Isotonic Data Augmentation for Knowledge Distillation

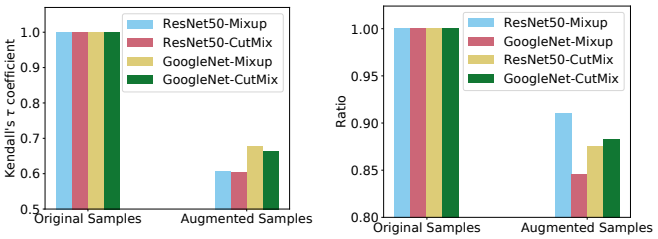
Wanyun Cui[1*], **Sen Yan**[2]
Shanghai University of Finance and Economics
cui.wanyun@sufe.edu.cn, woodthree333@gmail.com,

Abstract

Knowledge distillation uses both real hard labels and soft labels predicted by teacher models as supervision. Intuitively, we expect the soft labels and hard labels to be concordant w.r.t. their orders of probabilities. However, we found *critical order violations* between hard labels and soft labels in augmented samples. For example, for an augmented sample $x = 0.7 * panda + 0.3 * cat$, we expect the order of meaningful soft labels to be $P_{\text{soft}}(panda|x) > P_{\text{soft}}(cat|x) > P_{\text{soft}}(other|x)$. But real soft labels usually violate the order, e.g. $P_{\text{soft}}(tiger|x) > P_{\text{soft}}(panda|x) > P_{\text{soft}}(cat|x)$. We attribute this to the unsatisfactory generalization ability of the teacher, which leads to the prediction error of augmented samples. Empirically, we found the violations are common and injure the knowledge transfer.In this paper, we introduce order restrictions to data augmentation for knowledge distillation, which is denoted as isotonic data augmentation (IDA). We use isotonic regression (IR) – a classic technique from statistics – to eliminate the order violations. We show that IDA can be modeled as a tree-structured IR problem. We thereby adapt the classical IRT-BIN algorithm for optimal solutions with $O(c \log c)$ time complexity, where c is the number of labels. In order to further reduce the time complexity, we also propose a GPU-friendly approximation with linear time complexity. We have verified on variant datasets and data augmentation techniques that our proposed IDA algorithms effectively increases the accuracy of knowledge distillation by eliminating the rank violations.

1 Introduction

Data augmentation, as a widely used technology, is also beneficial to knowledge distillation [Das *et al.*, 2020]. For example, [Wang *et al.*, 2020b] use data augmentation to improve the generalization ability of knowledge distillation. [Wang *et al.*, 2020a] use Mixup [Zhang *et al.*, 2018], a widely applied data augmentation technique, to improve the efficiency of knowledge distillation. In this paper, we focus on the mixture-based data augmentation (e.g. Mixup and Cutmix [Yun *et al.*, 2019]), arguably one of the most widely used type of augmentation techniques.



(a) The Kendall's τ coefficient between the soft label distribution and the hard label distribution. Larger τ means higher ordinal association.

(b) The ratio of augmented samples in which at least one original label is in the top 2 soft labels.

Figure 1: Both 1a and 1b reveal that, the orders of soft labels and hard labels are highly concordant for the original samples. But for augmented samples, the order concordance is broken seriously. This motivates us to introduce the order restrictions in data augmentation for knowledge distillation.

Intuitively, we expect the order concordance between soft labels and hard labels. In Fig. 2, for an augmented sample $\tilde{x} = 0.7 * panda + 0.3 * cat$, the hard label distribution is $P_{\text{hard}}(panda|\tilde{x}) = 0.7 > P_{\text{hard}}(cat|x) = 0.3 > P_{\text{hard}}(other|\tilde{x}) = 0$. Then we expect the soft labels to be monotonic w.r.t. the hard labels: $P_{\text{soft}}(panda|\tilde{x}) > P_{\text{soft}}(cat|\tilde{x}) > P_{\text{soft}}(other|\tilde{x})$.

However, we found critical order violations between hard labels and soft labels in real datasets and teacher models. To verify this, we plot the Kendall's τ coefficient [Kendall, 1938] between the soft labels and the hard labels of different teacher models and different data augmentation techniques in CIFAR-100 in Fig. 1a. We only count label pairs whose orders are known. In other words, we ignore the orders between two "other" labels, since we do not know them. A clear phenomenon is that, although the hard labels and soft labels are almost completely concordant for original samples, they are likely to be discordant for augmented samples. What's more surprising is that, in Fig. 1b, we find that there are a

[*]Contact Author

Figure 2: Using isotonic regression to introduce order restrictions to soft labels.

proportion of augmented samples, in which none of the original labels are in the top 2 of the soft labels.We attribute this to the insufficient generalization ability of the teacher, which leads to the prediction error of the augmented sample. We will show in Sec 5.3 that the order violations will injury the knowledge distillation. As far as we know, the order violations between hard labels and soft labels havn't been studied in previous studies.

A natural direction to tackle the problem is to reduce the order violations in soft labels. To this end, we leverage the isotonic regression (IR) – a classic technique from statistics – to introduce the order restrictions into the soft labels. IR minimizes the distance from given nodes to a set defined by some order constraints. In Fig. 2, by applying order restrictions to soft labels via IR, we obtain concordant soft labels while keeping the original soft label information as much as possible. IR has numerous important applications in statistics [Barlow and Brunk, 1972], operations research [Maxwell and Muckstadt, 1985], and signal processing [Acton and Bovik, 1998]. To our knowledge, we are the first to introduce IR in knowledge distillation.

Some other studies also noticed the erroneous of soft labels in knowledge distillation and were also working on mitigating it [Wen *et al.*, 2019; Ding *et al.*, 2019; Tian *et al.*, 2019]. However, none of them revealed the order violations of soft labels.

2 Related Work

Knowledge Distillation with Erroneous Soft Labels. In recent years, knowledge distillation [Hinton *et al.*, 2015] as a model compression and knowledge transfer technology has received extensive research interests. Since the teacher model is non-optimal, how to deal with the errors of soft labels has become an important issue. Traditional methods often solve this problem via optimizing the teacher model or student model.

For teacher optimization, [Cho and Hariharan, 2019] finds that a larger network is not necessarily a better teacher, because student models may not be able to imitate a large network. They proposed that early-stopping should be used for the teacher, so that large networks can behave more like small networks [Mahsereci *et al.*, 2017], which is easier to imitate. An important idea for teacher model optimization is "strictness" [Yang *et al.*, 2019], which refers to tolerating the teacher's probability distribution outside of hard labels.

The training optimization of the student model is mainly to modify the loss function of distillation. [Wen *et al.*, 2019] proposed to assign different τs to different samples based on their deceptiveness to teacher models. [Ding *et al.*, 2019] proposed that the label correlation represented by student should be consistent with teacher model. They use residual labels to add this goal to the loss function.

However, none of these studies reveal the phenomenon of rank violations in data augmented knowledge distillation.

Data Mixing is a typical data augmentation method. Mixup [Zhang *et al.*, 2018] first randomly combines a pair of samples via weighted sum of their data and labels. Some recent studies include CutMix [Yun *et al.*, 2019], and RICAP [Takahashi *et al.*, 2019]. The main difference among the different mixing methods is how they mix the data.

The difference between our isotonic data augmentation and the conventional data augmentation is that we focus on relieving the error transfer of soft labels in knowledge distillation by introducing order restrictions. Therefore, we pay attention to the order restrictions of the soft labels, instead of directly using the mixed data as data augmentation. We verified in the experiment section that our isotonic data augmentation is more effective than directly using mixed data for knowledge distillation.

3 Data Augmentation for Knowledge Distillation

3.1 Standard Knowledge Distillation

In this paper, we consider the knowledge distillation for multi-class classification. We define the teacher model as $\mathcal{T}(x) : \mathcal{X} \rightarrow \mathbb{R}^c$, where $\mathcal{X}$ is the feature space, $\mathcal{C} = \{1, \cdots, c\}$ is the label space. We denote the student model as $\mathcal{S}(x) : \mathcal{X} \rightarrow \mathbb{R}^c$. The final classification probabilities of the two models are computed by $softmax(\mathcal{T}(x))$ and $softmax(\mathcal{S}(x))$, respectively. We denote the training dataset as $\mathcal{D}_{train} = \{(x^{(1)}, y^{(1)}), \cdots, (x^{(n)}, y^{(n)})\}$, where $y^{(i)}$ is one-hot encoded. We denote the score of the j-th label for $y^{(i)}$ as $y^{(i)}_j$.

The distillation has two objectives: hard loss and soft loss. The hard loss encourages the student model to predict the supervised hard label $y^{(i)}$. The soft loss encourages the student model to perform similarly with the teacher model. We use

cross entropy (CE) to measure both similarities:

$$\mathcal{L}_{\text{hard}}(x,y) = CE(softmax(\mathcal{S}(x)), y)$$
$$\mathcal{L}_{\text{soft}}(x,y) = CE(softmax(\frac{\mathcal{S}(x)}{\tau}), softmax(\frac{\mathcal{T}(x)}{\tau})) \quad (1)$$

where τ is a hyper-parameter denoting the temperature of the distillation.

The overall loss of the knowledge distillation is the sum of the hard loss and soft loss:

$$\mathcal{L}_{\text{KD}} = E_{(x,y)\sim\mathcal{D}_{train}} \alpha\tau^2 \mathcal{L}_{\text{soft}}(x,y) + (1-\alpha)\mathcal{L}_{\text{hard}}(x,y) \quad (2)$$

where α is a hyper-parameter.

3.2 Knowledge Distillation with Augmented Samples

In this subsection, we first formulate data augmentation for knowledge distillation. We train the student model against the augmented samples instead of the original samples from $\mathcal{D}_{train}$. This method is considered as a baseline without introducing the order restrictions. We then formulate the data augmentation techniques used in this paper.

Data Augmentation-base Knowledge Distillation. In this paper, we consider two classic augmentations (i.e., CutMix [Yun *et al.*, 2019] and Mixup [Zhang *et al.*, 2018]). Our work can be easily extended to other mixture-based data enhancement operations (e.g. FCut [Harris *et al.*, 2020], Mosiac [Bochkovskiy *et al.*, 2020]). As in Mixup and CutMix, we combine two original samples to form a new augmented sample. For two original samples $(x^{(i)}, y^{(i)})$ and $(x^{(j)}, y^{(j)})$, data augmentation generates a new sample $(\tilde{x}, \tilde{y})$. The knowledge distillation based on augmented samples has the same loss function as in Eq. (2):

$$\mathcal{L}_{\text{KD-aug}} = E_{(\tilde{x},\tilde{y})\sim\mathcal{D}_{train}} \alpha\tau^2 \mathcal{L}_{soft}(\tilde{x},\tilde{y}) + (1-\alpha)\mathcal{L}_{hard}(\tilde{x},\tilde{y}) \quad (3)$$

where the augmented sample $(\tilde{x}, \tilde{y}) \sim \mathcal{D}_{train}$ is sampled by first randomly selecting 2 original samples $\{(x^{(i)}, y^{(i)}), (x^{(j)}, y^{(j)})\}$ from $\mathcal{D}_{train}$, and then mixing the samples.

We formulate the process of augmenting samples as:

$$\tilde{x} = A(x^{(i)}, x^{(j)}, \gamma)$$
$$\tilde{y} = \gamma y^{(i)} + (1-\gamma) y^{(j)} \quad (4)$$

where A denotes the specific data augmentation technique. $\tilde{y}$ is distributed in two labels (e.g. $P(panda|\tilde{y}) = 0.7, P(cat|\tilde{y}) = 0.3$). We will formulate different data augmentation techniques below.

CutMix augments samples by cutting and pasting patches for a pair of original images. For $x^{(i)}$ and $x^{(j)}$, CutMix samples a patch $B = (r_x, r_y, r_w, r_h)$ for both of them. Then CutMix removes the region B in $x^{(i)}$ and fills it with the patch cropped from B of $x^{(j)}$. We formulate CutMix as:

$$A_{\text{CutMix}}(x^{(i)}, x^{(j)}, \gamma) = M \odot x^{(i)} + (1-M) \odot x^{(j)} \quad (5)$$

where $M \in \{0,1\}^{W\times H}$ indicates whether the coordinates are inside (0) or outside (1) the patch. We follow the settings in [Yun *et al.*, 2019] to uniformly sample r_x and r_y and keep the aspect ratio of B to be proportional to the original image:

$$r_x \sim Unif(0, W), r_w = W\sqrt{1-\gamma}$$
$$r_y \sim Unif(0, H), r_h = W\sqrt{1-\gamma} \quad (6)$$

Mixup augments a pair of sample by a weighted sum of their input features:

$$A_{\text{Mixup}} = \gamma x^{(i)} + (1-\gamma) x^{(j)} \quad (7)$$

where each $\gamma \sim Beta(a, a)$ for $a \in (0, \inf)$.

4 Isotonic Data Augmentation

In this section, we introduce the order restrictions in data augmentation for knowledge distillation, which is denoted as isotonic data augmentation. In Sec 4.1, we analyze the partial order restrictions of soft labels. We propose the new objective of knowledge distillation subjected to the partial order restrictions in Sec 4.2. Since the partial order is a special directed tree, we propose a more efficient Adapted IRT algorithm based on IRT-BIN [Pardalos and Xue, 1999] to calibrate the original soft labels. In Sec 4.3, we directly impose partial order restrictions on the student model. We propose a more efficient approximation objective based on penalty methods.

4.1 Analysis of the Partial Order Restrictions

We hope that the soft label distribution by isotonic data augmentation and the hard label distribution have no order violations. We perform isotonic regression on the original soft labels $\mathcal{T}(\tilde{x})$ to obtain new soft labels that satisfy the order restrictions. We denote the new soft labels as the order restricted soft labels $m(\mathcal{T}(\tilde{x})) \in \mathbb{R}^c$. For simplicity, we will use m to denote $m(\mathcal{T}(\tilde{x}))$. We use m_i to denote the score of the i-th label.

To elaborate how we compute m, without loss of generality, we assume the indices of the two original labels of the augmented sample $(\tilde{x}, \tilde{y})$ are $1, 2$ respectively with $\gamma > 0.5$. So $\tilde{y}$ is monotonically decreasing, i.e. $\tilde{y}_1 = \gamma > \tilde{y}_2 = 1-\gamma > \cdots > \tilde{y}_c$.

We consider two types of order restrictions: (1) the order between two original labels (i.e., $m_1 \geq m_2$); (2) The order between an original label and a non-original label (i.e. $\forall i \in \{1,2\}, j \in \{3, \cdots, c\}, m_i \geq m_j$). For example, in Fig. 2, we expect the probability of *panda* is greater than that of *cat*. And the probability of *cat* is greater than other labels. We do not consider the order between two non-original labels.

We use $G(V, E)$ to denote the partial order restrictions, where each vertex $i = 1 \cdots c$ represents m_i, an edge $(i, j) \in E$ represents the restriction of $m_i \geq m_j$. E is formulated in Eq. (8). We visualize the partial order restrictions in Fig. 3.

$$E = \{(1,2)\} \cup \{(2,i) | i = 3 \cdots c\} \quad (8)$$

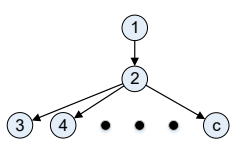


Figure 3: The partial order restrictions is a directed tree.

Lemma 1. *E is a directed tree.*

4.2 Knowledge Distillation via Order Restricted Soft Labels

For an augmented sample $(\tilde{x}, \tilde{y})$, we first use the teacher model to predict its soft labels. Then, we calibrate the soft labels to meet the order restrictions. We use the order-restricted soft label distribution m to supervise the knowledge distillation. We formulate this process below.

Objective with Order Restricted Soft Labels. Given the hard label distribution $\tilde{y}$ and soft label distribution $\mathcal{T}(\tilde{x})$ of an augmented sample $(\tilde{x}, \tilde{y})$, the objective of knowledge distillation with isotonic data augmentation is:

$$\mathcal{L}_{\text{KD-i}} = \mathcal{L}_{\text{KD-aug}} + \beta E_{(\tilde{x},\tilde{y})\sim\mathcal{D}_{train}} CE(\tilde{y}, \hat{m}) \tag{9}$$

where $\hat{m}$ denotes the optimal calibrated soft label distribution.

To compute $\hat{m}$, we calibrate the original soft label $\mathcal{T}(\tilde{x})$ to meet the order restrictions. There are multiple choices for $\hat{m}$ to meet the restrictions. Besides order restrictions, we also hope that the distance between the original soft label distribution $\mathcal{T}(\tilde{x})$ and the calibrated label distribution m is minimized. Intuitively, the original soft labels contain the knowledge of the teacher model. So we want this knowledge to be retained as much as possible. We formulate the calibration below.

We compute $\hat{m}$ which satisfies the order restriction E while preserving most knowledge by minimizing the mean square error to the original soft labels:

$$\hat{m} = \arg\min_{m} mean\_square\_error(\mathcal{T}(\tilde{x}), m) \tag{10a}$$

$$\text{subject to } \forall (i,j) \in E, \hat{m}_i \geq \hat{m}_j \tag{10b}$$

Eq. (10b) denotes the order restrictions. Eq. (10a) denotes the objective of preserving most original information. The goal of computing $\hat{m}$ can be reduced to the classical isotonic regression in statistics.

Here we analyze the difference between isotonic data augmentation and probability calibration in machine learning [Niculescu-Mizil and Caruana, 2005]. Isotonic regression is also applied in probability calibration. While both the proposed isotonic data augmentation and probability calibration try to rectify the erroneous predicted by models, our proposed isotonic data augmentation only happens in the training phase when the groundtruth distribution (i.e. the hard labels) is known. We use the isotonic soft labels $\hat{m}$ as the extra supervision for model training. In contrast, the probability calibration needs to learn an isotonic function and uses it to predict the probability of unlabeled samples.

Algorithm. To optimize $\mathcal{L}_{\text{KD-i}}$, we need to compute $\hat{m}$ first. According to lemma 1, finding the optimal $\hat{m}$ can be reduced to the tree-structured IR problem, which can be solved by IRT-BIN [Pardalos and Xue, 1999] with binomial heap in $O(c \log c)$ time complexity. We notice that the tree structure in our problem is special: a star (nodes $2 \cdots c$) and an extra edge $(1, 2)$. So we give a more efficient implementation compared to IRT-BIN with only one sort in algorithm 1.

The core idea of the algorithm is to iteratively reduce the number of violations by merging node blocks until no order violation exists. Specifically, we divide the nodes into several blocks, and use B_i to denote the block for node i. At initialization, each B_i only contains node i itself. Since all nodes except 1 and 2 are leaf nodes with a common parent 2, we first consider the violations between 2 and $i = 3 \cdots c$ (line 4-7). Note that nodes $i = 3 \cdots c$ are sorted according to their soft probabilities $\mathcal{T}(\tilde{x})_i$. We enumerate $i = 3 \cdots c$ and iteratively determine whether there is a violation between node 2 and node i. If so, we absorb node i into B_2. This absorption will set all nodes in B_2 to their average value. In this way, we ensure that there are no violations among nodes $2 \cdots c$. Then, we consider the order between 1 and 2. If they are discordant (i.e. $m_1 < m_2$), we similarly absorb B_2 into B_1 to eliminate this violation (line 9-11). If this absorption causes further violations between 2 and a leaf node, we similarly absorb the violated node as above (line 12-15). Finally, we recover $\hat{m}$ from m according to the final block divisions.

Algorithm 1 Adapted IRT.

Data: $\mathcal{T}(\tilde{x})$;
1: Initialize $m_i \leftarrow \mathcal{T}(\tilde{x})_i$, $B_i \leftarrow \{i\}$ for $i = 1 \cdots c$
2: Sort m_i for $i = 3 \cdots c$ in descending order
3: $i \leftarrow 3$
4: **while** $i \leq c$ AND $m_2 < m_i$ **do**
5: $\quad m_2 \leftarrow \frac{m_2 \times |B_2| + m_i}{|B_2| + 1}$
6: $\quad B_2 \leftarrow B_2 \cup \{i\}$
7: $\quad i \leftarrow i + 1$
8: **end while**
9: **if** $m_1 < m_2$ **then**
10: $\quad m_1 \leftarrow \frac{m_1 + m_2 \times |B_2|}{|B_2| + 1}$
11: $\quad B_1 \leftarrow B_1 \cup B_2$
12: $\quad$ **while** $i \leq c$ AND $m_1 < m_i$ **do**
13: $\quad\quad m_1 \leftarrow \frac{m_1 \times |B_1| + m_i}{|B_1| + 1}$
14: $\quad\quad B_1 \leftarrow B_1 \cup \{i\}$
15: $\quad\quad i \leftarrow i + 1$
16: $\quad$ **end while**
17: **end if**
18: Recover $\hat{m}$ from m according to B
19: **Return** $\hat{m}$

Theorem 1. *[Pardalos and Xue, 1999] The Adapted IRT algorithm terminates with the optimal solution to $\hat{m}$.*

The correctness of the algorithm is due to the strictly convex function of isotonic data augmentation subject to convex constraints. Therefore it has a unique local minimizer which is also the global minimizer [Bazaraa *et al.*, 2013]. Its time complexity is $O(c \log c)$.

4.3 Efficient Approximation via Penalty Methods

We found two drawbacks of the proposed order restricted data augmentation in Sec 4.2: (1) although the time complexity is $O(c \log c)$, the algorithm is hard to compute in parallel in GPU; (2) The order restrictions are too harsh, which overly distorts information of the original soft labels. For example, if the probability of original labels are very low, then almost all nodes will be absorbed and averaged. This will loss all valid knowledge from the original soft labels. In this subsection, we loose the order restrictions and propose a more GPU-friendly algorithm.

Note that, the partial order E in Eq. (10b) introduces the

restrictions to the soft labels, and then uses the isotonic soft labels to limit the student model. If we directly use the partial order to limit the student model instead, the restrictions can be rewritten as:

$$\begin{aligned} &\forall(i,j) \in E, \mathcal{S}(\tilde{x})_i \geq \mathcal{S}(\tilde{x})_j \\ \Leftrightarrow &\mathcal{S}(\tilde{x})_1 \geq \mathcal{S}(\tilde{x})_2 \text{ AND } min(\mathcal{S}(\tilde{x})_{1,2}) \geq max(\mathcal{S}(\tilde{x})_{3\cdots c}) \end{aligned} \tag{11}$$

Note that we can replace $min(\mathcal{S}(\tilde{x})_{1,2}) \geq max(\mathcal{S}(\tilde{x})_{3\cdots c})$ with a simpler term $\mathcal{S}(\tilde{x})_2 \geq max(\mathcal{S}(\tilde{x})_{3\cdots c})$ without changing the actual restriction. We use $min(\mathcal{S}(\tilde{x})_{1,2}) \geq max(\mathcal{S}(\tilde{x})_{3\cdots c})$ because we want to ensure the loss below is equally sensitive to both $\mathcal{S}(\tilde{x})_1$ and $\mathcal{S}(\tilde{x})_2$.

Objective with Order Restricted Student. We convert the optimization problem subjected to Eq. (11) to the unconstraint case in Eq. (12) via penalty methods. The idea is to add the restrictions in the loss function.

$$\begin{aligned} \mathcal{L}_{\text{KD-p}} = \mathcal{L}_{\text{KD-aug}} + \sigma E_{(\tilde{x},\tilde{y})\sim\mathcal{D}_{train}}[&max(0, \mathcal{S}(\tilde{x})_2 - \mathcal{S}(\tilde{x})_1) \\ &+ max(0, max(\mathcal{S}(\tilde{x})_3 \cdots \mathcal{S}(\tilde{x})_c) - min(\mathcal{S}(\tilde{x})_1, \mathcal{S}(\tilde{x})_2))] \end{aligned} \tag{12}$$

where σ is the penalty coefficients. The penalty-based loss $\mathcal{L}_{\text{KD-p}}$ can be computed in $O(c)$ time and is GPU-friendly (via the max function).

5 Experiments

5.1 Setup

Models. We use teacher models and the student models of different architectures to test the effect of our proposed isotonic data augmentation algorithms for knowledge distillation. We tested the knowledge transfer of the same architecture (e.g. from ResNet101 to ResNet18), and the knowledge transfer between different architectures (e.g. from GoogLeNet to ResNet).

Competitors. We compare the isotonic data augmentation-based knowledge distillation with standard knowledge distillation [Hinton *et al.*, 2015]. We also compare with the baseline of directly distilling with augmented samples without introducing the order restrictions. We use this baseline to verify the effectiveness of the order restrictions.

Datasets. We use *CIFAR-100* [Krizhevsky *et al.*, 2009], which contains 50k training images with 500 images per class and 10k test images. We also use ImageNet, which contains 1.2 million images from 1K classes for training and 50K for validation, to evaluate the scalability of our proposed algorithms.

Implementation Details. For CIFAR-100, we train the teacher model for 200 epochs and select the model with the best accuracy on the validation set. The knowledge distillation is also trained for 200 epochs. We use SGD as the optimizer. We initialize the learning rate as 0.1, and decay it by 0.2 at epoch 60, 120, and 160. By default, we set $\beta = 3, \sigma = 2$, which are derived from grid search in $\{0.5, 1, 2, 3, 4, 5\}$. We set $\tau = 4.5, \alpha = 0.95$ from common practice. For ImageNet, we train the student model for 100 epochs. We use SGD as the optimizer with initial learning rate is 0.1. We decay the learning rate by 0.1 at epoch 30, 60, 90. We also set $\beta = 3, \sigma = 2$. We follow [Matsubara, 2021] to set $\tau = 1.0, \alpha = 0.5$. Models for ImageNet were trained on 4 Nvidia Tesla V100 GPUs. Models for CIFAR-100 were trained on a single Nvidia Tesla V100 GPU.

5.2 Main Results

Results on CIFAR-100. We show the classification accuracies of the standard knowledge distillation and our proposed isotonic data augmentation in Table 1. Our proposed algorithms effectively improve the accuracies compared to the standard knowledge distillation. This finding is applicable to different data augmentation techniques (i.e. CutMix and Mixup) and different network structures. In particular, the accuracy of our algorithms even outperform the teacher models. This shows that by introducing the order restriction, our algorithms effectively calibrate the soft labels and reduce the error from the teacher model. As Mixup usually performs better than CutMix, we only use Mixup as data augmentation in the rest experiments.

Results on ImageNet. We display the experimental results on ImageNet in Table 2. We use the same settings as [Tian *et al.*, 2019], namely using ResNet-34 as the teacher and ResNet-18 as the student. The results show that isotonic data augmentation algorithms are more effective than the original data augmentation technology. This validates the scalability of the isotonic data augmentation.

We found that KD-p is better on CIFAR-100, while KD-i is better on ImageNet. We think this is because ImageNet has more categories (i.e. 1000), which makes order violations more likely to appear. Therefore, strict isotonic regression in KD-i is required to eliminate order violations. On the other hand, since CIFAR-100 has fewer categories, the original soft labels are more accurate. Therefore, introducing loose restrictions through KD-i is enough. As a result, we suggest to use KD-i if severe order violation occurs.

Ablation. In Table 1, we also compare with the conventional data augmentation without introducing order restrictions (i.e. KD-aug). It can be seen that by introducing the order restriction, our proposed isotonic data augmentation consistently outperforms the conventional data augmentation. This verifies the advantages of our isotonic data augmentation over the original data augmentation.

5.3 Effect of Order Restrictions

Our basic intuition of this paper is that, order violations of soft labels will injure the knowledge distillation. In order to verify this intuition more directly, we evaluated the performance of knowledge distillation under different levels of order violations. Specifically, we use the Adapted IRT algorithm to eliminate the order violations of soft labels for $0\%, 25\%, \cdots, 100\%$ augmented samples, respectively. We show in Fig. 4 the effectiveness of eliminating different proportions of order violations in CIFAR-100. As more violations are calibrated, the accuracy of knowledge distillation continues to increase. This verifies that the order violations injure the knowledge distillation.

5.4 Efficiency of Isotonic Data Augmentation

We mentioned that KD-p based on penalty methods is more efficient and GPU-friendly than KD-i. In this subsection, we verified the efficiency of different algorithms. We selected

	ResNet101 ResNet18	ResNet50 ResNet18	ResNext50 ResNet18	GoogleNet ResNet18	DenseNet121 ResNet18	SeResNet101 ResNet18	SeResNet101 SeResNet18	DenseNet121 SeResNet18	Avg.
Teacher	78.28	78.85	78.98	78.31	78.84	78.08	78.08	78.84	
Student	77.55	77.55	77.55	77.55	77.55	77.55	77.21	77.21	
KD	79.78	79.41	79.88	79.33	79.84	79.41	77.45	79.65	79.34
(KD Mixup)KD-aug	79.39	79.75	80.14	80.15	79.75	78.35	78.94	79.52	79.50
(KD Mixup)KD-i	79.75	80.13	80.35	80.25	**80.38**	79.73	78.83	80.01	79.93
(KD Mixup)KD-p	**80.56**	**80.45**	**80.67**	**80.35**	80.36	**80.11**	**79.25**	**80.49**	**80.28**
(KD CutMix)KD-aug	79.73	80.02	80.19	79.71	79.77	79.19	78.55	80.23	79.67
(KD CutMix)KD-i	**79.95**	80.02	**80.67**	**79.98**	**80.27**	79.51	79.05	80.45	79.99
(KD CutMix)KD-p	79.93	**80.51**	80.34	79.96	79.98	**79.57**	**79.13**	**80.83**	**80.03**
CRD	79.76	79.75	79.59	79.74	79.74	79.22	79.35	79.86	79.63
(CRD Mixup)CRD-aug	79.52	79.38	80.03	79.92	80.05	79.69	79.41	80.43	79.81
(CRD Mixup)CRD-i	**79.97**	**79.84**	**80.49**	80.01	80.15	79.45	79.77	80.47	80.01
(CRD Mixup)CRD-p	79.91	79.82	80.04	**80.16**	**81.03**	**79.93**	**80.19**	**80.65**	**80.21**
(CRD CutMix)CRD-aug	79.77	79.63	79.96	80.13	80.18	79.17	79.49	80.37	79.84
(CRD CutMix)CRD-i	**80.04**	80.14	**80.62**	**80.37**	**80.59**	79.56	79.51	**80.52**	**80.17**
(CRD CutMix)CRD-p	79.91	**80.19**	80.11	80.28	**80.59**	**79.77**	**80.01**	80.48	**80.17**

Table 1: Results of CIFAR-100. KD means standard knowledge distillation [Hinton *et al.*, 2015] and CRD means contrastive representation distillation [Tian *et al.*, 2019]. $*-$aug means knowledge distillation using mixup-based data augmentation without calibrating the soft labels, $* - i$ means soft labels by isotonic regression and $* - p$ means soft labels by the efficient approximation.

	KD-aug	KD-i	KD-p
top-1/top-5	68.79/88.24	**69.71/89.85**	69.04/88.93

Table 2: Results of ImageNet.

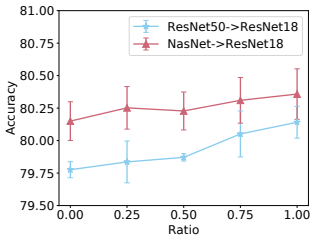


Figure 4: Effect of introducing order restrictions to different ratios of samples. Average over 5 runs. Restricting more samples will improve the effect.

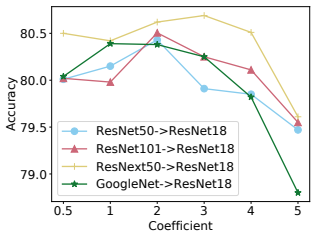


Figure 5: Effect of different σs. $\sigma = 2$ is a recommended value as it outperforms other values in most cases.

	KD	KD-aug	KD-i	KD-p
Mixup	1.00x	1.02x	3.33x	1.02x
CutMix	1.00x	1.01x	3.05x	1.01x

Table 3: Time costs for different data augmentation algorithms.

models from Table 1 and counted their average training time of one epoch. In Table 3, taking the time required for standard KD as the unit 1, we show the time of different data augmentation algorithms. It can be seen that KD-p based on penalty methods require almost no additional time. This shows that KD-p is more suitable for large scale data in terms of efficiency.

5.5 Effect of the Looseness of Order Restrictions

The coefficient σ in the Eq. (12) is the key hyper-parameter that controls the looseness of KD-p. It can be found that for most models, the model performs best when $\sigma = 2.0$. Therefore, $\sigma = 2$ is a recommended value for real tasks.

	SST	TREC	DBPedia
KD-aug	97.35	99.72	98.54
KD-i	97.85	99.78	98.95
KD-p	**98.24**	**99.95**	**99.01**

Table 4: Results on several NLP tasks.

5.6 Effect on NLP Tasks

Our proposed algorithm can also be extended to NLP tasks and Table 4 shows the results on several NLP tasks including SST [Socher *et al.*, 2013], TREC [Li and Roth, 2002] and DBPedia[Auer *et al.*, 2007]. We use Bert[Devlin *et al.*, 2019] as the teacher and DistilBert[Sanh *et al.*, 2019] as the student. We leverage the mixup method in Mixup-Transformer[Sun *et al.*, 2020], and the results indicate that comparing to KD-aug, KD-i and KD-p will improve student models' accuracy.

6 Conclusion

We reveal that the conventional data augmentation techniques for knowledge distillation have critical order violations. In this paper, we use isotonic regression (IR) - a classic statistical algorithm - to eliminate the rank violations. We adapt the traditional IRT-BIN algorithm to the adapted IRT algorithm to generate concordant soft labels for augmented samples. We further propose a GPU-friendly penalty-based algorithm. We have conducted a variety of experiments in different datasets with different data augmentation techniques and verified the effectiveness of our proposed isotonic data augmentation algorithms. We also directly verified the effect of introducing rank restrictions on data augmentation-based knowledge distillation.

Acknowledgements

This paper was supported by National Natural Science Foundation of China (No. 61906116), by Shanghai Sailing Program (No. 19YF1414700).

References

[Acton and Bovik, 1998] Scott T Acton and Alan C Bovik. Nonlinear image estimation using piecewise and local image models. *TIP*, 7(7):979–991, 1998.

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[Barlow and Brunk, 1972] Richard E Barlow and Hugh D Brunk. The isotonic regression problem and its dual. *JASA*, 67(337):140–147, 1972.

[Bazaraa *et al.*, 2013] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[Bochkovskiy *et al.*, 2020] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[Cho and Hariharan, 2019] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *ICCV*, pages 4794–4802, 2019.

[Das *et al.*, 2020] Deepan Das, Haley Massa, Abhimanyu Kulkarni, and Theodoros Rekatsinas. An empirical analysis of the impact of data augmentation on knowledge distillation. *arXiv preprint arXiv:2006.03810*, 2020.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[Ding *et al.*, 2019] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Shu-Tao Xia. Adaptive regularization of labels. *arXiv preprint arXiv:1908.05474*, 2019.

[Harris *et al.*, 2020] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, and Adam Prügel-Bennett Jonathon Hare. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, 2(3):4, 2020.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Kendall, 1938] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Li and Roth, 2002] Xin Li and Dan Roth. Learning question classifiers. In *COLING*, 2002.

[Mahsereci *et al.*, 2017] Maren Mahsereci, Lukas Balles, Christoph Lassner, and Philipp Hennig. Early stopping without a validation set. *arXiv preprint arXiv:1703.09580*, 2017.

[Matsubara, 2021] Yoshitomo Matsubara. torchdistill: A modular, configuration-driven framework for knowledge distillation. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 24–44, 2021.

[Maxwell and Muckstadt, 1985] William L Maxwell and John A Muckstadt. Establishing consistent and realistic reorder intervals in production-distribution systems. *OR*, 33(6):1316–1341, 1985.

[Niculescu-Mizil and Caruana, 2005] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, pages 625–632, 2005.

[Pardalos and Xue, 1999] Panos M Pardalos and Guoliang Xue. Algorithms for a class of isotonic regression problems. *Algorithmica*, 23(3):211–222, 1999.

[Sanh *et al.*, 2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642, 2013.

[Sun *et al.*, 2020] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. Mixup-transfomer: Dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*, 2020.

[Takahashi *et al.*, 2019] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Data augmentation using random image cropping and patching for deep cnns. *TCSVT*, 2019.

[Tian *et al.*, 2019] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2019.

[Wang *et al.*, 2020a] Dongdong Wang, Yandong Li, Liqiang Wang, and Boqing Gong. Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model. In *CVPR*, pages 1498–1507, 2020.

[Wang *et al.*, 2020b] Huan Wang, Suhas Lohit, Michael Jones, and Yun Fu. Knowledge distillation thrives on data augmentation. *arXiv preprint arXiv:2012.02909*, 2020.

[Wen *et al.*, 2019] Tiancheng Wen, Shenqi Lai, and Xueming Qian. Preparing lessons: Improve knowledge distillation with better supervision. *arXiv preprint arXiv:1911.07471*, 2019.

[Yang *et al.*, 2019] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan L Yuille. Training deep neural networks in generations: A more tolerant teacher educates better students. In *AAAI*, volume 33, pages 5628–5635, 2019.

[Yun *et al.*, 2019] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019.

[Zhang *et al.*, 2018] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.