini-batch (without any augmentation operations) and $\widetilde{\boldsymbol{x}} = \{\boldsymbol{x_{aug}}, \boldsymbol{x_{ori}}\}$ represents the selectively augmented mini-batch processed by SelectAugment.

Reward Function The objective of our proposed data selection policy is to enhance the target network by improving the benefits of DA as possible.

Therefore, in order to be consistent with the objective, the improvement of mini-batch augmented with SelectAugment $\widetilde{\boldsymbol{x}}$, compared to two standards (*i.e.*, the performance of original mini-batch $\boldsymbol{x}$ and fully-augmented mini-batch $\overline{\boldsymbol{x}}$ on the target network $\phi(\cdot)$), is used as the reward to guide the policy learning. Mathematically, the reward function can be formulated as:

$$r = [l(\phi(\boldsymbol{x}), \boldsymbol{y}) - l(\phi(\widetilde{\boldsymbol{x}}), \boldsymbol{y})] + [l(\phi(\overline{\boldsymbol{x}}), \boldsymbol{y}) - l(\phi(\widetilde{\boldsymbol{x}}), \boldsymbol{y})], \quad (3)$$

where $l(\cdot)$ is the loss function of the mainstream task. The parent policy and the child policy serve for the same goal, thus, we adopt the same reward function as the Eq.3.

The improvement of mini-batch augmented with our method, compared to two standards (*i.e.*, the performance of original mini-batch and fully-augmented mini-batch on

the target network), is used as the reward to guide our proposed data selection policy learning. Accordingly, the reward function design is consistent with the objective of our policy which aims at enhancing the target task.

Hierarchical Policy Learning Considering the action spaces of both the parent policy and the child policy are discrete, in this work, we perform policy learning by adopting the widely used Advantage Actor-Critic (A2C) algorithm (Mnih et al. 2016; Zhang et al. 2019) where the actor network is to learn a discrete control policy $\pi(a\,|s)$ while the critic network aims to estimate the value of state $V^{\pi}(s)$. Here, we reformulate it appropriately under our task scenario. Similar to (Mnih et al. 2016), we model the value of each state-action pair with a Q-value, which is formulated as:

$$Q^{\pi}(a^P, \boldsymbol{a^C}, \boldsymbol{s^P}, \boldsymbol{s^C}) = \mathbb{E}_{\pi}[r\,|\mathcal{A}=(a^P, \boldsymbol{a^C}), \mathcal{S}=(\boldsymbol{s^P}, \boldsymbol{s^C})\,]. \tag{4}$$

The model for the parent policy learning comprises a parent actor network and a parent critic network as shown in Fig.3. We use θ_P and φ_P to denote their corresponding trainable network parameters, respectively. Here, we define the advantage function of parent policy for updating θ_P and φ_P:

$$A_P(a^P, \boldsymbol{s^P}) = Q^{\pi}(a^P, \boldsymbol{a^C}, \boldsymbol{s^P}, \boldsymbol{s^C}) - V^{\pi_{\theta_P}}_{\varphi_P}(\boldsymbol{s^P}). \tag{5}$$

Specifically, we take the square value of the advantage function as the loss function to update the φ_P:

$$L(\varphi_P) = \big(A_P(a^P, \boldsymbol{s^P})\big)^2. \tag{6}$$

Moreover, the loss function for updating the θ_P is:

$$L(\theta_P) = -\log \pi_{\theta_P}(a^P|\boldsymbol{s^P}) A_P(a^P, \boldsymbol{s^P}). \tag{7}$$

Similarly, the model for the child policy learning comprises a child actor (with parameters θ_C) and a child critic (with parameters φ_C). The advantage function of the child policy and the loss function for updating the φ_C are defined as:

$$A_C(\boldsymbol{a^C}, \boldsymbol{s^C}) = Q^{\pi}(a^P, \boldsymbol{a^C}, \boldsymbol{s^P}, \boldsymbol{s^C}) - V^{\pi_{\theta_C}}_{\varphi_C}(\boldsymbol{s^C}), \tag{8}$$

$$L(\varphi_C) = \big(A_C(\boldsymbol{a^C}, \boldsymbol{s^C})\big)^2. \tag{9}$$

Inspired by top-k policy (Chen et al. 2019a), the loss function used to update θ_C is modified to the following:

$$L(\theta_C) = -\sum\nolimits_{a_i^C \in topK(\boldsymbol{a^C})} \log \pi_{\theta_C}(a_i^C|\boldsymbol{s^C}) A_C(\boldsymbol{a^C}, \boldsymbol{s^C}). \tag{10}$$

The Superiority on the Action Space

The action space of RL refers to the set including all possible actions. For the data selection problem we tackle in this paper, a straightforward solution is to independently determine whether to execute the DA operation or not for each sample in a batch, which makes the size of the action space be 2^b (b is the batch size). A large action commonly renders difficult policy exploration.

Thanks to the hierarchy of our proposed SelectAugment based on HRL, the action space size is effectively reduced.

In our design, the action space size of the parent policy is the size of the ratio pool $|\mathcal{A}^P|$. For the action space of child policy, its size is conditioned on the decision of the parent policy, denoted as $C_b^{\lfloor b \times a^P \rfloor}$ where C_n^m represents the number of combinations of m selected from n. Note that $C_b^{\lfloor b \times a^P \rfloor} \leq C_b^{\frac{b}{2}} < 2^b$. To be more intuitive, we take the batch-size is 256 as an example. Even in the extreme case (*i.e.*, the ratio given by the parent policy is 0.5), the action space of child policy is up to the largest, which still achieves 20 times reduction compared to 2^b.

Therefore, the size of the action space of SelectAugment is effectively reduced by decomposing the data selection problem into a two-step decision processing. As explained in (Zahavy et al. 2018; Wei, Wicke, and Luke 2018), a smaller action space is beneficial to reduce the difficulty of policy exploration. Furthermore, we experimentally demonstrate the effectiveness of the hierarchical policy (see ablation study *only-child*).

Experiments and Results

Experimental Settings For the target network, following (Cubuk et al. 2018; Lim et al. 2019; Lin et al. 2021; Yun et al. 2019), for CIFAR-10 and CIFAR-100, we respectively use Wide-ResNet-28-10 (WRN) (Zagoruyko and Komodakis 2016), ShakeShake(26 2×32d) and ShakeShake (26 2×96d) (Gastaldi 2017) (aliased as SS(32) and SS(96)) as target network. For ImageNet (Deng et al. 2009), ResNet-50 and ResNet-200 (He et al. 2016) are adopted as target models, which are trained from scratch. For fine-grained image classification, we follow the previous works (Du et al. 2020; Chen et al. 2019b) and utilize pre-trained ResNet-50 and ResNet-101 models as the target network. Unless specified otherwise, the input image size is 32×32 for CIFAR while 224×224 for ImageNet, CUB-200-2011 and Stanford Dogs. We set the batch size to 128 for experiments on CIFAR and 1024 for experiments on ImageNet as well as two fine-grained datasets. The hyperparameters for target networks, such as training epochs and the learning rate of target network are the same as previous works (Yun et al. 2019; Zhang et al. 2018; Cubuk et al. 2018) for fair comparison.

Comparison with State-of-the-arts

Our work is actually DA methods agnostic, in the sense that ours is compatible with any specific DA operations. Thus, we apply our adaptive sample selection policy to improve some of the most representative and commonly used DA approaches including Mixup (Zhang et al. 2018), Cutmix (Yun et al. 2019), AutoAugment (Cubuk et al. 2018) and RandAugment (Cubuk et al. 2020). We also compare our SelectAugment with state-of-the-art methods including Cutout (DeVries and Taylor 2017), Co-Mixup (Kim et al. 2021), Super-Mix (Dabouei et al. 2021) and DADA (Li et al. 2020).

Classification Results on CIFAR The results are reported in Table 1, which shows that our proposed SelectAugment delivers consistent improvements in the classification accuracy when applied to different DA methods. We observe that the SelectAugment is a very general data selection tool that can be applied to different network architectures and different off-the-shelf DA methods.

Method	CIFAR-10		CIFAR-100	
	WRN	SS(32)	WRN	SS(96)
Baseline	96.13	96.26	81.20	82.85
Cutout	96.89	97.03	81.62	84.07
DADA	97.25	97.32	82.58	84.94
Co-Mixup	97.29	97.35	83.23	85.11
Super-Mix	97.30	97.38	83.33	85.13
Mixup	97.14	97.12	82.41	84.77
SelectMixup	**97.33**	**97.38**	**83.37**	**85.17**
CutMix	97.24	97.21	83.12	84.97
SelectCutMix	**97.41**	**97.44**	**83.45**	**85.24**
AutoAugment	97.28	97.37	83.08	85.66
SelectAutoAugment	**97.65**	**97.61**	**83.81**	**85.87**
RandAugment	97.26	97.32	83.16	85.53
SelectRandAugment	**97.56**	**97.60**	**83.77**	**85.86**

Table 1: Test accuracy (%) on CIFAR-10 and CIFAR-100. Our proposed method is compatible with any DA methods. We apply our adaptive sample selection policy to some of the most representative and commonly used DA methods. We prefix them with "*Select-*". We also compare our SelectAugment with SOTA methods. Best in bold.

Method	ResNet-50		ResNet-200	
	Top-1	Top-5	Top-1	Top-5
Baseline	76.28	93.05	78.47	94.19
Cutout	76.74	93.31	79.26	94.65
DADA	77.46	93.46	79.43	94.80
Co-Mixup	77.58	93.70	80.05	94.93
Super-Mix	77.64	93.73	80.56	93.48
Mixup	77.01	93.43	79.62	94.83
SelectMixup	**77.84**	**93.78**	**80.42**	**95.26**
CutMix	77.23	93.54	79.92	94.90
SelectCutMix	**78.02**	**93.90**	**80.66**	**95.31**
AutoAugment	77.61	93.82	79.96	95.02
SelectAutoAugment	**78.16**	**93.97**	**80.78**	**95.36**
RandAugment	77.57	93.76	79.91	95.08
SelectRandAugment	**78.08**	**93.84**	**80.81**	**95.22**

Table 2: Validation set Top-1 and Top-5 accuracy (%) on ImageNet.

Classification Results on ImageNet We conduct experiments on larger-scale ImageNet dataset. Table 2 shows our proposed SelectAugment still brings significant improvements in top-1 and top-5 accuracy. This further demonstrates the effectiveness of SelectAugment and shows the consistent benefits for the larger dataset and deeper networks.

Effectiveness on Fine-grained Classification Additionally, we evaluate our proposed method on fine-grained classification. As reported in Table 3, our proposed SelectAugment is also effective in giving full play to the role of data augmentations through the learned deterministic data selection policy on fine-grained image classification.

Ablation Study

The Influence of Reinforcement Learning As mentioned above, due to the diversity of the image content and the dynamic change of the training target network status, it is difficult to use uniform and simple criteria to decide which images to be augmented or not. Therefore, we adopt Reinforcement Learning (RL) as our base algorithm, which is dynamically determined through the interaction of policy network and target network. To intuitively show the rationality and superiority of RL, we compare our method with other policies where an image is decided to be augmented with a probability (see Table 4). The difference between these strategies lies in the setting of probability values. There is a crucial observation that our method outperforms other policies by a significant margin. This demonstrates the effectiveness of RL where images are deterministically determined to be augmented or not.

Method	CUB-200-2011		Stanford Cars	
	R-50	R-101	R-50	R-101
Baseline	85.32	85.73	91.71	93.05
Cutout	85.65	85.98	92.02	93.28
DADA	86.53	87.39	93.06	94.01
Co-Mixup	86.79	88.06	93.09	94.17
Super-Mix	86.92	88.13	93.17	94.12
Mixup	85.94	87.73	92.31	94.03
SelectMixup	**87.01**	**88.25**	**93.25**	**94.33**
CutMix	86.12	87.90	92.30	94.11
SelectCutMix	**87.16**	**88.30**	**93.51**	**94.26**
AutoAugment	86.83	88.22	93.68	94.20
SelectAutoAugment	**87.35**	**88.42**	**94.13**	**94.37**
RandAugment	86.78	88.17	93.62	94.24
SelectRandAugment	**87.21**	**88.38**	**94.02**	**94.32**

Table 3: Test accuracy (%) on fine-grained classification datasets. "*R-*" represents " *ResNet-*".

Method	Policy	Dataset		
		CIFAR-10	CIFAR-100	ImageNet
Baseline		96.13	81.20	76.28
Mixup	all	97.14	82.41	77.01
	random	96.85	82.25	76.86
	fixed	96.88	82.87	76.82
SelectMixup		**97.33**	**83.37**	**77.84**

Table 4: Performance comparisons of SelectAugment using RL and other policies that use a simple probability to roughly control whether an image is augmented or not. "*all*" means that all images are augmented. "*random*" means that each sample is augmented with a probability p sampled from a uniform distribution $U(0, 1)$. '*fixed*" denotes that we randomly augment each sample with a fixed probability. We set $p = 0.8$ which performs the best. We report test accuracy (%) on CIFAR with WRN-28-10 and Top-1 validation accuracy (%) on ImageNet with ResNet-50.

The Influence of Hierarchical Architecture Here, we conduct an ablation study on the hierarchical design of our proposed SelectAugment. As reported in Table 5, our proposed SelectAugment achieves consistent improvements compared to *only-parent* and *only-child* policies. This indicates that the "divide and conquer" idea is more purposeful and effective in selecting data for DA than non-hierarchical

Method	Policy		Dataset	
	Parent	Child	CIFAR-10	ImageNet
Baseline	-	-	96.13	78.28
Mixup	-	-	97.14	79.62
Only-Parent	✔	✗	97.05	77.15
Only-Child	✗	✔	97.21	77.34
SelectMixup	✔	✔	**97.33**	**77.84**

Table 5: The ablation results on the influence of the hierarchical architecture. "*only-parent*" denotes the setting in which we adopt one-level RL to choose the augmentation ratio and then samples are randomly selected in the batch under the ratio. "*only-child*" denotes the setting in which we adopt one-level RL to directly decide whether an image is augmented or not in the batch.

Method	Parameters	Accuracy (%)	GPU hours
Mixup	0M	97.14	6.5
SelectMixup	1.99M	97.33	7.9

Table 6: Comparison of additional parameters, performance and training time on CIFAR-10 under WRN-28-10 between the models equipped with Mixup and SelectMixup.

designs. Specifically, the *only-parent* policy lacks a deterministic decision based on the characteristic of each image, like previous DA works. The *only-child* policy encounters the trouble of the large action space of RL rendering difficult policy exploration.

Complexity Analysis

We analyze the parameters and the time complexity of our proposed SelectAugment. Following the related works in this field (Cubuk et al. 2018; Lim et al. 2019; Zhang et al. 2020), we report the complexity measure results on CIFAR-10 (See Table 6). The parent and child policy network have 0.01M and 1.98M parameters respectively, leading to 1.99M parameter increase in total (less than 6% of the target network, *i.e.*, WRN-28-10 with 36.5M parameters). As for the time consuming, our proposed approach will bring 1.4 GPU hours increase in the training time.

Visualization Analysis

Grad-CAM Visualization ation in a convolutional neural network. As exampled in Fig. 4, we find that after applying SelectAugment to the DA process for training, the target model (for the mainstream task) latches on discriminative cues more precisely and comprehensively. As shown in the first and the second rows of Fig. 4, the visualization results of the target model trained with SelectAugment indicate that it localizes the foreground region with higher spatial precisions than others. Moreover, the third and fourth rows manifest that the model with SelectAugment localizes more class-related cues(Selvaraju et al. 2016).

Visualization of the Learned Augmentation Ratio We further visualize the augmentation ratio of one certain mini-batch, to analyze the policy learning results. As shown in Fig. 5, we observe that 1) the learned augmentation ratio changes with the training status of the target network; 2) the learned policy varies a lot for different DA methods. This indicates that it is necessary to select samples to be augmented considering the status of the target network.

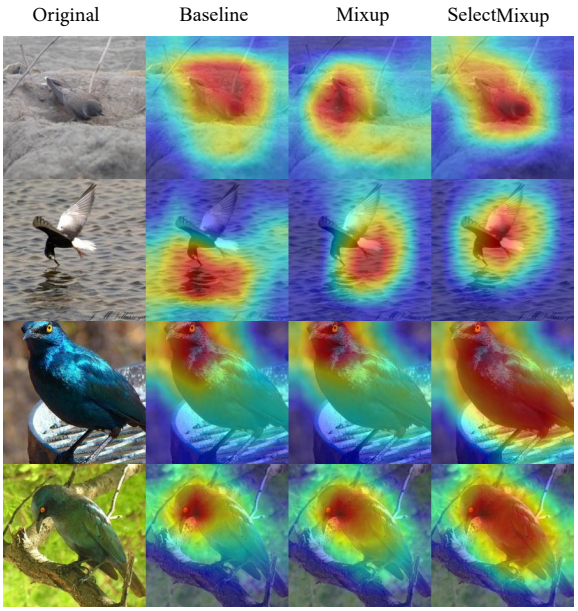

Figure 4: Grad-CAM (Selvaraju et al. 2017) visualization on the examples from CUB-200-2011. The first column shows original images. The remaining columns show the visualization results for the ResNet-50 trained with the baseline methods, Mixup and SelectMixup, respectively.

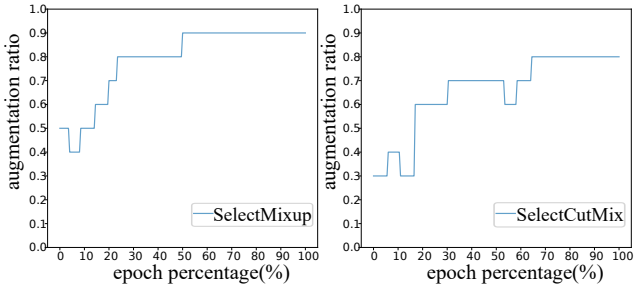

Figure 5: Visualization augmentation ratio given by the parent policy in SelectMixup and SelectCutMix.

Conclusion

In this paper, we point out that randomly selecting samples to do data augmentation may cause content destruction and visual ambiguities, leading to the distribution shift and thus negatively affecting the target model training. To tackle this, we propose a hierarchical deterministic sample selection strategy to give full play to data augmentation. We adopt HRL to facilitate policy learning towards this goal. Our proposed approach is easy to use and generally applicable for different existing DA methods.

Acknowledgements

This work was supported by National Natural Science Foundation of China (NSFC) under Grants U1908209 and 62021001.

References

Chen, M.; Beutel, A.; Covington, P.; Jain, S.; Belletti, F.; and Chi, E. H. 2019a. Top-k off-policy correction for a REINFORCE recommender system. In *WSDM*.

Chen, Y.; Bai, Y.; Zhang, W.; and Mei, T. 2019b. Destruction and construction learning for fine-grained image recognition. In *CVPR*.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2013. Multi-class classification with maximum margin multiple kernel. In *ICML*.

Cuayáhuitl, H.; Renals, S.; Lemon, O.; and Shimodaira, H. 2010. Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Computer Speech & Language*.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR*.

Dabouei, A.; Soleymani, S.; Taherkhani, F.; and Nasrabadi, N. M. 2021. Supermix: Supervising the mixing data augmentation. In *CVPR*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Du, R.; Chang, D.; Bhunia, A. K.; Xie, J.; Song, Y.-Z.; Ma, Z.; and Guo, J. 2020. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. *arXiv preprint arXiv:2003.03836*.

Fan, Y.; Tian, F.; Qin, T.; Li, X.-Y.; and Liu, T.-Y. 2018. Learning to teach. In *ICLR*.

Fawzi, A.; Samulowitz, H.; Turaga, D.; and Frossard, P. 2016. Adaptive data augmentation for image classification. In *2016 IEEE international conference on image processing (ICIP)*.

Florensa, C.; Held, D.; Geng, X.; and Abbeel, P. 2018. Automatic goal generation for reinforcement learning agents. In *ICML*.

Gastaldi, X. 2017. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Ho, D.; Liang, E.; Chen, X.; Stoica, I.; and Abbeel, P. 2019. Population based augmentation: Efficient learning of augmentation policy schedules. In *ICML*.

Jiang, L.; Meng, D.; Mitamura, T.; and Hauptmann, A. G. 2014. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*.

Kim, J.-H.; Choo, W.; Jeong, H.; and Song, H. O. 2021. Co-mixup: Saliency guided joint mixup with supermodular diversity. *ICLR*.

Kulkarni, T. D.; Narasimhan, K. R.; Saeedi, A.; and Tenenbaum, J. B. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *arXiv preprint arXiv:1604.06057*.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-Paced Learning for Latent Variable Models. In *NIPS*.

Levy, A.; Konidaris, G.; Platt, R.; and Saenko, K. 2017. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*.

Li, Y.; Hu, G.; Wang, Y.; Hospedales, T.; Robertson, N. M.; and Yang, Y. 2020. DADA: Differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780*.

Lim, S.; Kim, I.; Kim, T.; Kim, C.; and Kim, S. 2019. Fast autoaugment. *arXiv preprint arXiv:1905.00397*.

Lin, C.; Guo, M.; Li, C.; Yuan, X.; Wu, W.; Yan, J.; Lin, D.; and Ouyang, W. 2019. Online hyper-parameter learning for auto-augmentation strategy. In *ICCV*.

Lin, S.; Yu, T.; Feng, R.; Li, X.; Jin, X.; and Chen, Z. 2021. Local Patch AutoAugment with Multi-Agent Collaboration. *arXiv preprint arXiv:2103.11099*.

Lin, S.; Zhang, Z.; Huang, Z.; Lu, Y.; Lan, C.; Chu, P.; You, Q.; Wang, J.; Liu, Z.; Parulkar, A.; et al. 2022. Deep Frequency Filtering for Domain Generalization. *arXiv preprint arXiv:2203.12198*.

Mikołajczyk, A.; and Grochowski, M. 2018. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *ICML*.

Montserrat, D. M.; Lin, Q.; Allebach, J.; and Delp, E. J. 2017. Training object detection and recognition CNN models using data augmentation. *Electronic Imaging*.

Nachum, O.; Gu, S.; Lee, H.; and Levine, S. 2018. Data-efficient hierarchical reinforcement learning. *arXiv preprint arXiv:1805.08296*.

Peng, B.; Li, X.; Li, L.; Gao, J.; Celikyilmaz, A.; Lee, S.; and Wong, K.-F. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. *arXiv preprint arXiv:1704.03084*.

Perez, L.; and Wang, J. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*.

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

Selvaraju, R. R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; and Batra, D. 2016. Grad-CAM: Why did you say that? *arXiv preprint arXiv:1611.07450*.

Simard, P. Y.; Steinkraus, D.; Platt, J. C.; et al. 2003. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*.

Takahashi, R.; Matsubara, T.; and Uehara, K. 2019. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*.

Tang, D.; Li, X.; Gao, J.; Wang, C.; Li, L.; and Jebara, T. 2018. Subgoal discovery for hierarchical dialogue policy learning. *arXiv preprint arXiv:1804.07855*.

Wei, E.; Wicke, D.; and Luke, S. 2018. Hierarchical approaches for reinforcement learning in parameterized action space. In *AAAI*.

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

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zahavy, T.; Haroush, M.; Merlis, N.; Mankowitz, D. J.; and Mannor, S. 2018. Learn what not to learn: Action elimination with deep reinforcement learning. *arXiv preprint arXiv:1809.02121*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond empirical risk minimization. *ICLR*.

Zhang, X.; Wang, Q.; Zhang, J.; and Zhong, Z. 2020. Adversarial autoaugment. *ICLR*.

Zhang, Z.; Chen, J.; Chen, Z.; and Li, W. 2019. Asynchronous episodic deep deterministic policy gradient: Toward continuous control in computationally complex environments. *IEEE transactions on cybernetics*, 51(2): 604–613.

Zhao, A.; Balakrishnan, G.; Durand, F.; Guttag, J. V.; and Dalca, A. V. 2019. Data augmentation using learned transformations for one-shot medical image segmentation. In *CVPR*.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2020. Random erasing data augmentation. In *AAAI*.