A Fast Knowledge Distillation Framework for Visual Recognition

Zhiqiang Shen[1,2,3] and Eric Xing[1,3]

[1] Carnegie Mellon University, Pittsburgh, USA
[2] Hong Kong University of Science and Technology, Hong Kong, China
[3] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
zhiqiangshen@cse.ust.hk,epxing@cs.cmu.edu
Project Page: http://zhiqiangshen.com/projects/FKD/index.html

Abstract. While Knowledge Distillation (KD) has been recognized as a useful tool in many visual tasks, such as supervised classification and self-supervised representation learning, the main drawback of a vanilla KD framework is its mechanism that consumes the majority of the computational overhead on forwarding through the giant teacher networks, making the entire learning procedure inefficient and costly. The recently proposed solution ReLabel suggests creating a label map for the entire image. During training, it receives the cropped region-level label by RoI aligning on a pre-generated entire label map, which allows for efficient supervision generation without having to pass through the teachers repeatedly. However, as the pre-trained teacher employed in ReLabel is from the conventional multi-crop scheme, there are various mismatches between the global label-map and region-level labels in this technique, resulting in performance deterioration compared to the vanilla KD. In this study, we present a Fast Knowledge Distillation (FKD) framework that replicates the distillation training phase and generates soft labels using the multi-crop KD approach, meanwhile training faster than ReLabel since no post-processes such as RoI align and softmax operations are used. When conducting multi-crop in the same image for data loading, our FKD is even more efficient than the traditional image classification framework. On ImageNet-1K, we obtain 80.1% Top-1 accuracy on ResNet-50, outperforming ReLabel by 1.2% while being faster in training and more flexible to use. On the distillation-based self-supervised learning task, we also show that FKD has an efficiency advantage.

1 Introduction

Knowledge Distillation (KD) [16] has been a widely used technique in various visual domains, such as the supervised recognition [29,49,48,23,34,2] and self-supervised representation learning [32,9,4]. The mechanism of KD is to force the student to imitate the output of a teacher network or ensemble teachers, as well as converge on the ground-truth labels. Given the parameters θ of the target student at iteration (t), we can learn the next iteration parameters $\theta^{(t+1)}$ by

Table 1. Feature-by-feature comparison between ReLabel [52] and our FKD.

Method	**Generating label**	**Label storage**	**Info. loss**	**Training**
Vanilla KD	Implicit	None	No	Slow
ReLabel [52]	Fast	Efficient	Yes	Fast
FKD (Ours)	Slow	Efficient	No	Faster

minimizing the following objective which contains two terms:

$$\boldsymbol{\theta}_{\text{student}}^{(t+1)} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N}(1-\lambda)\mathcal{H}\left(\boldsymbol{y}_n, \boldsymbol{S}_\theta\left(\boldsymbol{x}_n\right)\right) + \lambda\mathcal{H}\left(\boldsymbol{T}^{(t)}(\boldsymbol{x}_n), \boldsymbol{S}_\theta\left(\boldsymbol{x}_n\right)\right) \tag{1}$$

where $\boldsymbol{y}_n$ is the ground-truth for n-th sample. $\boldsymbol{T}^{(t)}$ is the teacher's output at iteration (t) and $\boldsymbol{S}_\theta(\boldsymbol{x}_n)$ is the student's prediction for the input sample $\boldsymbol{x}_n$. $\mathcal{H}$ is the cross-entropy loss function. λ is the coefficient for balancing the two objectives. The first term aims to minimize the entropy between one-hot ground-truth label and student's prediction while the second term is to minimize between teacher and student's predictions. The teacher $\boldsymbol{T}$ can be pre-trained in either supervised or self-supervised manners. Many literature [35,52,2,34] have empirically shown that the first term of true hard label in Eq. 1 is not required on larger-scale datasets like ImageNet [7] with more training budget if the teacher or ensembled teachers are accurate enough. In this work, we simply minimize the soft predictions between teacher and student models for the fast distillation design.

The inherent disadvantage in such a paradigm, according to KD's definition, is that a considerable proportion of computing resources is consumed on passing training data through large teacher networks to produce the supervision $\boldsymbol{T}^{(t)}$ in each iteration, rather than updating or training the target student parameters. Intuitively, the forward propagation through teachers can be shared across epochs since the parameters of them are frozen for the entire training. Based on this perspective, the vanilla distillation framework itself is inefficient, and how to reduce or share the forward computing of teacher networks across different epochs becomes the core for accelerating KD frameworks. A natural solution to overcome this drawback is to generate one probability vector as the soft label for each training image in advance, then reuse the pre-generated soft labels circularly for different training epochs. However, in modern neural network training, it is usually imposed various data augmentation strategies to avoid overfitting, particularly the random crop technique. This causes the inconsistency where the global-level soft vector for the entire image cannot precisely reflect the true probability distribution of the local image region after applying these augmentations.

To address the data augmentation, specially random-crop caused inconsistency issue in generating one global vector to the region-level input, while, preserving the advantage of informative soft labels, ReLabel [52] proposes to store the global label map from a pre-trained strong teacher and reutilize cross epochs by RoI align [13], as shown in Fig. 1 (left). However, because of the inconsistent processes of input on teachers, this strategy is essentially not equivalent to vanilla KD procedure. The mismatches are primarily from two factors: **(i)** the teacher

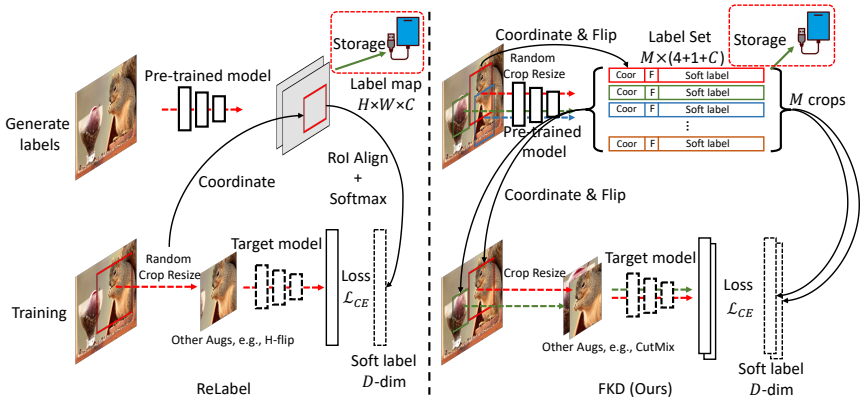

Fig. 1. Mechanism explanation of ReLabel and Fast Knowledge Distillation (FKD) framework. In label generation phase, ReLabel produces global-level label map through feeding the whole images into the pre-trained teacher, while FKD inputs regions of images, and maintains a set of soft labels. In network training phase, ReLabel employs *RoI Align* and *Softmax* to obtain the corresponding cropped labels for aligning the input, in contrast, FKD directly assigns the target soft label without any post-processing.

is usually trained with a random-crop-resize scheme, whereas, in ReLabel, the global label map is obtained by feeding into the whole image. Since in distillation the random-crop-resize is employed in the input space, thus the global label map cannot reflect the real soft distribution for image regions; **(ii)** RoI align will involve unexpected predictions on label maps, which cannot guarantee the sameness from this strategy and vanilla KD, thus, information loss exists.

In this work, we introduce a Fast Knowledge Distillation (FKD) framework to overcome the mismatching drawback and further avoid information loss on soft labels. Our strategy is straightforward: As shown in Fig. 1 (right), in the label generation phase, we directly store the soft probability from multiple random-crops into the label files, together with the coordinates and other data augmentation status like flipping. During training, we assign these stored coordinates back to the input image to generate the crop-resized input for passing through the networks, and compute the loss with the corresponding soft labels. The advantages of such a strategy are twofold: **(i)** Our region-based label generating process is identical to vanilla KD, so the obtained soft label for each input region is the same as oracle, indicating that no information is lost during the label creation phase; **(ii)** Our training phase enjoys a faster pace since no post-process is required, such as RoI align, softmax, etc. We can further assign multiple regions from the same image in a *mini*-batch to lessen the burden of data loading.

We demonstrate the advantages of our FKD in terms of accuracy and training speed on supervised and self-supervised learning tasks. In the supervised learning scenario, we compare the baseline ReLabel and vanilla KD (Oracle) from scratch across a variety of backbone network architectures, such as CNNs,

vision transformers, and the competitive MEAL V2 framework with pre-trained initialization. Our FKD is more than 1% higher and slightly faster than ReLabel on ImageNet-1K, and 3∼5× faster than oracle KD and MEAL V2 with similar performance. On the self-supervised distillation task, we employ S^2-BNN as the baseline for verifying the speed advantage of our proposed efficient framework.

Our contributions of this work:

- We present a fast knowledge distillation (FKD) framework that achieves the same high-level performance as vanilla KD, while keeping the same fast training speed and efficiency as non-KD approach without sacrificing performance.

- We reveal a discovery that in image classification frameworks, one image can be sampled many times with multiple crops within a *mini*-batch to facilitate data loading and speed up training, meanwhile obtaining better performance.

- To demonstrate the effectiveness and versatility of our approach, we perform FKD on a variety of tasks and distillation frameworks, including supervised classification and self-supervised learning with better results than prior art.

2 Related Work

Knowledge Distillation. The principle behind Knowledge Distillation [16] is that a student is encouraged to emulate or mimic the teachers' prediction, which helps student generalize better on unseen data. One core advantage of KD is that the teacher can provide softened distribution which contains richer information about input data compared to the traditional one-hot labels, especially when the data augmentation such as random-crop is used on the input space. Distillation can avoid incorrect labels by predicting them from the strong teachers in each iteration, which reflects the real situation of the transformed input data. We can also impose a temperature on the logits to re-scale the output distributions from teacher and student models to amplify the inter-class relationship on supervisions. Recently, many variants and extensions are proposed [24,18,44,53,23,25,34,48,50,6,36], such as employing internal feature representations [29], adversarial training with discriminators [31], transfer flow [49], contrastive distillation [40], patient and consistent [2], etc. For the broader overviews of related methods for knowledge distillation, please refer to [11,43].

Efficient Knowledge Distillation. Improving training efficiency for knowledge distillation is crucial for pushing this technique to a wider usage scope in real-world applications. Previous efforts in this direction are generally not sufficient. ReLabel [52] is a recently proposed solution that addresses this inefficient issue of KD surpassingly. It generates the global label map for the strong teacher and then reuses them through RoI align across different epochs. Our proposed FKD lies in an essentially different consideration and solution. We consider the characteristics of vanilla KD to generate the randomly-cropped region-level soft labels from the strong teachers and store them, then reuse them by allocating to different epochs in training. Our approach enjoys the same accuracy as vanilla KD and the same or faster training speed as regular non-KD classification frameworks, making it superior to ReLabel in both performance and training speed.

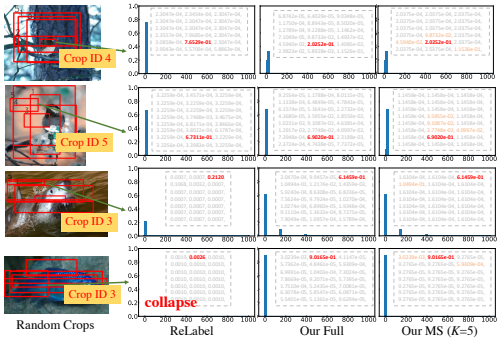

Fig. 2. Illustration of label distributions of ReLabel, our FKD full label and our quantized label (Top-5). "MS" denotes the marginal smoothed labels, more details can be referred in Sec. 3.5. Gray numbers in each block are the corresponding partial (as limited by space) probabilities/soft labels from different frameworks.

3 Approach

In this section, we begin by introducing several observations and properties from ReLabel's global-level soft label and FKD's region-level soft label distributions. Then, we present the detailed workflow of our FKD framework and elaborately analyze the generated label quality, training speed and the applicability on supervised and self-supervised learning. Finally, we analyze different strategies of soft label compression and provide their storage requirements for practical usage.

Preliminaries: Limitations of Previous Solution

According to the mechanism of ReLabel which is enabled by RoI align on global map, it is an approximation solution that inevitably will lose information on labels compared to the vanilla KD of region-level soft labels. In Fig. 2, we visualize the region-level label distributions of ReLabel and FKD on ImageNet-1K, and several empirical observations are noticed: **(i)** ReLabel is more confident in many cases of the regions, so the soft information is weaker than FKD. We analyze this is because ReLabel feeds the global images into the network instead of local regions, which makes the generated global label map encode more category information and ignore the backgrounds, forcing the soft label too close to the semantic ground-truth, as shown in Fig. 2 (row 1). Though sometimes the maximal probabilities are similar between ReLabel and FKD, FKD contains more informative subordinate probabilities in the label distribution, while ReLabel's are equally distributed, as shown in Fig. 2 (row 2); **(ii)** For some outlier regions, FKD is substantially more robust than ReLabel, such as the loose bounding boxes of objects, partial object, etc., as shown in Fig. 2 (row 3); **(iii)** In some particular circumstance, ReLabel unexpectedly collapsed with nearly uniform distribution, while FKD still works well, as shown in the bottom row of Fig. 2.

Moreover, there are existing mismatches between the soft label from ReLabel and oracle teacher prediction in KD when employing more data augmentations such as Flip, Color jittering, etc., since these augmentations are randomly applied during training. In ReLabel design, we cannot take them into account and prepare in advance when generating the global label map. In contrast, FKD is adequate to handle this situation: it is with ease to involve extra augmentations and record all information (ratio, degree, coefficient, etc.) for individual regions from

the same or different images, and generate corresponding soft label by feeding the transformed image regions into the pre-trained teacher networks. However, this strategy will increase the requirement of storage, so if it is budgeted, the alternative is to perform extra augmentations after receiving the cropped image regions during training, similar to ReLabel. Note that this will cause slightly mismatch between the transformed samples and corresponding soft labels, which is similar to the conventional augmentation mechanism but with soft labels.

3.1 Fast Knowledge Distillation

In a traditional visual training system, the deep network propagation and data loading are typically two main bottlenecks for resources. However, in a distillation framework, huge teachers have been the key training burden in addition to these computing demands. Our FKD seeks to address this intractable problem.
Label Generation Phase. Following the regular random-crop resize training strategy, we randomly crop $\boldsymbol{M}$ regions from one image and employ other augmentations like flipping on them, then feed these regions into the teachers to generate the corresponding soft label vectors $\boldsymbol{P}_i$, i.e., $\boldsymbol{P}_i = \boldsymbol{T}(\boldsymbol{R}_i)$ where $\boldsymbol{R}_i$ is the transformed region by transformations $\mathcal{F}_i$ and $\boldsymbol{T}$ is the pre-trained teacher network, i is the region index. We store all the region coordinates and augmentation hyper-parameters $\{\mathcal{F}\}$ together with the soft label set $\{\boldsymbol{P}\}$ for the following training phase, as shown in Fig. 1 (upper right). A detailed analysis of how to store these required values on hard drive is provided in the following section.
Training Phase. In the training stage, instead of randomly generating crops as the conventional image classification strategy, we directly load the label file, and assign our stored crop coordinates and data augmentations for this particular image to prepare the transformed region-level inputs. The corresponding soft label will be used as the supervision of these regions for training. With the cross-entropy loss, the objective is: $\mathcal{L} = -\sum_i \boldsymbol{P}_i \log \boldsymbol{S}_\theta(\boldsymbol{R}_i)$, where $\boldsymbol{S}_\theta(\boldsymbol{R}_i)$ is the student's prediction for the input region $\boldsymbol{R}_i$, θ is the parameter of the student model that we need to learn. The detailed training procedure is shown in Fig. 1.

3.2 Higher Label Quality

Distance Analysis. We analyze the quality of various formulations of labels through the entropy distance with measures on their mutual cross-entropy matrix. We consider three types of labels: (1) human-annotated one-hot label, ReLabel, and our FKD. We also calculate the distance of the predictions on four pre-trained models with different accuracies, including: vanilla PyTorch pre-trained model (weakest), Timm pre-trained model [46] (strongest), ReLabel trained model and FKD trained model. An overview of our illustration is shown in Fig. 3. The upper curves, as well in (2), are averaged cross-entropy across 50 classes of (ReLabel→FKD), (ReLabel→One-hot) and (FKD→One-hot). Here, we derive an important observation:

$$(\mathcal{D}^{CE}_{R\to F} = -\boldsymbol{P}_{FKD}\log \boldsymbol{P}_{ReLabel}) > (\mathcal{D}^{CE}_{R\to O} \ \ \boldsymbol{OR} \ \ \mathcal{D}^{CE}_{F\to O}) \tag{2}$$

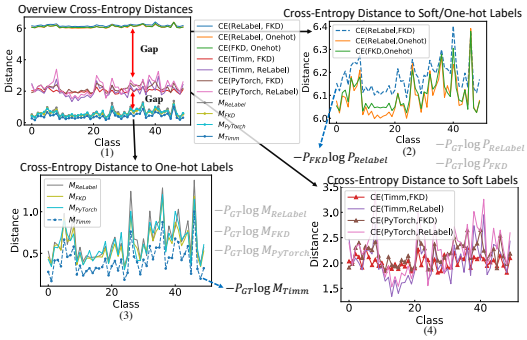

Fig. 3. Entropy distance analysis between different pairs of soft/one-hot labels and different trained models. (1) is the overall distance visualization. (2), (3), (4) represent each detailed group in (1). We illustrate the first 50 classes in ImageNet-1K dataset.

where $\mathcal{D}^{CE}_{R\to F}$ is the cross-entropy value of ReLabel $\to$ FKD. Essentially, FKD soft label can be regarded as the oracle KD label and $\mathcal{D}^{CE}_{R\to F}$ is the distance to such "KD ground truth". From Fig. 3 (2) we can see its distance is even larger than ReLabel and FKD to the one-hot label. Since ReLabel (global-map soft label) and FKD (region-level soft label) are greatly discrepant from the one-hot hard label, the gap between ReLabel and FKD ("KD ground truth") is fairly significant and considerable. If we shift attention to the curves of $\mathcal{D}^{CE}_{R\to O}$ and $\mathcal{D}^{CE}_{F\to O}$, they are highly aligned across different classes with similar values. In some particular classes, $\mathcal{D}^{CE}_{F\to O}$ are slightly larger. This is sensible as one-hot label is basically not the "optimal label" we desired.

In the bottom group of Fig. 3 (3), the entropy values are comparatively small. This is because the curves are from the pre-trained models with decent performance under the criterion of one-hot label. Among them, M_{Timm} has the minimal cross-entropy to the one-hot label, this is expected since timm model is optimized thoroughly to fit one-hot label with the highest accuracy. In Fig. 3 (4), $\mathcal{D}^{CE}_{Timm\to F}$ and $\mathcal{D}^{CE}_{PT\to F}$ lie in the middle of $\mathcal{D}^{CE}_{Timm\to R}$ and $\mathcal{D}^{CE}_{PT\to R}$ with smaller variances. This reflects that FKD is more stable than Relabel pre-trained models.

3.3 Faster Training Speed

Multi-crop sampling within a *mini*-batch. As illustrated in Fig. 1 (right), we can use multiple crops in the same image to facilitate loading image and label files. Intuitively, this will reduce the diversity of training samples in a *mini*-batch since some of the samples are from the same image. However, our experimental results indicate that it will not hurt the model's performance, in contrast, it even boosts the accuracy when the number of crops from the same image is within a reasonable range (e.g., 4∼8). We analyze this is because it can mitigate samples' variance dramatically for each *mini*-batch to make training more stable.

Serrated learning rate scheduler. Since FKD samples multiple crops (*#crop*) from one image, when iterating over the entire dataset once, we actually train the dataset *#crop* epochs with the same learning rate. It has no effect while using milestone/step *lr* scheduler, but it will change the *lr* curve to be serrated if applying continuous *cosine* or *linear* learning rate strategies. The accuracy may also be enhanced by multi-crop training for this reason.

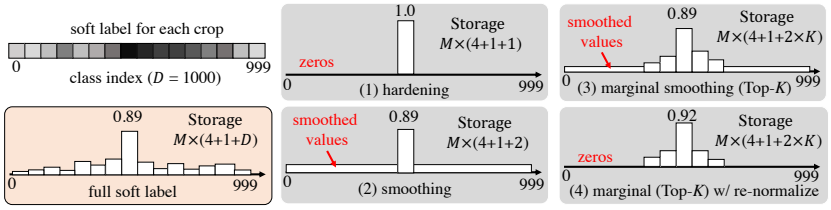

Fig. 4. Different label compression strategies and storage analyses for our fast knowledge distillation (FKD) framework. See Sec. 3.5 for more details.

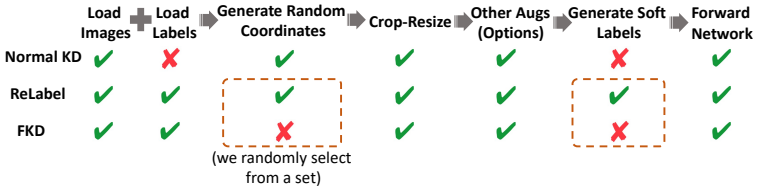

Fig. 5. Training workflow and analysis for vanilla KD, ReLabel and our fast knowledge distillation (FKD) framework. Maroon dashed boxes indicate that the processes are only required by ReLabel while not existing in our FKD. Note that "generate soft labels" indicates *RoI align + softmax* in ReLabel. We both have the recovering process from the compressed label to full soft label as discussed in Sec. 3.3.

Training Time Analysis:

1. Data Load Data loading strategy in FKD is efficient. For instance, when training with a *mini*-batch of 256, traditional image classification framework requires to load 256 images and ReLabel will load 256 images + 256 label files, while in our method, FKD only needs to load $\frac{256}{\#crop}$ images + $\frac{256}{\#crop}$ label files, even faster than traditional training if we choose a slightly larger value for $\#crop$ (when $\#crop > 2$)[4].

2. Label Preparation We assign #crop regions in an image to the current *mini*-batch for training. Since we store the label probability after *softmax* (in supervised learning), we can use assigned soft labels for the *mini*-batch samples directly without any post-process. This assignment is fast and efficient in implementation with a *randperm* function in PyTorch [26]. If the label is compressed using the following strategies, we will operate with an additional simple recovering process (as shown in Fig. 4) to obtain D-way soft label distributions. Note that ReLabel also has this process so the time consumption on this part will be similar to ReLabel. A detailed item-by-item workflow is shown in Fig. 5.

3.4 Training Self-supervised Model with Supervised Scheme

In this section, we introduce how to apply our FKD to the self-supervised learning (SSL) task with a faster training speed than the widely-used Siamese SSL

[4] Assume that loading each image and label file will consume similar time by CPUs.

Table 2. Detailed comparison of different label quantization/compression strategies on ImageNet-1K. M is the number of crops within an image during soft label generation, here we choose 200 crops as an example to calculate space consumption. N_{im} is the number of images, i.e., 1.2M for ImageNet-1K. S_{LM} is the size of label map. C_{class} is the number of classes. D_{DA} is the parameter dimension of data augmentations to store.

	ReLabel (Full)	ReLabel (Top-5)	Full	Hard	Smoothing	M Re-Norm (K=5)	MS (K=5)	MS (K=10)
Calculation	$N_{\text{im}} \times S_{\text{LM}} \times C_{\text{class}}$	$N_{\text{im}} \times S_{\text{LM}} \times 2C_{\text{Top-5}}$	$N_{\text{im}} \times (C_{\text{class}} + D_{\text{DA}})$	$N_{\text{im}} \times (1 + D_{\text{DA}})$	$N_{\text{im}} \times (2 + D_{\text{DA}})$	$N_{\text{im}} \times (2K + D_{\text{DA}})$	$N_{\text{im}} \times (2K + D_{\text{DA}})$	$N_{\text{im}} \times (2K + D_{\text{DA}})$
Dim. of Soft Label	$15 \times 15 \times 1,000$	$15 \times 15 \times 10$	$M \times 1{,}000$	$M \times 1$	$M \times 2$	$M \times 10$	$M \times 10$	$M \times 20$
+ Coordinate & Flip	–	–	$M \times 1{,}005$	$M \times 6$	$M \times 7$	$M \times 15$	$M \times 15$	$M \times 25$
Real Cons. on Disk	∼1TB	10GB	∼0.9TB	5.3GB	6.2GB	13.3GB	13.3GB	22.2GB

frameworks. The label generation (from the self-supervised strong teachers), label preparation and training procedure are similar to the supervised scheme. However, we keep the projection head in original SSL teachers as soft labels following [32] and store the soft labels before *softmax* for operating temperature[5].

3.5 Label Compression and Storage Analysis

We consider and formulate the following four strategies for compressing soft label for storage, an elaborated comparison of them can be referred to Table 2.

– **Hardening.** In hardening quantization strategy, the hard label $\boldsymbol{Y}_{\text{H}}$ is generated using the index of the maximum logits from the teacher predictions of regions. In general, label hardening is the one-hot label with correction by strong teacher models in region-level space.

$$\boldsymbol{Y}_{\text{H}} = \underset{\boldsymbol{c}}{\text{argmax}}\, \boldsymbol{z}_{\text{FKD}}(\boldsymbol{c}) \tag{3}$$

where $\boldsymbol{z}_{\text{FKD}}$ is the logits for each randomly cropped region produced by FKD.

– **Smoothing.** Smoothing quantization replaces one-hot hard label $\boldsymbol{Y}_{\text{H}}$ with a mixture of soft $\boldsymbol{y_c}$ and a uniform distribution same as label smoothing [37]:

$$\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{S}} = \begin{cases} \boldsymbol{p_c} & \text{if } \boldsymbol{c} = \textit{hardening}\ \ \textit{label}, \\ (1 - \boldsymbol{p_c})/(\boldsymbol{C} - 1) & \text{otherwise.} \end{cases} \tag{4}$$

where $\boldsymbol{p_c}$ is the probability after *softmax* at c-th class and $\boldsymbol{C}$ is the number of total classes. $(1 - \boldsymbol{p_c})/(\boldsymbol{C} - 1)$ is a small value for flattening the one-hot labels. $\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{S}} \in \boldsymbol{Y}_{\mathbf{S}}$ is the smoothed label at $\boldsymbol{c}$-th class.

– **Marginal Smoothing with Top-K (*MS*).** Marginal smoothing quantization reserves more soft information (Top-K) of teacher prediction than the single smoothing label $\boldsymbol{Y}_{\text{S}}$:

$$\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{MS}} = \begin{cases} \boldsymbol{p_c} & \text{if } \boldsymbol{c} \in \{\mathbf{Top}{-}K\}, \\ \dfrac{1 - \sum\limits_{\boldsymbol{c} \in \{\mathbf{Top}-K\}} \boldsymbol{p_c}}{\boldsymbol{C} - K} & \text{otherwise.} \end{cases} \tag{5}$$

where $\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{MS}} \in \boldsymbol{Y}_{\mathbf{MS}}$ is the marginally smoothed label at $\boldsymbol{c}$-th class.

[5] The temperature τ is applied on the *logits* before the *softmax* operation for self-supervised distillation.

Table 3. Comparison between ReLabel and our FKD on ImageNet-1K. "♢" denotes our retraining following the same protocol in the Appendix w/o distillation. Note that more augmentations (e.g., CutMix [51]) will further improve the accuracy, as provided in Sec. 4.1. Models are trained from scratch.

Method	Network	Top-1 (%)	Top-5 (%)	Training Time
Vanilla♢	ResNet-50	78.1	94.0	1.0
ReLabel [52]	ResNet-50	78.9	–	↑0.5% [52]
FKD (Ours)$_{\text{w/o warmup\&colorj}}$	ResNet-50	79.8	94.6	↓**0.5%**
FKD (Ours)	ResNet-50	**80.1**$^{+1.2}$	**94.8**	↓**0.5%**
Vanilla♢	ResNet-101	79.7	94.6	1.0
ReLabel [52]	ResNet-101	80.7	–	↑0.5% [52]
FKD (Ours)$_{\text{w/o warmup\&colorj}}$	ResNet-101	81.7	95.6	↓**0.5%**
FKD (Ours)	ResNet-101	**81.9**$^{+1.2}$	**95.7**	↓**0.5%**

– **Marginal Re-Norm with Top-K (*MR*).** Marginal re-normalization will re-normalize Top-K predictions to $\sum_{\boldsymbol{c}\in\{\mathbf{Top}-K\}} \boldsymbol{p_c}=1$ and maintain other logits to be zero (Different from ReLabel, we use *normalize* to calibrate the sum of Top-K predictions to 1, since our soft label is stored after softmax.):

$$\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{M}} = \begin{cases} \boldsymbol{p_c} & \text{if } \boldsymbol{c} \in \{\mathbf{Top}-K\}, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

$$\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{MR}} = \text{Normalize}(\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{M}}) = \frac{\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{M}}}{\sum_{\boldsymbol{c}=1}^{C}(\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{M}})} \tag{7}$$

where $\boldsymbol{y}_{\boldsymbol{c}}^{\mathbf{MR}} \in \boldsymbol{Y}_{\mathbf{MR}}$ is the re-normalized label at $\boldsymbol{c}$-th class.

4 Experiments

Experimental Settings and Datasets. Detailed lists of our hyper-parameter choices are shown in Appendix. Warmup and color jittering are not employed in the ablation studies. Except for experiments on MEAL V2, we use *EfficientNet-L2-ns-475* [38,48] as the teacher model, we also tried weaker teachers but the performance in our experiment is slightly worse. For MEAL V2, we follow its original design by using *SENet154 + ResNet152_v1s* ensemble (gluon version [12]) as the soft label. ImageNet-1K [7] is used for the supervised classification and self-supervised learning. COCO [21] is used for the transfer learning experiments.
Network Architectures. Experiments are conducted on Convolutional Neural Networks [19], such as ResNet [14], MobileNet [17], FBNet [47], EfficientNetv2 [39], and Vision Transformers [42,8], such as DeiT [41], SReT [33]. For binary backbone, we use ReActNet [22] in the self-supervised experiments.
Baseline Knowledge Distillation Methods.

▶ ReLabel [52] (Label Map Distillation). ReLabel used the pre-generated global label maps from the pre-trained teacher for reducing the cost on the teacher branch when conducting distillation.

Table 4. Comparison of MEAL V2 [35] and our FKD on ImageNet-1K. "w/ FKD" denotes the model is trained using the same protocol and hyper-parameters as original MEAL V2. "♡" represents the training using *cosine lr* and 1.5× epochs. Models are trained from the pre-trained initialization.

Method	Network	#Params	Top-1	Top-5	Speedup
MEAL V2 [35]	ResNet-50	25.6M	80.67	95.09	1.0
MEAL V2 w/ FKD	ResNet-50	25.6M	80.70	95.13	**0.3**×
MEAL V2 w/ ♡FKD	ResNet-50	25.6M	80.91	95.39	**0.5**×
MEAL V2 [35]	MobileNet V3-S0.75	2.04M	67.60	87.23	1.0
MEAL V2 w/ ♡FKD	MobileNet V3-S0.75	2.04M	67.83	87.35	**0.4**×
MEAL V2 [35]	MobileNet V3-S1.0	2.54M	69.65	88.71	1.0
MEAL V2 w/ ♡FKD	MobileNet V3-S1.0	2.54M	69.94	88.82	**0.4**×

- ▶ MEAL V2 [35] (Fine-tuning Distillation). MEAL V2 proposed to distill student network from the pre-trained parameters[6] and giant teacher ensemble for fast convergence and better accuracy.
- ▶ FunMatch [2] (Oracle Distillation). FunMatch is a standard knowledge distillation framework with strong teacher models and augmentations. We consider it as the strong baseline approach for efficient KD when using the same or similar teacher supervisors.
- ▷ S^2-BNN [32] (Self-supervised Distillation). S^2-BNN is a plain distillation solution for self-supervised learning task. The teacher is pre-learned from the self-supervised learning methods, such as MoCo V2 [5], SwAV [3], etc.

4.1 Supervised Learning

CNNs.

(i) ReLabel. The comparison with ReLabel is shown in Table 3, using the training settings introduced in our Appendix, which is the same as ReLabel, our accuracies on ResNet-50/101 both outperform ReLabel by more than **1.0%** with slightly faster training speed. These significant also consistent improvements of FKD show great potential and superiority for practical applications.

(ii) MEAL V2. We use FKD to train MEAL V2 models. The results are shown in Table 4, when employing the same hyper-parameters and teacher networks, FKD can speed up **2∼4× without compromising accuracy**. Using *cosine lr* and more epochs in training further improves the accuracy.

(iii) FunMatch (Oracle). We consider FunMatch as the oracle/strong KD baseline, our plain FKD w/o extra augmentations is slightly lower than FunMatch (80.5%) as they used more augmentations in training. After employing CutMix, which is similar to the FunMatch training setting, our result (80.9%)[7] outperforms FunMatch by 0.4%. Note that FunMatch needs 10× more budget

[6] The pre-trained parameter is from timm [45] with version<=0.4.12.

[7] The state-of-the-art non-KD training result on ResNet-50 (Timm [46]) with massive data augmentations is 79.8%, which is 1.1% lower than FKD.

Table 5. FKD with supervised Vision Transformer [8] and its variants on ImageNet-1K using 224×224 input resolution. Models are trained from scratch.

Method	Network	Epochs	#Params (M)	FLOPs (B)	Extra Data Aug.	Top-1 (%)	Speedup
DeiT [41] w/o KD	ViT-T	300	5.7	1.3	MixUp+CutMix+RA	72.2	–
DeiT [41] w/ KD	ViT-T	300	5.7	1.3	MixUp+CutMix+RA	74.5	1.0
ViT [8] (Vanilla)	ViT-T	300	5.7	1.3	**None**	68.7 [15]	–
ViT w/ FKD (Ours)	ViT-T	300	5.7	1.3	**None**	**75.2**	**0.15**×
SReT [33] w/o KD	SReT-LT	300	5.0	1.2	MixUp+CutMix+RA	76.7	–
SReT [33] w/ KD	SReT-LT	300	5.0	1.2	MixUp+CutMix+RA	77.7	1.0
SReT [33] (Vanilla)	SReT-LT	300	5.0	1.2	**None**	–	–
SReT w/ FKD (Ours)	SReT-LT	300	5.0	1.2	**None**	**78.7**	**0.14**×

Table 6. Ablation results (Top-1) on ImageNet-1K of different label quantization strategies. $m = 8$ is used in this ablation.

Method	Network	Full	Hard	Smoothing	Mar. Re-Norm (K=5)	Mar. Smoothing (K=5)	Mar. Smoothing (K=10)
MEAL V2 w/ FKD	ResNet-50	**80.65**	80.20	80.23	80.40	80.58	80.52
FKD (from scratch)	ResNet-50	79.48	79.09	79.37	79.23	**79.51**	79.44

Table 7. Ablation results (Top-1) on ImageNet-1K with different numbers (m) of cropping regions from the same image within a *mini*-batch.

Method	Network	$m = 1$	$m = 2$	$m = 4$	$m = 8$	$m = 16$	$m = 32$
Vanilla	ResNet-50	77.18	77.91	**78.14**	77.89	75.89	70.09
MEAL V2 w/ FKD	ResNet-50	80.67	**80.70**	80.66	80.58	80.36	80.17
FKD (from scratch)	ResNet-50	79.59	79.62	**79.76**	79.51	78.12	74.61

Table 8. ImageNet-1K clarification results on tiny networks.

FBNet-C Arch.	FLOPs: 375M	Acc.: 75.12%	+FKD: **77.13%**$^{+\mathbf{2.01\%}}$
EfficientNetv2-B0 Arch.	FLOPs: 700M	Acc.: 78.35%	+FKD: **79.94%**$^{+\mathbf{1.59\%}}$

for training than FKD (2 days *vs.* 20 days) with the same number of GPUs (e.g., 8 V100) since they explicitly forward giant teachers at each iteration of training.

(iv) Tiny Models. We also examine the generalization ability using the mobile-level models, such as FBNet [47], EfficientNetv2 [39] from [45]. As shown in Table 8, FKD consistently improves the base models by 2.01% and 1.59%, respectively. The training settings for them are provided in Appendix.

Vision Transformers.

(i) ViT/DeiT. The results are shown in Table 5 of the first group. Our non-extra augmentation result (75.2%) using ViT-T backbone is better than DeiT-T with distillation (74.5%), while we only require **0.15**× training resources than DeiT distillation protocol to train the model.

(ii) SReT. We also examine FKD using SReT-LT, result (78.7%) is consistently better than its original KD design (77.7%) with a faster training speed.

Ablations: (i) Effects of Crop Number in Each Image During Training. We explore the effect of different numbers of crops sampled from the same image within a *mini*-batch to the final performance. For the conventional data preparation strategy, on each image we solely sample one crop for a *mini*-batch to train the model. Here, we evaluate the m from 1 crop to 32 crops as shown in Table 7. Surprisingly, using a few crops from the same image leads to better per-

Table 9. Linear evaluation results of FKD with self-supervised Binary CNN (ReActNet [22]), Real-valued CNN (ResNet-50 [14]). FKD can speed up training by 3× with the same or similar linear evaluation performance.

Method	Network	Teacher	#Dims for Distilling	Training Epochs	Top-1 (%)	Speedup
S^2-BNN [32]	ReActNet	MoCo V2-800ep	128	200	61.5	1.0
FKD	ReActNet	MoCo V2-800ep	128	200	61.7	**0.4**×
S^2-BNN [32]	ResNet-50	SwAV/RN50-w4	3000	100	68.7	1.0
FKD	ResNet-50	SwAV/RN50-w4	3000	100	68.8	**0.3**×

formance than the single crop solution with a non-negligible margin, especially on the traditional image classification system. This indicates that the internal diversity of samples in a *mini*-batch has a limit for tolerance, properly reducing such diversity can mitigate the variance and boost accuracy, while we can also observe that after m>8, the performance decreases substantially, thus the diversity is basically still critical for learning good status of the model. Nevertheless, this is a good observation for us to speed up data loading in our FKD framework.

(ii) Effects of Crop Number for Soft Labels During Label Generation. Ideally, the number of crops is aligned with the number of training epochs by a shuffling and non-overlapping sampling strategy, which can exactly replicate the vanilla KD. We found FKD is surprisingly robust on fewer crops of soft labels, which can maintain a decent accuracy without a significant drop. We examined 100 crops (4.75G storage), result (79.7%) is tolerably inferior.

(iii) Different Label Compression Strategies. We evaluate the performance of different label compression strategies. We use m=8 for this ablation and the results are shown in Table 6. On MEAL V2 w/ FKD, we obtain the highest accuracy of 80.65% when using the full soft labels, while on the standard FKD, the best performance is from *Marginal Smoothing (K=5)* with 79.51%. Increasing K decreases both the accuracies in these two scenarios, we analyze that larger K will involve more noise or unnecessary minor information on the soft labels. While, they are still better than the *Hard* and *Smoothing* strategies.

4.2 More Comparison on ReaL [1] and ImageNetV2 [27] Datasets

In this section, we provide more results on ImageNet ReaL [1] and ImageNetV2 [27] datasets. On ImageNetV2 [27], we verify our FKD models on three metrics "Top-Images", "Matched Frequency", and "Threshold 0.7" as ReLabel. We conduct experiments on two network structures: ResNet-50 and ResNet-101. The results are shown in Table 10, we achieve consistent improvement over baseline ReLabel on both ResNet-50 and ResNet-101.

4.3 Self-Supervised Learning

S^2-BNN [32] is a pure distillation-based framework for self-supervised learning, thus the proposed FKD approach is eligible to train S^2-BNN [32] in the proposed way efficiently. We employ SwAV [3] and MoCo V2 [5] pre-trained models as

Table 10. Results of FKD on ImageNet ReaL [1] and ImageNetV2 [27] with ResNet-{50, 101}. * indicates that results are tested using their provided pre-trained model.

Method	ImageNet-1K	ReaL	ImageNetV2 Top-images	ImageNetV2 Matched-frequency	ImageNetV2 Threshold-0.7
ResNet-50:					
ReLabel	78.9	85.0	80.5	67.3	76.0
FKD	**80.1**	**85.8**	**81.2**	**68.2**	**76.9**
ResNet-101:					
ReLabel*	80.7	86.5	82.4	69.7	78.2
FKD	**81.9**	**87.1**	**83.2**	**70.7**	**79.1**

Table 11. Comparison of transfer learning performance with ReLabel on detection and instance segmentation tasks. The training and evaluation are conducted on COCO dataset [21].

Method	Network	Faster RCNN w/ FPN	Mask-RCNN w/ FPN	
		bbox AP	bbox AP	mask AP
Baseline	ResNet-50	37.7	38.5	34.7
ReLabel	ResNet-50	38.2	39.1	35.2
FKD	ResNet-50	38.7	39.7	35.9

the teacher networks. Considering that the distribution from the SSL learned teachers is more flattening than the supervised teacher predictions (meaning that the subordinate classes from SSL trained teachers carry crucial information), we use the full soft label in this scenario, and leave the label compression strategies on SSL task as a future study. We employ ReActNet [22] and ResNet-50 [14] as the target/student backbones in these experiments. The results are shown in Table 9, our FKD trained models achieve slightly better performance than S^2-BNN with roughly 3× acceleration since we only use a single branch for training, the same as traditional classification pipeline that uses soft label and CE loss. The slight boosts are from our lite data augmentation for FKD when generating SSL soft labels. This is interesting and it is worth exploring further on the data augmentation strategies for distillation-based or FKD-equipped SSL methods.

4.4 Transfer Learning

We further examine whether FKD obtained improvements on ImageNet-1K can be transferred to various downstream tasks. As in Table 11, we present the results of object detection and instance segmentation on COCO [21] with models pretrained on ImageNet-1K using FKD. We also employ Faster RCNN [28] and Mask RCNN [13] with FPN [20] following ReLabel. Over the vanilla baseline and ReLabel, our FKD pre-trained weights show consistent gains on the downstream tasks. More visualizations, analyses and discussions are provided in Appendix.

5 Conclusion

It is worthwhile investigating approaches to boost the training efficiency and speed of vanilla KD given its widespread use and exceptional performance in training compact and efficient networks. In this paper, we have presented a fast distillation framework through the pre-generated region-level soft label scheme. We have elaborately discussed the strategies of compressing soft label for practical storage and their performance comparison. We identified an interesting discovery that the training samples within a *mini*-batch can be cropped from the same input images to facilitate data loading with better accuracy. We exhibit the effectiveness and adaptability of our framework by demonstrating it on supervised image classification and self-supervised representation learning tasks.

Appendix

A Visualization, Analysis and Discussion

To investigate the learned differences of information between ReLabel and FKD, we depict the intermediate attention maps using gradient-based localization [30]. There are three important observations that align our aforementioned analyses in Fig. 6 and 7.

(i) FKD's predictions are less confident than ReLabel with more surrounding context; This is reasonable since in random-crop training, many crops are basically backgrounds (context), the soft predicted label from the teacher model might be completely different from the ground-truth one-hot label and the training mechanism of FKD can leverage the additional information from context.

(ii) FKD's attention maps have a larger active area on the object regions, which indicates that FKD trained model utilizes more cues for prediction and also captures more subtle and fine-grained information. However, it is interesting to see that the *guided backprop* is more focused than ReLabel.

(iii) ReLabel's attention is more aligned with PyTorch pre-trained model, while FKD's results are substantially unique to them. It implies that FKD's learned attention differs significantly from one-hot and global label map learned models.

B Training Details and Experimental Settings

Training details for Table 3 of the main text. We employ the training settings and hyper-parameters following Table 12, which are the same as ReLabel. We use 4 as the number of crops in each image during training.

Table 12. Training hyper-parameters and details for ReLabel [52] and FKD used in Table 3 of the main text.

Method	ReLabel [52] or FKD
Teacher	EfficientNet-L2-ns-475
Epoch	300
Batch size	1,024
Optimizer	SGD
Init. *lr*	0.1
lr scheduler	cosine
Weight decay	1e-4
Random crop	Yes
Flipping	Yes
Warmup epochs	5
Color jittering	Yes

Table 13. Training hyper-parameters and details for the comparison in Table 5 of the main text when employing ViT [8], DeiT [41] and SReT [33] as the backbone networks. Table is adapted from [41].

Method	ViT-B [8]	DeiT [41]/SReT [33]	FKD
Epoch	300	300	300
Batch size	4096	1024	1024
Optimizer	AdamW	AdamW	AdamW
Init. *lr*	0.003	0.001	0.002
lr scheduler	cosine	cosine	cosine
Weight decay	0.3	0.05	0.05
Warmup epochs	3.4	5	5
Label smoothing	None	0.1	None
Dropout	0.1	None	None
Stoch. Depth	None	0.1	0.1
Repeated Aug	None	Yes	None
Gradient Clip.	Yes	None	None
Rand Augment	None	9/0.5	None
Mixup prob.	None	0.8	None
Cutmix prob.	None	1.0	None
Erasing prob.	None	0.25	None

Training details for Table 5 of the main text. When comparing our FKD with ViT [8]/DeiT [41]/SReT [33] (Table 5 of the main text), we employ the training settings and hyper-parameters following Table 13.

Training details for Table 8 of the main text. The training settings and hyper-parameters of FKD with FBNet-C100 [47] and EfficientNetv2-B0 [39] backbones (Table 8 of the main text) are provided in Table 13 which are the same as the training protocol on ViT, DeiT and SReT. We use 4 as the number of crops in each image during training.

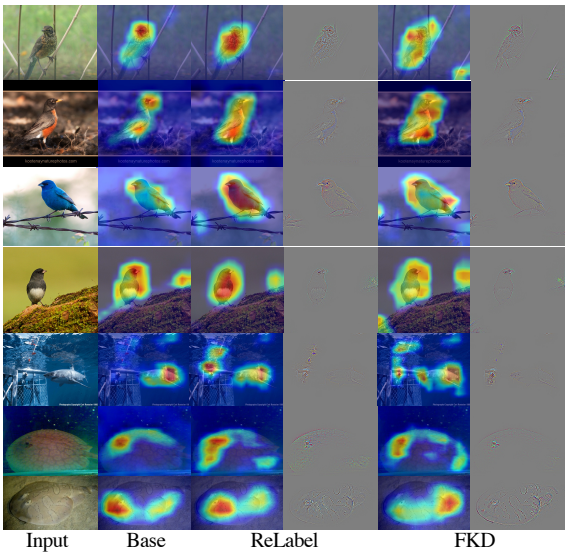

Fig. 6. Visualization of learned attention map using GradCAM [30,10]. "Base" indicates the pre-trained PyTorch model. In each group of ReLabel and FKD, left is *Grad-CAM* and right is *Guided Backprop*.

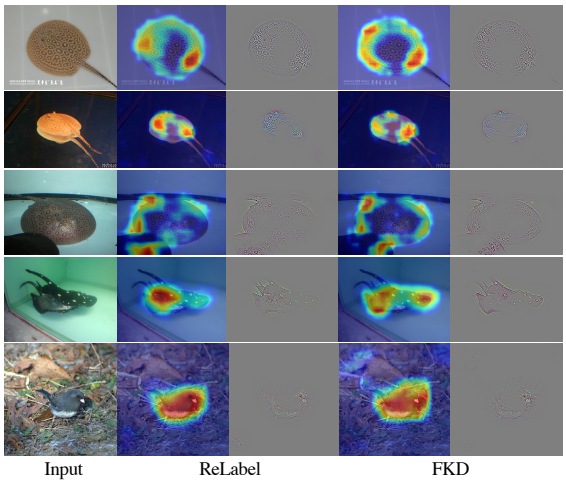

Fig. 7. More visualization of response/attention maps.

References

1. Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., Oord, A.v.d.: Are we done with imagenet? arXiv preprint arXiv:2006.07159 (2020) 13, 14
2. Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., Kolesnikov, A.: Knowledge distillation: A good teacher is patient and consistent. arXiv preprint arXiv:2106.05237 (2021) 1, 2, 4, 11
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2020) 11, 13
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021) 1
5. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020) 11, 13
6. Chung, I., Park, S., Kim, J., Kwak, N.: Feature-map-level online adversarial knowledge distillation. In: International Conference on Machine Learning. PMLR (2020) 4
7. 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) 2, 10
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020) 10, 12, 16
9. Fang, Z., Wang, J., Wang, L., Zhang, L., Yang, Y., Liu, Z.: Seed: Self-supervised distillation for visual representation. In: ICLR (2021) 1
10. Gildenblat, J., contributors: Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam (2021) 17
11. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision **129**(6), 1789–1819 (2021) 4
12. Guo, J., He, H., He, T., Lausen, L., Li, M., Lin, H., Shi, X., Wang, C., Xie, J., Zha, S., Zhang, A., Zhang, H., Zhang, Z., Zhang, Z., Zheng, S., Zhu, Y.: Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing. Journal of Machine Learning Research (2020) 10
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 2, 14
14. 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) 10, 13, 14
15. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: International Conference on Computer Vision (ICCV) (2021) 12
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 1, 4
17. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017) 10
18. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017) 4

19. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks (1995) 10
20. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017) 14
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) 10, 14
22. Liu, Z., Shen, Z., Savvides, M., Cheng, K.T.: Reactnet: Towards precise binary neural network with generalized activation functions. In: European Conference on Computer Vision. pp. 143–159. Springer (2020) 10, 13, 14
23. Müller, R., Kornblith, S., Hinton, G.: When does label smoothing help? In: NeurIPS (2019) 1, 4
24. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE symposium on security and privacy (SP). pp. 582–597. IEEE (2016) 4
25. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: CVPR. pp. 3967–3976 (2019) 4
26. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32**, 8026–8037 (2019) 8
27. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: ICML. pp. 5389–5400. PMLR (2019) 13, 14
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015) 14
29. 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) 1, 4
30. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017) 15, 17
31. Shen, Z., He, Z., Xue, X.: Meal: Multi-model ensemble via adversarial learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4886–4893 (2019) 4
32. Shen, Z., Liu, Z., Qin, J., Huang, L., Cheng, K.T., Savvides, M.: S2-bnn: Bridging the gap between self-supervised real and 1-bit neural networks via guided distribution calibration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2165–2174 (2021) 1, 9, 11, 13
33. Shen, Z., Liu, Z., Xing, E.: Sliced recursive transformer. arXiv preprint arXiv: 2111.05297 (2021) 10, 12, 16
34. Shen, Z., Liu, Z., Xu, D., Chen, Z., Cheng, K.T., Savvides, M.: Is label smoothing truly incompatible with knowledge distillation: An empirical study. In: ICLR (2021) 1, 2, 4
35. Shen, Z., Savvides, M.: Meal v2: Boosting vanilla resnet-50 to 80%+ top-1 accuracy on imagenet without tricks. arXiv preprint arXiv:2009.08453 (2020) 2, 11
36. Stanton, S., Izmailov, P., Kirichenko, P., Alemi, A.A., Wilson, A.G.: Does knowledge distillation really work? arXiv preprint arXiv:2106.05945 (2021) 4
37. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016) 9

38. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. PMLR (2019) 10
39. Tan, M., Le, Q.: Efficientnetv2: Smaller models and faster training. In: International Conference on Machine Learning. pp. 10096–10106. PMLR (2021) 10, 12, 16
40. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: International Conference on Learning Representations (2019) 4
41. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021) 10, 12, 16
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 10
43. Wang, L., Yoon, K.J.: Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2021). https://doi.org/10.1109/TPAMI.2021.3055564 4
44. Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A.: Dataset distillation. arXiv preprint arXiv:1811.10959 (2018) 4
45. Wightman, R.: Pytorch image models. `https://github.com/rwightman/pytorch-image-models` (2019). https://doi.org/10.5281/zenodo.4414861 11, 12
46. Wightman, R., Touvron, H., Jégou, H.: Resnet strikes back: An improved training procedure in timm. arXiv preprint arXiv:2110.00476 (2021) 6, 11
47. Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., Keutzer, K.: Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10734–10742 (2019) 10, 12, 16
48. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10687–10698 (2020) 1, 4, 10
49. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4133–4141 (2017) 1, 4
50. Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8715–8724 (2020) 4
51. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6023–6032 (2019) 10
52. Yun, S., Oh, S.J., Heo, B., Han, D., Choe, J., Chun, S.: Re-labeling imagenet: from single to multi-labels, from global to localized labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2340–2350 (2021) 2, 4, 10, 15
53. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., Ma, K.: Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In: ICCV (2019) 4