Prime-Aware Adaptive Distillation

Youcai Zhang[⋆], Zhonghao Lan[2], Yuchen Dai[1], Fangao Zeng[1], Yan Bai[3], Jie Chang[1], and Yichen Wei[1]✉

[1] Megvii Inc. {daiyuchen,zengfangao,changjie,weiyichen}@megvii.com
[2] University of Science and Technology of China, lans@mail.ustc.edu.cn
[3] Tongji University, yan.bai@tongji.edu.cn
yczhang12@fudan.edu.cn

Abstract. Knowledge distillation(KD) aims to improve the performance of a student network by mimicing the knowledge from a powerful teacher network. Existing methods focus on studying what knowledge should be transferred and treat all samples equally during training. This paper introduces the adaptive sample weighting to KD. We discover that previous effective hard mining methods are not appropriate for distillation. Furthermore, we propose Prime-Aware Adaptive Distillation (PAD) by the incorporation of uncertainty learning. PAD perceives the prime samples in distillation and then emphasizes their effect adaptively. PAD is fundamentally different from and would refine existing methods with the innovative view of unequal training. For this reason, PAD is versatile and has been applied in various tasks including classification, metric learning, and object detection. With ten teacher-student combinations on six datasets, PAD promotes the performance of existing distillation methods and outperforms recent state-of-the-art methods.

Keywords: Knowledge Distillation, Adaptive Weighting, Uncertainty Learning

1 Introduction

Obtaining highly accurate and lightweight (small model size and low computation) deep neural networks is crucial for practical applications. Knowledge Distillation (KD) methods [13,28,40,24,32,6] are effective for this purpose and have been extensively explored in the past few years. KD aims to transfer the knowledge from a high-capacity *teacher* network with a huge amount of parameters and heavy computation to a relatively lightweight *student* network. By mimicking the "behavior" of teacher, the student model achieves better performance than as is trained from scratch.

Most knowledge distillation methods focus on studying *what knowledge* should be transferred, *e.g.*, soft predictions [13], feature maps [28], activation-based attention maps [40] or relation between samples [24], etc. However, we notice that in the current literature it is rarely studied that *which samples* contribute more to learning the student model and should be focused.

[⋆] This work was done when Youcai worked at Megvii Inc. Research Shanghai.

This work, for the first time, studies the problem of adaptive sample weighting in knowledge distillation. First, we note that sample weighting has been widely used in other tasks, such as object detection [20,22] and metric learning [12,37]. In these tasks, hard samples are assigned larger weights during training as they are more important to learn the model. These samples are termed “prime samples” in a recent work [3]. However, such “hard mining” approaches are not appropriate in knowledge distillation. As validated in experiments, we found that hard samples have detrimental effect on learning the student model. We conjecture that the mismatch between student’s and teacher’s capacities makes student less capable of learning teacher’s knowledge in these hard samples. Thus, we propose that sample weighting should be biased towards these easy samples.

Inspired by previous sample weighting approaches, we first introduce a few baseline weighting methods. They allocate weights based on the pre-defined function of sample’s loss contribution. Samples with small loss are given large weights. However, these methods suffer from the sensitive hyper-parameters and the poor ability to distinguish prime samples.

To overcome the drawbacks, we propose a Prime-aware Adaptive Distillation (PAD) method. By modelling knowledge distillation with data uncertainty, each sample is modeled by a Gaussian distribution. The mean of the distribution is the prediction from the student and the variance measures the uncertainty about distilling the knowledge from the teacher. According to the estimated uncertainty from the network, PAD perceives the prime samples corrupted with weak noise in a more feasible and reasonable manner than the baseline methods.

In principle, our approach is fundamentally different from and would complement most (if not all) distillation methods. For this reason, PAD is a versatile method. It is applied in various tasks including classification, metric learning, and detection. It is validated to outperform our baseline weighting methods, and boost the performance of previous distillation methods, achieving new state-of-the-art methods on six datasets.

To summarize, this work makes three contributions.

1. This is the first systematic study on the problem of sample weighting for knowledge distillation. In particular, we point out that previous hard mining approaches are not appropriate for our problem.
2. We propose a simple yet effective distillation method (PAD). PAD helps the student network to perceive the prime samples in distillation by modelling the knowledge distillation with data uncertainty.
3. Comprehensive experiments validate that our approach is effective and establishes the new state-of-the-arts, on a vast range of datasets, baselines and previous distillation methods.

2 Related Work

Knowledge Distillation(KD). Hinton et al. [13] are pioneers in the field of knowledge distillation. They adopted the KL-divergence to penalize the softer probability distribution over classes between the teacher and student. From then on, many seminal studies [28,40,6,32,39] have sprung up. For example, an information-theoretic framework [1] was proposed by maximizing the mutual information between the teacher and the student networks. The correlation between instances was also proved as the valuable knowledge by [24,25].

Besides studies on classification task, DarkRank [5] was proposed to perform distillation in metric learning via cross sample similarities transfer. ROI mimic distillation [21] and fine-grained feature imitation [34] were designed for detection tasks by distilling the regions of interest.

Adaptive Sample Weighting. Adaptive sample weighting by adjusting the contributions of samples for training is a well-studied topic in computer vision. Hard-mining is a typical technique which plays a critical role in one-stage detectors [20,22] and metric learning [12,37]. Hard-mining methods reduce the relative loss for easy samples, putting more focus on hard ones. In contrast, Huber loss and smooth l_1 loss reduce the contribution of hard samples by down-weighting the loss of them. Many researchers[14,41] extended this idea to face recognition for noisy-robust feature learning. Cao et al. [3] paid more attention to prime samples to achieve high detection performance. Our work is the first exploration about adaptive sample weighting in distillation, to our best knowledge.

Uncertainty Estimation. Uncertainty estimation is an effective technique to promote the robustness and interpretability of discriminant deep neural networks [23,15,16]. In deep uncertainty learning, there are mainly two types of uncertainty. Model uncertainty estimates the noise of the parameters in deep neural networks, and data uncertainty measures the noise caused by input data. Predictions of neural networks are unreliable when the input sample is out of the training distribution or corrupted by noise. Uncertainty estimation has attracted much attention in computer vision tasks, *e.g.*, face recognition [30], semantic segmentation [15,16], object detection [7,17] and person re-identification(re-ID) [38]. This paper models knowledge distillation with uncertainty estimation to predict the prime samples in distillation.

3 Introducing Sample Weighting to KD

3.1 A General Formulation of KD

Most of the distillation studies obey a paradigm, whether they are in the task of classification, metric learning or object detection. The paradigm is to make students regress to the knowledge obtained from the teachers. The knowledge can be soft predictions [13], embedding feature [21], feature maps of intermediate layer [28], or activation-based attention maps [40]. Given the input $x_i \in X$, the regression aims to find the approximation function $f_s(\cdot)$ for student, where $\boldsymbol{f}_s(x_i)$ should be close to the knowledge $\boldsymbol{y}_i \in Y$ derived from the teacher.

A typical distillation loss can be formulated as

$$\mathcal{L}_{distill} = \frac{1}{N}\sum_{i=1}^{N} w_i d[\boldsymbol{f}_s(x_i), \boldsymbol{y}_i]. \tag{1}$$

N is the number of samples in a batch. $d[\cdot,\cdot]$ is the distance metric to measure the gap between the student and the teacher. L_2 is in common use as the metric. If there is a mismatch of dimension between $\boldsymbol{f}_s(x_i)$ and $\boldsymbol{y}_i$, a simple projection function will be added for dimension conversion, following FitNet [28]. For notation convenience, we abbreviate the gap $d[\boldsymbol{f}_s(x_i), \boldsymbol{y}_i]$ between teacher and student for sample x_i to $d(x_i)$. Noted w_i is the weight to measure the importance of x_i to the overall loss. w_i is set to be same for all samples in conventional knowledge distillation methods.

To guide the learning of student network, distillation loss should be combined with the original task-specific loss $\mathcal{L}_{task}$. The overall loss is formulated as

$$\mathcal{L} = \mathcal{L}_{task} + \lambda\mathcal{L}_{distill}, \tag{2}$$

where λ is a hyper-parameter to balance the task loss and distillation loss.

3.2 Sample Weighting and A Few Baselines

Conventional distillation methods treat all samples equally, without considering the difference among them. Hard sample mining is a well-studied sample weighting method in metric learning [37] and object detection [22]. The loss contribution usually reflects the level of sample difficulty [22,37]. The greater the loss is, the more difficult the sample is. Therefore, weighting samples according to their loss contribution is common practice in hard example mining.

Following [37], we also conduct hard sample mining by increasing the weights of samples with large gaps(d) in distillation. Table 1 shows hard-mining is conversely worse than treating samples equally in distillation. Thus, we argue that hard samples are not prime ones in distillation. For further verification, we design a simple weighting scheme, discarding hard samples with the largest distillation gaps(d) in each batch during training. We find that training without the hardest samples slightly improves the baselines. Combining the hard-mining and hard-discarding results, we can come to the conclusion that hard samples have detrimental effect on the training of the student model.

Table 1. Comparisons of hard-mining (training with large weights for hard samples) and hard-discarding (training without the hardest samples). "↓" means worse than baseline. Experimental settings will be introduced in detail in Sec. 5.

Method	CIFAR100	TinyImageNet	Duke-reID	CUB-200	Pascal VOC07
all samples equally	72.85	63.08	79.5	58.6	45.53
hard-mining	72.12↓	62.57↓	79.0↓	55.7↓	45.44↓
hard-discarding	72.91	63.16	79.8	58.3↓	45.94

Besides the above simple either-or method, we continue to design better weighting schemes. The weighting function w can be formulated as $w_i = w(d(x_i))$. $w(\cdot)$ is a monotone increasing function instead of a decreasing one in hard example mining. We design two soft weighting schemes, exponential weighting and polynomial weighting [37], to allocate each sample a continuous weight. Exponential weighting is defined as

$$w_{\textit{soft-exp}}(d(x_i)) = \frac{e^{-d(x_i)/T}}{\sum_{i=1}^{N} e^{-d(x_i)/T}}, \tag{3}$$

where $T > 0$ controls the power of the suppression against the outliers. And polynomial weighting function can be formulated as

$$w_{\textit{soft-poly}}(d(x_i)) = \frac{(1 + d(x_i))^{-\alpha}}{\sum_{i=1}^{N}(1 + d(x_i))^{-\alpha}}, \tag{4}$$

where $\alpha > 0$ is a coefficient for adjusting the weight distribution.

4 Prime-Aware Adaptive Distillation

Besides easy sample mining in distillation, there also exist problems in introducing sample weighting to distillation. The performance of the above baselines is not satisfactory and sensitive to the additional hyper-parameters as shown in Table 6. In light of this, we propose a novel Prime-aware Adaptive Distillation (PAD) method by modeling knowledge distillation with data uncertainty. As data uncertainty can capture the noise inherent in the observations, PAD can perceive the prime samples with little noise and weight them adaptively.

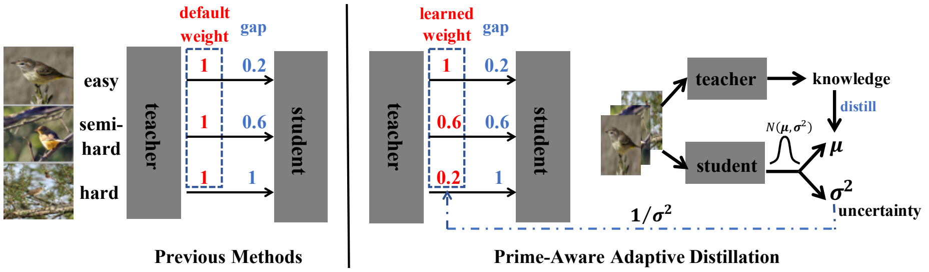

Fig. 1. Previous methods ignore the quality of different samples and allocate the same weight to them. The proposed method perceives the prime samples (usually with small gaps) and enhances them by given large weights, *i.e.*, Prime-aware Adaptive Distillation (PAD). Each sample is modeled by a Gaussian distribution $N(\mu, \sigma^2)$. The mean μ is the prediction of student and the variance σ^2 measures the uncertainty about distilling the knowledge from the teacher. Prime samples are assigned large weights ($1/\sigma^2$), *i.e.*, small variance (σ^2), indicating that the student is confident about their knowledge.

Modelling KD with data uncertainty. As discussed in Sec. 3.1, KD can be viewed as a regression task. In most real-world scenarios, paired input-target data contains noise. The target values $\boldsymbol{f}_s(x_i)$ is inevitably corrupted by input-dependent noise $\boldsymbol{n}(x_i)$. The observed data can be modeled by $\boldsymbol{y}_i = \boldsymbol{f}_s(x_i) + \boldsymbol{n}(x_i)$, where the additive noise $\boldsymbol{n}(x_i)$ can be viewed as errors that move the targets away from their true values $\boldsymbol{f}_s(x_i)$ to their observed values $\boldsymbol{y}_i$.

Supposing the noise obeys Gaussian distribution as $\boldsymbol{n}(x_i) \sim N(0, \boldsymbol{\sigma}^2(x_i))$, where $\boldsymbol{\sigma}^2$ is the variance of the noise. The target probability distribution $\boldsymbol{y}_i$ for input x_i based on the least-square regression can be formulated as

$$p(\boldsymbol{y}_i|x_i) = \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}_i^2}} \exp\left(-\frac{(\boldsymbol{f}_s(x_i) - \boldsymbol{y}_i)^2}{2\boldsymbol{\sigma}_i^2}\right), \tag{5}$$

where $\boldsymbol{f}_s(x_i)$ corresponds to the mean $\boldsymbol{\mu}_i$ of this distribution and $\boldsymbol{\sigma}_i^2$ is the variance to measure the uncertainty of the predicted value $\boldsymbol{f}_s(x_i)$.

We should maximize the log likelihood for the input $\boldsymbol{x}_i$ over the observation $\boldsymbol{y}_i$, so the negative log likelihood is formulated as

$$-\ln p\left(\boldsymbol{y}_i|x_i\right) = \frac{(\boldsymbol{f}_s(x_i) - \boldsymbol{y}_i)^2}{2\boldsymbol{\sigma}_i^2} + \frac{1}{2}\ln\boldsymbol{\sigma}_i^2 + \frac{1}{2}\ln 2\pi. \tag{6}$$

Based on the above derivation, we can formulate a prime-aware distillation loss to help the student network perceive sample difficulty, as:

$$\mathcal{L}_{PAD} = \sum_{i=1}^{N}(\frac{(\boldsymbol{f}_s(x_i) - \boldsymbol{y}_i)^2}{\boldsymbol{\sigma}_i^2} + \ln\boldsymbol{\sigma}_i^2). \tag{7}$$

Without loss of generality, we take the embedding feature as the distillation target for simplicity. It is easy to generalize to other forms of targets, *e.g.*, by flattening feature map to a vector, we can treat feature map as the feature. Due to the least-square assumption, Eq. (7) can be directly applied to L_2-based methods, *e.g.*, L_2 on the feature, FitNet [28], AT [40], and many detection distillation methods [21,34].

For the application of PAD, the only modification is the addition of a variance branch parallel with $\boldsymbol{f}_s(x)$ to estimate the variance $\boldsymbol{\sigma}$. And $\boldsymbol{\mu}$ can be viewed as a drop-in replacement of $\boldsymbol{f}_s(x)$ as shown in Fig. 1. Using an auxiliary branch to estimate the variance is a common practice and has been proved to be effective in data uncertainty [23,15,16,38]. The motivation of an auxiliary branch is its provision of the network with the ability to measure the uncertainty. It should also be pointed out that the variance branch only exists in training, thus it will not bring about any extra computational cost for inference. We also find that simple design for the variance branch achieves satisfactory results and the variance branch has almost no effect on training efficiency.

Discussion. In this part, we analyze how uncertainty effects the distillation from the perspective of loss function. First, Eq. (7) indicates the effect from samples with large variances is weakened in network training. And then, a question comes out: what kind of sample will be given a large variance? $ln(\boldsymbol{\sigma}^2)$ can be regarded as a regularization term, which prevents the model predicting large variances for all samples. Simultaneously, model will not produce small variances for all samples, which will make the first term of Eq. (7) increase rapidly. The model therefore allocates large variances to samples with large gaps(d) to reduce the overall loss of Eq. (7). Samples with large gaps(d) are usually difficult for student. That explains why they are termed hard samples. The rationality of the above hypothesis is proved by experimental analysis in Sec. 5.4.

To sum up, $\frac{1}{\sigma^2}$ can be viewed as a weight coefficient. PAD actually provides a mechanism of adaptive weighting different training samples. Easy samples are highlighted and hard samples are restricted under the influence of uncertainty.

Comparison with sample weighting baselines. PAD obtains the weight directly from the network instead of a conversion from the distance by a predefined function. The weights learned from PAD are more accurate in describing sample difficulty than those from baselines, which can be proved by Table 7 and Fig. 3. Besides, PAD introduces no extra hyper-parameters that need to be tuned carefully in those baseline methods. For this reason, PAD exhibits a versatile generalization on various methods and tasks.

Comparison with VID [1]. Though the formula of VID and PAD is similar, the difference is minor but essential. VID is one of the existing distillation works which treats samples equally. As a contrast, PAD is motivated by the "adaptive sample weighting" (*i.e.*, unequal training), which is well-studied in CV community but missing in distillation. Our study is the first work to exploit adaptive sample weighting in distillation and proposes PAD as an effective weighting way. This difference between PAD and VID results in the different derivation of loss and the meaning of variance. PAD is derived from the data uncertainty, while VID is formulated by the information theory. Also, the variance of PAD has a clear meaning, which indicates the strong correlation between variance and sample quality. Last but not least, PAD performs better than VID with a remarkable margin as shown in Table 2.

5 Experiment

In this section, we firstly apply the proposed PAD on three visual tasks: image classification, metric learning, and object detection. Our experiments exhibit two advantages over others. 1) **Stronger baselines**: student baselines in our experiments are obviously higher than those in recent studies [24,32,34], which will be elaborated in the following part. 2) **More extensive experiments on more challenging tasks**: we conduct experiments with ten teacher-student combinations on various tasks, including ImageNet, where it is hard to achieve positive results for distillation [40]. As a comparison, RKD [24] uses five teacher-student combinations and lacks of experiments on large-scale datasets. Also, we compare different methods on detection, whereas almost all the related works [34,21,10,4] only report their methods without any comparison.

Secondly, comprehensive analysis is given to delve into the learning process of PAD. Finally, we conduct ablation experiments to compare different unequal-learning schemes. All the experiments are conducted on Pytorch. For all the datasets, we follow the train/test splits suggested as popular papers. We carefully tune the weight of distillation loss λ and use the SGD optimizer with a momentum of 0.9 for all the experiments.

5.1 Classification

Firstly, we evaluate PAD on the image classification task where most knowledge distillation methods report their performance.

Dataset settings. CIFAR100 [18], TinyImageNet [19], and ImageNet [8] are adopted. CIFAR100 contains 50K training images with 500 images per class and 10K test images. TinyImageNet has 200 classes, each with 500 training images and 50 validation images. ImageNet [8] provides 1.2 million images from 1K classes for training and 50K for validation.

Implementation details. Following original papers or popular implementations, we apply L_2, RKD [24], and CC [25] on the last embedding layer before classification, apply the FitNet [2] on the last two blocks of CNN, and apply AT [40] and VID [1] on the last four blocks. We re-implement the HKD [13], FitNet [2], and L_2 based on the original paper. For RKD, CC, and AT, we use author-provided codes. For VID, we use the author-verified code from [32].

For CIFAR100 and TinyImageNet, we run each model for 200 epochs with a batch size of 128, and the weight decay of 5e-4. We set the initial learning rate to 0.1, dropping 0.2x at 60, 120, 160 epoch. For ImageNet, we run each model for 90 epochs with a batch size of 512, and set the initial learning rate to 0.2, dropping 0.1x at 30, 60 epoch. The weight decay is set to 1e-4. When combined with L_2 and AT, PAD adopts a fully connected(FC) layer, followed by a batch normalization(BN) layer to generate variance. When combined with FitNet, PAD adopts a 1×1 convolutional layer followed by a BN layer.

Results on CIFAR100 and TinyImageNet. We adopt the ResNet18 [11] as the teacher network, MobileNetV2 [29] as the student network. We report the top-1 test accuracy on CIFAR100 and TinyImageNet datasets in Table 2.

Table 2. Top-1 accuracy (%) on CIFAR100 and TinyImageNet

	teacher	student	HKD	RKD	CC	VID	AT	PAD-AT	L_2	PAD-L_2	FitNet	PAD-FitNet
CIFAR100	75.86	68.16	70.29	68.34	70.0	68.2	69.06	**69.92**	72.85	**74.06**	71.74	**73.45**
tinyImageNet	63.46	56.16	59.52	55.88	57.14	57.06	58.24	**59.26**	63.08	**65.64**	63.34	**64.20**

Results of baseline and compared methods by our implementation are reliable, as they are almost the same as those of [39], and are higher than those of [32]. Weobserve that PAD promotes the performance of three existing classic knowledge distillation methods(AT [40], L_2, and FitNet [28]) with significant margins. For example, PAD obtains a 1.71% gain for FitNet on CIFAR100 and 2.56% for L_2 on TinyImagenet. PAD combined L_2 outperforms other methods. We also find that relation-level distillation methods (CC [25] and RKD [24]) are not superior to the instance-level methods (HKD [13], FitNet [28], and L_2) on these two classification tasks. VID [1] also performs worse than HKD [13]. Similar observations can be found in [32]. Compared with VID [1], our method obtains significant improvement, which shows the effectiveness of allocating different weights to different samples instead of a same weight.

Results on ImageNet. For fair comparisons with AT [40] and CRD [32], we adopt the models from them, ResNet34 as the teacher and ResNet18 as the student. As shown in Table 3, L_2 performs better than FitNet and AT on ImageNet, so we just apply PAD to L_2. It is worth mentioning that L_2 on the last embedding layer is an effective method for distillation, but we do not find its previous use in classification. The gap of top-1 accuracy between the teacher and student is 3.56%. Our PAD-L_2 method reduces this gap by 1.96%, ahead of the state-of-the-art CRD with a margin of 0.54%. Results on ImageNet indicate that our method is generically applicable in the large-scale classification task.

Table 3. Top-1 accuracy (%) on ImageNet. "*" means the result from the paper.

teacher	student	HKD[13]	FitNet[28]	CC[25]	AT[40]	CRD*[32]	L_2	PAD-L_2
73.31	69.75	70.80	70.62	70.74	70.43	71.17	70.90	**71.71**

5.2 Metric Learning

Secondly, we demonstrate the effectiveness of PAD on metric learning.

Dataset settings. We consider two typical metric learning tasks, *i.e.*, fine-grained image retrieval on CUB-200-2011 [33] and person re-ID on DukeMTMC-reID [27]. CUB-200-2011 contains 200 different classes of birds. We use the first 100 classes with 5,864 images for training and the last 100 classes with 5,924 images for testing. $R@1$ is adopted as the evaluation metric. DukeMTMC-reID is a subset of the DukeMTMC dataset designed for person re-ID. It consists of 36,411 human bounding boxes belonging to 1,404 identities. The training set contains 16,522 images of 702 identities and the rest 702 identities are assigned to the testing set. $R@1$ and mean accuracy precision(mAP) are adopted as metrics.

Implementation details. Embedding feature is directly used for retrieval in metric learning, thus, we choose the embedding feature as the distillation target layer for all methods. There are few works designed specifically for metric learning. We choose typical L_2, Darkrank [5] and RKD [24] as the compared methods. Two losses are proposed in Darkrank [5], *i.e.*, HardRank and SoftRank loss. We adopt the HardRank loss as it is computationally efficient and also comparable to the SoftRank loss. For RKD [24], we apply both RKD-D and RKD-A with a weight 1:2 on the feature without normalization. The paper [24] suggests that the student should be trained purely by the RKD loss, *i.e.*, removing the task loss. We report results of the student trained with and without task loss.

For CUB-200-2011, we train for 30,000 steps with a batch size of 60 (12 classes and each class 5 samples), weight decay of 4e-5. The initial learning rate is 1e-3, and is divided by 10 every 10,000 steps. For DukeMTMC-reID, the batch size is 36 (12 classes and each class 3 samples) and the initial learning is 0.005 decaying once at 20,000 step. Other settings are same to CUB-200-2011. The variance branch adopts a FC layer followed by BN.

Results on CUB-200-2011. Inspired by RKD, we conduct both compression distillation and self-distillation on CUB-200-2011 with a strong baseline. Compression distillation means distillation to a smaller network and self-distillation means the teacher and student share the same architecture. For the teacher, we adopt the GoogLeNet [31] trained by multi-similarity loss [35], which provides a higher R@1 of 64.7 than 61.2 in RKD paper. A strong teacher is beneficial to explore the real performance of self-distillation. For the student in compression distillation, we choose ResNet18 [11] with multi-similarity loss, which also provides a higher R@1 of 55.6 than 53.9 in RKD paper.

Table 4 shows the results of different distillation methods on CUB-200-2011. RKD achieves promising results and PAD outperforms the Darkrank and RKD. Self-distillation further improves the performance of the initial teachers. Although our reproduced results (64.7) are inferior to those reported in the paper (65.7) [35], self-distillation by Darkrank, RKD and our PAD outperform the initial teacher and exceeded the reported results of 65.7. The proposed method brings a 1.6% gain. Similarly to RKD, the repeated self-distillation does not provide additional benefits.

Results on DukeMTMC-reID. For person re-ID, we also choose a strong baseline. The teacher is the ResNet50 trained with both softmax and triplet loss, which gives a gain of 2.1% R@1 than only softmax. Multi-similarity loss gives no additional gain than triplet loss in our experiments. Table 4 shows distillation results of different students with different methods. As we see, improvement obtained by distillation is slight when the student baseline is strong. While the improvement becomes remarkable, when the student is trained only by triplet loss. The proposed PAD outperforms RKD and DarkRank based on different baselines. Besides, PAD demonstrates excellent performance in self-distillation compared with other methods.

Table 4. Results on metric learning tasks. w/o task indicates students are trained without task loss, only by RKD. w/ task indicates students are trained with both task loss and RKD.

	DukeMTMC-reID						CUB-200-2011	
	compression distillation				self-distillation		compression distillation	self-distillation
	softmax+triplet		only triplet		softmax+triplet		multi-similarity	multi-similarity
Method	R@1	mAP	R@1	mAP	R@1	mAP	R@1	R@1
teacher	84.9	71.9	84.9	71.9	84.9	71.9	64.7	64.7
student	79.3	61.1	63.6	43.5	84.9	71.9	55.6	64.7
DarkRank	79.9	62.2	69.9	50.9	85.2	72.2	60.1	65.2
RKD w/ task	68.3	48.9	68.3	48.9	73.1	55.3	60.7	65.8
RKD w/o task	80.3	63.2	70.9	52.3	85.1	73.1	60.7	63.0
L_2	79.5	62.5	72.6	53.7	86.0	72.8	58.6	64.4
PAD-L_2	**81.0**	**63.3**	**77.5**	**60.5**	**87.3**	**74.3**	**61.4**	**66.3**

5.3 Object Detection

Thirdly, we validate the proposed method on object detection which is a fundamental and more challenging task in computer vision.

Dataset settings. We conduct experiments on Pascal VOC dataset [9] using both two-stage (Faster-RCNN [26]) and one-stage (RetinaNet [22]) frameworks. Following ROI Mimic [21], we use VOC2007 trainval set of 5k images and VOC2012 trainval set of 16k images as training data. And we evaluate our method on VOC2007 test set of 5K images. Besides mAP@0.5 that considers one generous threshold of IoU $\geqslant 0.5$, we adopt the overall $mmAP$ metric, averaging over the 10 IoU thresholds.

Implementation details. All the detection experiments including baselines are implemented on Detectron2 [36], which provides the strongest baselines for popular object detection frameworks. For example, Detectron2's ResNet50 based Faster R-CNN achieves a mAP@0.5 of 80.9 while the fine-grained paper [34] reports that of only 69.1. All the hyper-parameters related to the object detection are consistent with the standard configurations provided by Detectron2. We only tune the parameters of the distillation part.

For Faster R-CNN, the variance branch consists of two 3x3 convolutional layers and two FC layers, generating variance for each ROI extracted by the sampler. For RetinaNet, variance branch consists of five convolutional layers for the whole feature map and then the spatial mask generated in Fine-frained's way [34] filters out the background. Since FPN is adopted in all experiments, distilled features are sampled from all FPN layers.

Results on Pascal VOC07. Following fine-grained [34], we adopt two settings of backbones, *i.e.*, from ResNet101 to ResNet50 and from VGG16 to VGG11. Table 5 summaries the results of different distillation methods based on different architectures. We have two observations from Table 5. First, PAD is applied to the two most mainstream frameworks and consistently improves the performance of students based on different backbones. Second, the improvement brought by distillation is not as impressive as the original paper [21,34]. We argue that we should analyze the improvement from a relative rather than absolute perspective when baselines become strong. For example, in first column of Table 5, ROI mimic [21] improves the $mmAP$ of student from 54.00% to 55.52%. This gain covers 68% of the gap between teacher and student. Our PAD not only further narrows the gap but also makes the student exceed the teacher slightly, which is a rare phenomenon in detection distillation. Similarly, Fine-grained method covers 54.40% and 59.25% of the $mmAP$ gaps using two backbones respectively based on RetinaNet. Combined with PAD, the gaps decrease by 90.67% and 93.58% and the performance of students become very close to those of teachers. Results using two typical distillation methods based on Faster-RCNN and RetinaNet show that PAD still performs well on detection.

Table 5. Results on Pascal VOC07 with different backbones. ResNet101-50 refers to the backbone of teacher and student respectively. Results of ROI-mimic based on RetinaNet are missing, as ROI-mimic can only be applied to two-stage frameworks.

	Faster R-CNN				RetinaNet			
	ResNet101-50		VGG16-11		ResNet101-50		VGG16-11	
Method	$mmAP$	mAP@0.5	$mmAP$	mAP@0.5	$mmAP$	mAP@0.5	$mmAP$	mAP@0.5
teacher	56.22	82.53	46.53	76.86	57.45	82.09	47.18	74.34
student	54.00	81.98	42.18	72.99	55.52	81.21	43.13	69.57
ROI mimic [21]	55.52	82.25	45.00	75.04	-	-	-	-
PAD-ROI-mimic	**56.41**	**82.46**	**45.62**	**75.79**	-	-	-	-
Fine-grained [34]	54.91	81.93	44.60	74.59	56.57	81.46	45.53	71.99
PAD-Fine-grained	**55.39**	**82.29**	**45.21**	**75.17**	**57.27**	**81.94**	**46.92**	**73.18**

5.4 Analysis

Finally, we give visualization analysis to show the mechanism and effectiveness of PAD. Also, we compare different weighting schemes and conduct ablation experiments. Baselines in this section are students distilled with L_2.

How PAD Affects Distillation? We analyze how PAD affects the student model training in this part. From Fig. 2, we realize that there is a positive correlation between the gap and the variance. From Fig. 3, we observe that the quality of samples decreases with the increase of the learned variance. The above two observations are consistent with our analysis in the method part. With the help of PAD, hard samples are assigned small weights while easy samples are assigned large weights. After adaptive weighting samples by PAD, the effect of hard samples is obviously weakened, while the effect of easy ones is strengthened as the solid lines in Fig. 2 show. Also, PAD narrows the gaps between teacher and student, which means student trained with PAD achieves a better result in regression to the teacher than the baseline.

Impact of the hyper-parameters. For sample weighting baselines, extra hyper-parameters are introduced. As shown in Table 6, soft-weighting schemes slightly improve the performance of distillation, and good results rely heavily on the careful selection of parameters.

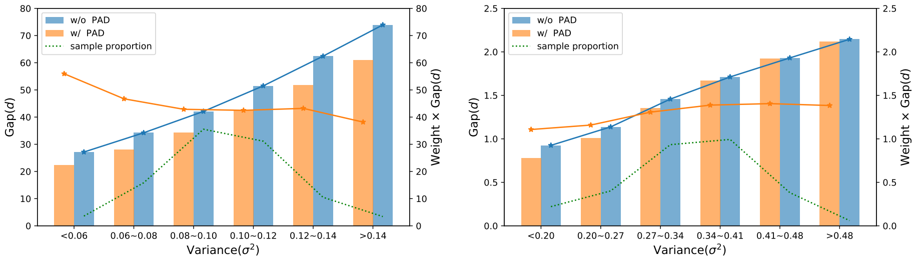

Fig. 2. Relationship between the variance(σ^2) and gap(d) on the training set of TinyImageNet(*left*) and CUB-200-2011(*right*). Bars refer to the gap(d) between teacher and student for different samples, and x-axis is the corresponding variance bins of these samples. Green dotted lines indicate the sample proportion. Solid lines demonstrate the actual effect ($Weight \times Gap$) of different samples on distillation. The effect of samples with large variances is obviously weakened. Note that the absolute effect of different methods should not be compared, as we tune the distillation loss weight λ for all methods.

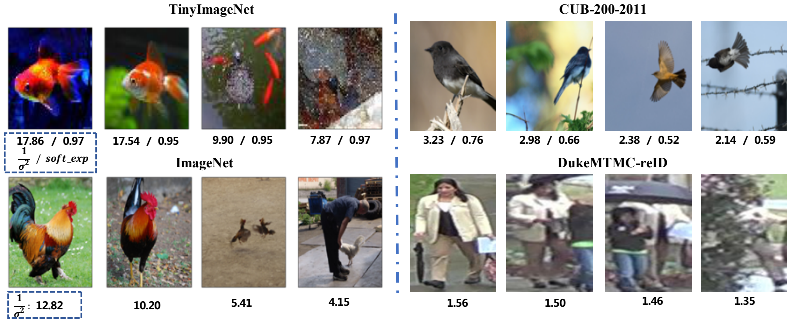

Fig. 3. Examples of the same class or the same person from four datasets. Generally, the larger the variance(σ^2) is, the more difficult the sample is. Thus, PAD assigns large weights ($1/\sigma^2$) to the prime samples. Weights from soft-exp are not always accurate. For example, soft-exp give large weights to the right hard samples.

Table 6. Effect of hyper-parameters (α and T) in sample weighting baselines. "↑" means better than baselines. Few experiments are better than baselines.

		baseline	0.1	0.5	1	2	5	10
soft-exp	TinyImageNet	63.1	61.2	63.2↑	62.6	63.4↑	63.6↑	61.8
	CUB-200-2011	58.6	54.9	58.6	60.1↑	59.2↑	55.8	55.3
soft-poly	TinyImageNet	63.1	62.6	63.0	62.6	63.5↑	62.3	61.8
	CUB-200-2011	58.6	59.4↑	60.4↑	57.9	57.5	59.5↑	60.3↑

Comparisons of different weighting schemes. We compare different weighting schemes and draw two conclusions from Table 7. First, assigning different weights to different samples is effective in distillation even using a manual way. Soft-weighting schemes perform better than the hard-discarding schemes. Second, the weights learned from the off-the-shelf PAD models can better describe the importance of samples than other schemes. Besides the quantitative indicators in Table 7, we see the difference of PAD and soft-exp from the qualitative results in the first row of Fig. 3.

Distillation loss warm up. We also discover that variances become small at the beginning and then stable during the training process with PAD loss. This leads to an increase of weight for all samples at the beginning. To explore this phenomenon, we design a *distillation loss warm up* experiment, *i.e.*, increasing the weight of distillation loss from 0 to the default weight linearly during the initial epochs. We observe that distillation loss warm up also performs slightly better than the baseline but worse than PAD, which means both increasing the sample weight during training and assigning different samples different weights are effective techniques in distillation. PAD easily combines these two techniques and exhibits as a powerful approach to complement knowledge distillation.

Table 7. Comparisons PAD with other weighting schemes and warm up. Top-1 accuracy on CIFAR100 and TinyImageNet. *R@1* on DukeMTMC-reID and CUB-200-2011. *mmAP* on Pascal VOC07.

Method	CIFAR100	TinyImageNet	Duke-reID	CUB-200	Pascal VOC07
all samples equally	72.85	63.08	79.5	58.6	45.53
soft-exp	73.54	63.64	80.2	60.1	46.23
soft-poly	73.38	63.50	79.7	60.4	46.25
weights from PAD	73.72	65.54	80.6	60.2	46.52
warm up	73.13	63.94	80.4	59.0	46.05
PAD	74.06	65.64	81.0	61.4	46.92

6 Conclusion

This paper explores adaptive sample weighting in knowledge distillation, which is innovative and effective for distillation. With more attention paid to easy samples, simple weighting schemes promote the performance of knowledge distillation. Moreover, Prime-Aware Adaptive Distillation (PAD) is proposed for further improvement. PAD is seamlessly combined with existing methods to refine them by enhancing the perceived prime samples. We hope the new state-of-the-arts established by PAD can serve as a starting point for future research.

Acknowledgements. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700800. Thanks Xiruo Tang for her help on paper writing.

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9163–9171 (2019)
2. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: Advances in neural information processing systems. pp. 2654–2662 (2014)
3. Cao, Y., Chen, K., Loy, C.C., Lin, D.: Prime sample attention in object detection. arXiv preprint arXiv:1904.04821 (2019)
4. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Advances in Neural Information Processing Systems. pp. 742–751 (2017)
5. Chen, Y., Wang, N., Zhang, Z.: Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
6. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4794–4802 (2019)
7. Choi, J., Chun, D., Kim, H., Lee, H.J.: Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 502–511 (2019)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
10. Gao, M., Shen, Y., Li, Q., Yan, J., Wan, L., Lin, D., Loy, C.C., Tang, X.: An embarrassingly simple approach for knowledge distillation. arXiv preprint arXiv:1812.01819 (2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
14. Hu, W., Huang, Y., Zhang, F., Li, R.: Noise-tolerant paradigm for training face recognition cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11887–11896 (2019)
15. Isobe, S., Arai, S.: Deep convolutional encoder-decoder network with model uncertainty for semantic segmentation. In: 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA). pp. 365–370. IEEE (2017)
16. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in neural information processing systems. pp. 5574–5584 (2017)
17. Kraus, F., Dietmayer, K.: Uncertainty estimation in one-stage object detection. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). pp. 53–60. IEEE (2019)

18. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
19. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
20. Li, B., Liu, Y., Wang, X.: Gradient harmonized single-stage detector. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8577–8584 (2019)
21. Li, Q., Jin, S., Yan, J.: Mimicking very efficient network for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6356–6364 (2017)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
23. Nix, D.A., Weigend, A.S.: Estimating the mean and variance of the target probability distribution. In: Proceedings of 1994 ieee international conference on neural networks (ICNN'94). vol. 1, pp. 55–60. IEEE (1994)
24. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3967–3976 (2019)
25. Peng, B., Jin, X., Liu, J., Li, D., Wu, Y., Liu, Y., Zhou, S., Zhang, Z.: Correlation congruence for knowledge distillation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5007–5016 (2019)
26. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
27. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision. pp. 17–35. Springer (2016)
28. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
29. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
30. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6902–6911 (2019)
31. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
32. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. arXiv preprint arXiv:1910.10699 (2019)
33. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
34. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4933–4942 (2019)
35. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5022–5030 (2019)
36. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2 (2019)
37. Yu, R., Dou, Z., Bai, S., Zhang, Z., Xu, Y., Bai, X.: Hard-aware point-to-set deep metric for person re-identification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 188–204 (2018)

38. Yu, T., Li, D., Yang, Y., Hospedales, T.M., Xiang, T.: Robust person reidentification by modelling feature uncertainty. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 552–561 (2019)
39. Yuan, L., Tay, F.E., Li, G., Wang, T., Feng, J.: Revisit knowledge distillation: a teacher-free framework. arXiv preprint arXiv:1909.11723 (2019)
40. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
41. Zhong, Y., Deng, W., Wang, M., Hu, J., Peng, J., Tao, X., Huang, Y.: Unequal-training for deep face recognition with long-tailed noisy data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7812–7821 (2019)