Enhanced Knowledge Distillation by Auxiliary Classifiers

by

Aryan Asadian

A thesis submitted to the School of
Graduate and Postdoctoral Studies in
partial fulfillment of the requirements for
the degree of

Master of Science in Computer Science

Faculty of Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

August 2021

Thesis Examination Information

Submitted by: **Aryan Asadian**

Master of Science in **Computer Science**

Thesis title:Enhanced Knowledge Distillation by Auxiliary Classifiers

An oral defense of this thesis took place on July 27, 2021 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Heidar (Kourosh) Davoudi
Research Supervisor	Dr. Amirali Salehi-Abari
Examining Committee Member	Dr. Jaroslaw Szlichta
Examining Committee Member	Dr. Julie Thorpe
Thesis Examiner	Dr. Mehran Ebrahimi

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Deep neural models have shown promising results in various areas, e.g., computer vision and natural language processing, at the cost of high computation and storage resource consumption. These characteristics of deep neural networks have acted as a barrier in resource-constraint environments, e.g., smartphones. Among numerous proposed approaches to mitigate this limitation, *knowledge distillation* has gained much attention due to its generalizability and simplicity in implementation. This thesis introduces the enhanced knowledge distillation (EKD), a simple yet effective approach to outperform the canonical knowledge distillation using multiple classifier heads at various teachers' depths. First, multiple classifier heads are attached to the teacher model in different depths. The mounted heads benefit from the fully trained teacher model and converge fast while the backbone teacher is frozen. The cohort of all classifiers supervises the student in the last step. EKD showed superior performance in comparison with some of the state-of-the-art distillation frameworks.

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Aryan Asadian

Statement of Contributions

I hereby certify that I have been the primary contributor of this thesis by developing the algorithms, implementing them, and designing the experiments. I have also written most content of this thesis. However, some texts of this thesis are borrowed from the conference paper jointly coauthored by my thesis supervisor Dr. Amirali Salehi-Abari and me [4].

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Amirali Salehi-Abari, for his dedicated support and guidance. Amirali continuously encouraged and was always willing and enthusiastic to assist in any way he could throughout the research project. Besides research, he has been a mentor to improve my critical thinking skills, and practical research. I would also like to express my special thanks to my lovely family, my mother, who has been an inspiring leader for our family, my father for his unconditional love and support, and my brother, Arash, who has been my best friend all the times. I owe all of my success to their unconditional love.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

SGD ..... Stochastic Gradient Descent.
ResNet ..... Residual Network.
WResNet ..... Wide Residual Network.
KD ..... Knowledge Distillation.
AT ..... Attention Distillation.
EKD ..... Enhanced Knowledge Distillation by Auxiliary Classifiers.
TOFD ..... Task-Oriented Feature Distillation.
TAKD ..... Knowledge Distillation via Teacher Assistant.
MHKD ..... Multi Head Knowledge Distillation.
BWN ..... Binary Weight Network.
TWN ..... Ternary Weight Network.
FSP ..... Flow of Solution Procedure.
AB ..... Activation Boundaries.
RCO ..... Route-Constrained Optimization.
GAN ..... Generative Adversarial Network.
DML ..... Deep Mutual Learning.
HD ..... Hierarchical Distillation.
HNE ..... Hierarchical Neural Ensemble.
RKD ..... Residual Error-Based Knowledge Distillation.
ES-KD ..... Early-Stopping Knowledge Distillation.
MCH ..... Multi-Classifier Head.
MSDNet ..... Multi-Scale Dense Network.
DSN ..... deeply-supervised Network.
CDL ..... Conditional Deep Learning.
ELF ..... Early-Exiting Framework.
RDI-Net ..... Robust Dynamic Inference Network.
SDN ..... Shallow Deep Network.
MSD ..... Multi-Self-Distillation.
PKD ..... Patient Knowledge Distillation.
NAS ..... Neural Architecture Search.
CRD ..... Contrastive Representation Distillation.
CE ..... Cross-Entropy.
KL-Div ..... Kullback-Leibler Divergence.

τ ..... Temperature.
α ..... Distillation Balancing Weight.
$\mathbf{z}$ ..... Logits.
$H()$ ..... Entropy.
$||\mathbf{x}||_p$ ..... L_p norm.

Chapter 1

Introduction

1.1 Motivation

Deep neural networks have exhibited state-of-the-art performance in various domains such as computer vision [8, 31] and natural language processing [9, 14]. However, these models notoriously contain many parameters, requiring large storage space and intensive computation resources for training and inference. These characteristics have impeded the deployment of deep neural networks in resource-limited environments (e.g., mobile phones or embedded devices). The infeasibility of deploying independent deep neural networks in resource-limited environments has also raised security and privacy concerns. The client (i.e., the resource-limited device) usually has to send their generated data (e.g., confidential medical records) to a server that hosts a trained deep neural network through the internet. These problems and concerns have led to a broad range of solutions for acquiring more compact yet effective models such as network pruning [7], network quantization [63], design of efficient architectures [56], and knowledge distillation [24].

Knowledge distillation has gained popularity due to its applicability to different

domains and simple implementation [24]. Knowledge distillation is a way of training a model as *student* by using a powerful optimized *teacher* model. The student model learns to approximate the teacher's behavior with fewer parameters. Canonical knowledge distillation (KD) uses the smooth teacher's generated probabilities [24] to improve the student.

Recently, various knowledge distillation frameworks have tried to outperform the canonical KD by defining new sources of knowledge [65, 70] and establishing new approaches to transfer teacher's knowledge [42, 46]. *Hint distillation* [46] transfers the teacher's intermediate knowledge to the student in addition to the class probabilities. Attention distillation (AT) [70] distills the teacher's intermediate knowledge in the form of *attention maps*, i.e., averaged intermediate representations. These approaches have not shown guaranteed improvement to KD due to their high sensitivity to the hyper-parameters (e.g., the selected pair of intermediate layers) and limiting assumptions (e.g., teacher and student with same-size intermediate layers).

One of the main challenges in knowledge distillation is that the teachers and students are usually entirely dissimilar in size and complexity, which leads to different-size intermediate representations. Addressing this incompatibility in distillation requires either introducing multiple new hyperparameters or selecting the teacher and student with similar architecture [46, 65, 70].

Canonical KD has also shown weak performance between teachers and students with different model complexity (i.e., notable capacity gap) [17, 39]. Knowledge distillation with teacher assistants (TAKD) [39] bridges the huge capacity gap between the teacher and student by establishing a chain of teacher assistant models. Although TAKD could, to some degree, cope with the capacity gap problem, it is computationally expensive and also requires setting numerous hyperparameters.

1.2 Contributions

This thesis proposes a new extension of knowledge distillation called Enhanced Knowledge Distillation by Auxiliary Classifiers (EKD). Inspired by multi-exit classifiers (deep neural models with multiple classifier heads at different layers), our approach exploits the teacher's intermediate knowledge (at multiple depths) with intermediate classifier heads. First, multiple classifier heads are mounted to various depths of a fully optimized teacher model. Since the teacher is already converged, the mounted classifier heads can cheaply be optimized while the teacher model is frozen. The intermediate classifier heads map the high dimensional intermediate representations to the probability distribution of the classes–an understandable, semantic space for the student. All the trained classifier heads, including the intermediate heads and the main one in the backbone teacher, create a cohort of teachers and distill their knowledge to the student. The main contributions of this thesis are as follows:

- We introduce *Enhanced Knowledge Distillation by Auxiliary Classifiers* (EKD), a general distillation framework that improves training the student, using multiple cheaply-acquired intermediate classifier heads on the teacher model. EKD is easy to implement and can also be used in combination with other model compression approaches (e.g., model pruning and quantization).

- Through extensive experiments, we show that EKD can distill the teacher's intermediate knowledge efficiently.

- Our experiments confirm that EKD can efficiently address the *capacity gap* problem of conventional knowledge distillation.

- Through extensive experiments on various teacher-student model pairs and three well-known image classification datasets, we show that EKD is not only

a straightforward approach for distillation, but it can also surpass many state-of-the-art distillation frameworks by a large margin.

- We also explore the reasons behind EKD's improvements using the concepts of *information entropy*, and *overthinking*.

1.3 Thesis Outline

The remainder of this thesis is organized as follows.

Chapter 2 provides complete background on model compression approaches, including knowledge distillation. Also, we explain the mathematical intuition behind distillation and various categories of it.

In **Chapter 3**, we present the related work to EKD, how they differ from EKD, and their strengths and weaknesses compared to EKD.

In **Chapter 4**, we describe our proposed distillation approach (EKD) and its the mathematical intuition.

In **Chapter 5**, we show how EKD outperforms multiple state-of-the-art distillation approaches by a large margin using different teacher-student pairs with diverse capacity gaps on three image classification benchmarks.

In **Chapter 6**, we conclude our thesis with a summary of our proposed framework and discuss the potential future directions for future research.

Chapter 2

Background

This Chapter is organized as follows. Model compression techniques are described in Section 2.1. Canonical knowledge distillation (KD), its motivation, and general categories of KD are presented in Section 2.2.

2.1 Model Compression

The encouraging results of deep neural networks in various domains, e.g., computer vision [8, 31] and natural language processing [9, 14], have led practitioners to benefit from these solutions in more applications. However, this achievement has been at the cost of excessive resource (storage and computation) consumption.

Started by AlexNet [33], deep neural models have shown significant performance in various computer vision tasks. Since then, numerous neural network architectures have been proposed to surpass the preceding models. Usually, these architectures benefit from deeper [19, 51] and wider neural structure [69], which generates high dimensional intermediate representations from the input data. However, these new massive architectures are more prone to overfitting. The forward and backward prop-

agations in these models require millions (or even billions) of high-dimensional matrix multiplications, leading to numerous float numbers with high precision as weights and activations. This massive volume of computation prolongs both training and inference duration.

Although these models have shown significant performance, deep learning practitioners use *ensemble* of multiple deep models to reach even higher accuracy. These practices have made deep neural models an infeasible choice for the environments where the resources are constrained (e.g., mobile phones and embedded devices) or time-delays are strictly prohibited (e.g., real-time applications).

Model compression refers to methods that alleviate the challenges mentioned above in various ways. In general, model compression can be divided into four general categories.

2.1.1 Model Pruning

Model pruning is an optimization framework that reduces the unnecessary parameters of a large model to gain an efficient smaller network. Although there are many variants of model pruning, they mostly follow a similar process. This technique gradually removes some of the parameters or sub-components of a fully trained model based on a scoring system; then, the pruned model goes through fine-tuning to recover its highest accuracy. The final result is a model with fewer parameters (i.e., smaller size) that preserves or even, in some cases, outperforms its not-pruned predecessor's accuracy.

Consider a fully-trained model as an approximation function $f(\mathbf{x}|\mathbf{W})$, where the input $\mathbf{x}$ is fed to the model with learnable parameters $\mathbf{W}$ to produce a high-dimensional representation. Model pruning aims to use the mentioned model and generates a new model $f(\mathbf{x}|\mathbf{W}_{pruned})$, where $\mathbf{W}_{pruned} = \mathbf{M} \odot \mathbf{W}$, is the remaining

weights after element-wise multiplication of $\mathbf{W}$ by the binary mask $\mathbf{M}$ that removes a fraction of the model's parameters. Besides the theoretical definition, pruning is usually done by removing the target parameter or setting the target parameter to zero.

The pruning happens either in an unstructured manner (i.e., a single parameter in a layer) or in a more structured way, i.e., the parameters in a specific layer or a channel in one of the convolutional layers [21]. The latter outperforms the former in terms of final accuracy [7]. Moreover, unstructured pruning generates sparsely incompatible models with the existing deep learning frameworks or even current hardware.

2.1.2 Model Quantization

The main idea behind *model quantization* is to reduce the computational and storage costs of the model by approximating a full-precision (e.g., 32 bit) deep neural network to a low-bit (e.g., integer-based or even binary) model. Model quantization methods can be categorized into two general sectors.

Approximation-based Quantization

In this category, the model is quantized using *step functions* in the forward propagation. The step functions approximate real-valued numbers into linear values by dividing the continuous axis into multiple intervals. Sign $sgn(x)$ and Heaviside are two examples of step functions. Due to the saturated gradient problem in step functions, the backward process should be approximated as well. Dissimilar approximations in the forward and backward computations lead to gradient mismatch. Binary-Connect [12] converts the 32-bit full precision weights $\mathbf{W}$ to $sgn(\mathbf{W})$ in the forward pass and approximates it by hard-tanh function in backward propagation to match the gradi-

ents. *XNOR-Net* and *Binary-Weight Networks (BWN)* [45] are two other approaches that follow binary quantization. The former only binarizes the convolutional layer's weights, while the latter quantizes the activations as well. These quantizations lead to up 32 and 58 times less memory consumption and faster computation, respectively, while preserving the accuracy of the full-precision models. Some proposed approaches move beyond binary quantization, e.g., *Ternary weight network (TWN)* [36] restricts the weights to one of the discrete values of $\{+1, 0, -1\}$.

Optimization-based Quantization

The optimization-based quantization is only available for weights. This approach contains a computationally-expensive iterative process. Leng et al. [35] propose a model quantization technique that compresses the neural network by reducing the parameters' precision, at the expense of accuracy [63].

In general, model quantization requires elaborate fine-tuning and multiple interventions during the forward and backward passes.

2.1.3 Designing Deep Neural Architectures

Designing efficient deep neural architectures is another category to satisfy the appropriate accuracy requirement without violating the resource boundaries. Efficient architectures try to preserve or surpass the conventional deep neural model's performance by using elaborately designed modules or modified layers.

Deep neural models are global approximators, and theoretically, a sufficiently deep or wide neural network can approximate any function [5]. However, it has been observed [19] that deep neural networks saturate after passing a certain depth. This phenomenon is due to vanishing or exploding gradients which is a common barrier in

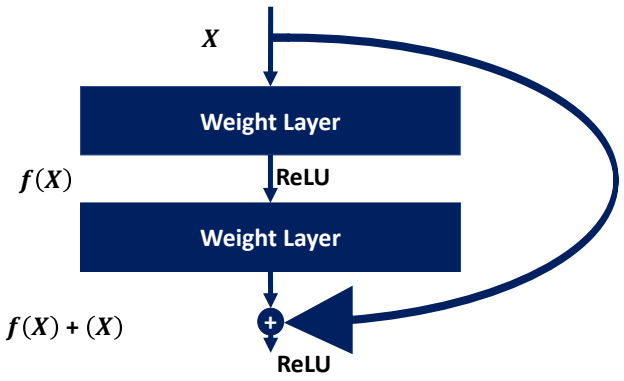


Figure 2.1: A residual block containing skip connections (identity mapping)

establishing deeper neural networks. By the introduction of *residual architecture*, and *resnet models* [19], neural networks could address the vanishing gradients to some degree and experience better generalization by using multiple identity mappings through the model. In general, each residual model is a stack of multiple residual blocks, each of which contains multiple convolutions, batch normalization, and pooling layers. Identity mapping shortcuts connect the beginning to the end of multiple residual blocks to address the vanishing gradients, i.e., identity mappings skip a series of blocks, and due to this functionality, it is also known as *skip connections*. Figure 2.1 illustrates a regular residual block.

Inspired by Network-in-Network [37] models in using 1x1 kernels, *GoogleNet* (also known as *Inception*) [54] modifies the conventional convolutional neural network to a more feasible modular model called Inception. In the Inception model, some of the regular 5x5 or 3x3 convolutional filters are replaced with 1x1 convolutions. This replacement significantly reduces the total number of multiplications with similar functionality to regular convolutions, e.g., by only replacing a 3x3 convolution filter with a 1x1, the number of multiplications reduces 9 times. Figure 2.2 depicts a single Inception module scheme.

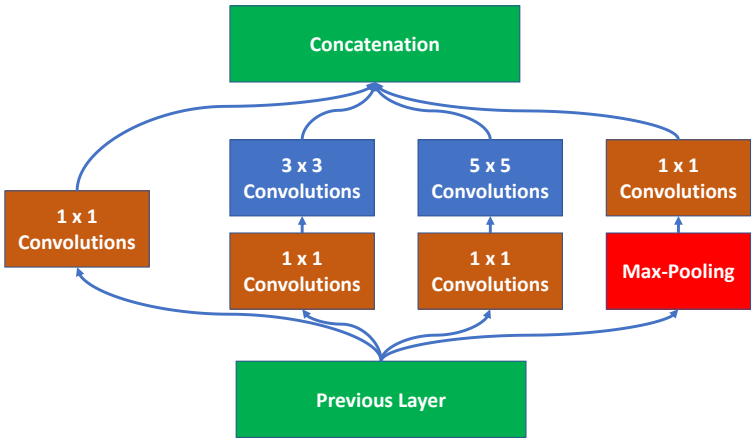


Figure 2.2: An Inception module. GoogleNet is a stack of multiple Inception modules, which benefit from efficient 1x1 convolutions.

Conventional deep neural models could also be problematic when there is no computational limitation during training. When the model is gigantic, distributed training over multiple devices, e.g., GPUs, would be challenging because huge models require more communication than their smaller counterparts. Squeeze-Net [28] has been proposed to address the mentioned issues. It reaches the AlexNet [33] test accuracy on ImagNet [13] dataset while it has 50% fewer parameters.

MobileNet-V1 [27] decreases the computational costs of the regular convolutional neural networks by replacing the canonical convolution layers with *Depthwise Separable Convolution* layers. Depthwise separable convolution (see Figure 2.3) divides the regular convolution layer into two sub-modules: a depthwise convolution (i.e., a single 3x3 convolution for each input dimension) and a pointwise 1x1 convolution layer concatenating the depthwise convolutions' outputs. This modularization sharply reduces the total computation while preserving a similar performance.

MobileNet-V2 [48] combines depth-wise convolutions and skip-connections to increase the model's depth in the resources-limited environments (e.g., mobile devices).

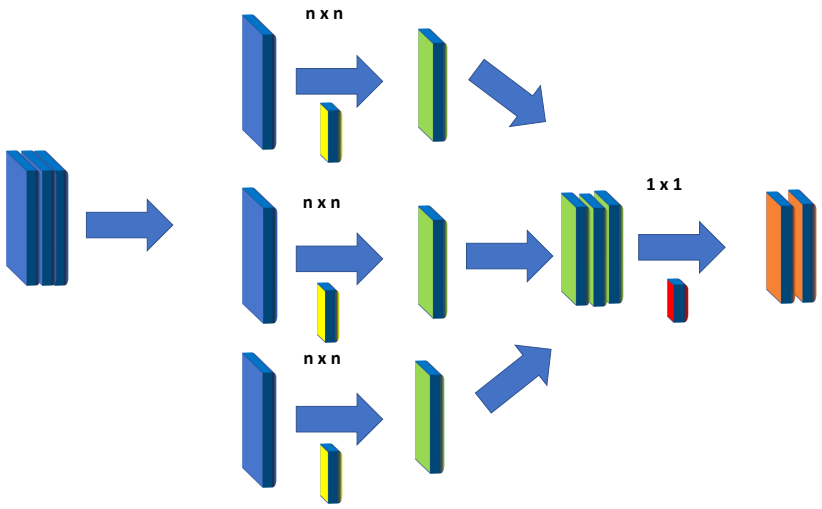
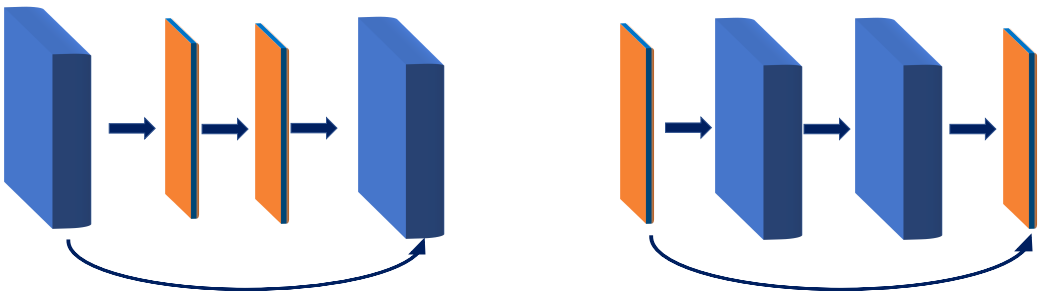


Figure 2.3: A regular depthwise separable convolution block [27].

This model uses *Inverted Residual blocks*, which are made of depth-wise convolutions (with the kernel size of 3x3), point-wise convolutions (with the kernel size of 1x1), both equipped with non-linear activations, and a final point-wise convolution with linear mapping similar to residual networks [19]. In contrast with regular residual networks, the layers with the lowest number of channels are connected via skip connections (known as inverted residual block). Figure 2.4 compares the conventional [20] with the inverted residual block.

Similar to MobileNet, Shuffle-Net [73] benefits from depthwise convolutions, but the pointwise 1x1 convolutions are replaced with *group convolutions* and *channel shuffle* modules. Shuffle-Net replaces the pointwise convolutions with group convolutions to decrease the computation cost. Besides, in canonical depth-wise convolutions, the mapping happens between equivalent channels, hurting the model's generalizability. Shuffle-Net addresses this issue by shuffling the input channels to each module. The generated feature map from the previous group is first divided into multiple subgroups, shuffled, and then fed to the next convolution group as input.



(a) Conventional residual block (b) Inverted residual block

Figure 2.4: (a) A traditional residual block in ResNet models [19] vs. (b) the inverted residual block used in MobileNet-V2 [48].

2.1.4 Knowledge Distillation

Knowledge distillation [24]—that can be used in combination with other model compression approaches—aims to preserve the accuracy of a powerful pre-trained teacher model in a smaller student model. A detailed review of knowledge distillation has been provided in the following Section 2.2.

2.2 Knowledge Distillation

Knowledge distillation (KD), as a model compression and training framework, has recently gained significant attention. KD is easy to implement, compatible with the current deep learning frameworks, and also can be used in combination with three other model compression techniques.

The motivation behind the canonical KD [24] comes from Caruana et al. [10], a framework to preserve a powerful ensemble's generalization into a small, constrained target model. A pre-trained powerful ensemble expands the training dataset by labeling the unlabeled data for optimizing the small target model.

KD is a training framework in which a huge pre-trained *teacher* model collaborates

in training a small *student* network [10, 24] by providing some hints (also known as *dark knowledge*) about the input sample and the way the teacher distinguishes different classes in the task. The trained student with this teacher-student framework has higher accuracy than the same model trained with only the ground truth labels. In canonical KD, dark knowledge is the teacher's soft probabilities. The soft probability distribution reveals valuable information regarding the similarity between different classes.

In the following, first, the mathematical notations and intuitions behind KD are discussed in Section 2.2.1. Then in Section 2.2.2, we discuss the potential reasons for KD's success. Finally, in Section 2.2.3, we explain two common categorizations of KD frameworks.

2.2.1 Mathematical Notations of Knowledge Distillation

The primary motivation behind KD is to train a small *student* network using the fully trained larger *teacher* model's soft outputs [24]. The student can approximate the teacher's behavior by learning complicated inter-class similarity patterns from the teacher's smooth probabilities.

In a classification task with M classes, and the dataset $\mathcal{D}$ with input-label pairs $(\mathbf{x}, \mathbf{y})$ and $\mathbf{y} \in \{1, \ldots, M\}$, the pre-trained teacher model T for each input $\mathbf{x}$ generates an output by:

$$\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x}) = \mathrm{softmax}\left(\frac{\mathbf{z}_T}{\tau}\right), \tag{2.1}$$

where $\mathbf{z}_T$ is the logits, i.e., the activation map before the softmax layer, and $\tau \geq 1$ is the temperature parameter. Temperature softens the teacher's generated probabilities to preserve the relative information between the classes, i.e., the class similarities. Note that the during teacher's training, $\tau = 1$. Like many other hyperparameters,

the magnitude of τ can considerably affect students' final accuracy. $\tau \leq 1$ leads to a more confident model, i.e., high probability (≈ 1) for one class and almost 0 for the rest of the classes, and large τ leads to the uniform probability distribution.

Similar to the teacher, the student network S predicts the probability for input $\mathbf{x}$ over classes by:

$$\mathrm{P}_{\mathrm{S}}^{\tau}(\mathbf{x}) = \mathrm{softmax}\left(\frac{\mathbf{z}_S}{\tau}\right). \tag{2.2}$$

The student S learns to generate the teacher T's probabilities using the KD loss objective:

$$\mathcal{L}_{\text{KD}}(S|T,\mathbf{x}) = \tau^2 D_{KL}\Big(\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x}) \,\Big|\Big|\, \mathrm{P}_{\mathrm{S}}^{\tau}(\mathbf{x})\Big), \tag{2.3}$$

where $D_{KL}(\mathbf{p} \;||\; \mathbf{q}) = \sum_i p_i \log \frac{p_i}{q_i}$ is Kullback-Leibler (KL) divergence distance of the *approximated distribution* $\mathbf{q}$ from the *target distribution* $\mathbf{p}$. Here, the KL distance measures how student S's final probability distribution $\mathrm{P}_{\mathrm{S}}^{\tau}(\mathbf{x})$ is different from the target teacher T's distribution $\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x})$ for input $\mathbf{x}$. Note that in the original paper of knowledge distillation [24], and some other distillation extensions, e.g., Fitnets [46], cross-entropy $\mathcal{H}\left(\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x}), \mathrm{P}_{\mathrm{S}}^{\tau}(\mathbf{x})\right)$, [1] has been used instead of the KL divergence $D_{KL}(\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x}) \;||\; \mathrm{P}_{\mathrm{S}}^{\tau}(\mathbf{x}))$ for knowledge distillation loss in Eq. 2.3. Both of these two metrics measure the difference between two distributions, and they can be used interchangeably because $\mathcal{H}\left(\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x}), \mathrm{P}_{\mathrm{S}}^{\tau}(\mathbf{x}) = D_{KL}(\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x}) \;||\; \mathrm{P}_{\mathrm{S}}^{\tau}(\mathbf{x})) + \mathcal{H}(\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x}))\right.$ and teacher's entropy $\mathcal{H}(\mathrm{P}_{\mathrm{T}}^{\tau}(\mathbf{x}))$ is independent of student's probability distribution. The weight τ^2 in Eq. 2.3 keeps the loss gradient magnitudes approximately constant when the temperature τ changes [24] in backward propagation.

For any given input $\mathbf{x}$, the student S learns from both the true label $y_{\texttt{true}}$ (hard

[1] Cross-entropy measures the difference between two probability distributions for a set of events $\mathcal{H}\left(\mathbf{p}, \mathbf{q}\right) = -\sum_i \mathbf{p}_i log \mathbf{q}_i$. Cross-entropy has been defined based on the notion of information entropy which refers to the amount of *surprise* or *uncertainty* in the outcome of a random event $\mathcal{H}(\mathbf{p}) = -\sum_i \mathbf{p}_i log \mathbf{p}_i$. As much as the outcome of a random variable is less confident, i.e., there is more surprise about it, the event would be more informative, and the entropy would be higher, and vice versa.

target) and the teacher T's output for input $\mathbf{x}$ (i.e., soft target) by minimizing the *total student loss*:

$$\mathcal{L}_{\text{S}}(S|T,\mathbf{x}) = \alpha\mathcal{L}_{\text{KD}}(S|T,\mathbf{x}) + (1-\alpha)\mathcal{L}_{\text{CE}}(S|\mathbf{x},y_{\text{true}}), \tag{2.4}$$

where $\mathcal{L}_{\text{CE}}(S|\mathbf{x},y_{\text{true}})$ is the conventional cross-entropy (CE) between student S's generated probability distribution $\text{P}_{\text{S}}^{\tau}(\mathbf{x})$ and the true label y_{true}.[2] Here, the weight α controls the trade-off between the two losses (i.e., the balance between hard and soft targets).

2.2.2 Why Knowledge Distillation Works?

Hinton et al. [24] speculated that KD outperforms regular CE mainly because of relative inter-class information. More specifically, the relative information between the wrong classes empowers the KD. For instance, given an input $\mathbf{x}$, if a classifier assigned similar probabilities to classes car and truck and a very different probability to class apple, the similar probabilities for classes car and truck would reveal that the teacher has exploited common patterns between these two classes. In contrast, the apple would be recognized as a less related class to car and truck classes. This inter-class information is also referred to as *dark knowledge* [39, 66].

KD has shown mediocre performance in classification tasks with a few classes. This behavior is in line with the dark knowledge hypothesis, i.e., few classes lead to insufficient inter-class information; therefore, canonical KD could not be very effective. Subclass distillation [40] is a KD extension where the teacher expands the number of classes by dividing each class into multiple sub-classes. Distillation using the sub-classes leads to more powerful students in a shorter training time than

[2] As we mentioned earlier, for this part of student S's training (i.e., CE loss $\mathcal{L}_{\text{CE}}$), we always set $\tau = 1$ as same as the original Knowledge distillation approach [24].

canonical KD.

2.2.3 Categories of Knowledge Distillation

KD can be categorized into various divisions from different points of view. However, two common categorizations differentiate between KD approaches based on the answers to the following questions:

1. *how to define the teacher and student?*

2. *what is the knowledge for distillation?*

The first question divides KD variants into two groups. Canonical KD requires a fully trained teacher model to train the student, i.e., the teacher should be optimized in advance. The KD variants that need a fully optimized teacher model are referred to as *offline* KD frameworks. On the other hand, some KD frameworks work without a pre-optimized teacher. This category is called *online* KD. Online approaches do not define teacher or student roles; instead, a cohort of models (known as *peers*) collaboratively train each other.

The second question discriminates between KD variants based on the source of knowledge for distillation, e.g., Canonical KD and some other KD frameworks use the final probabilities for distillation [11, 24, 39], while others benefit from the teacher's intermediate representations [46, 59, 65, 70, 72].

Chapter 3

Related Work

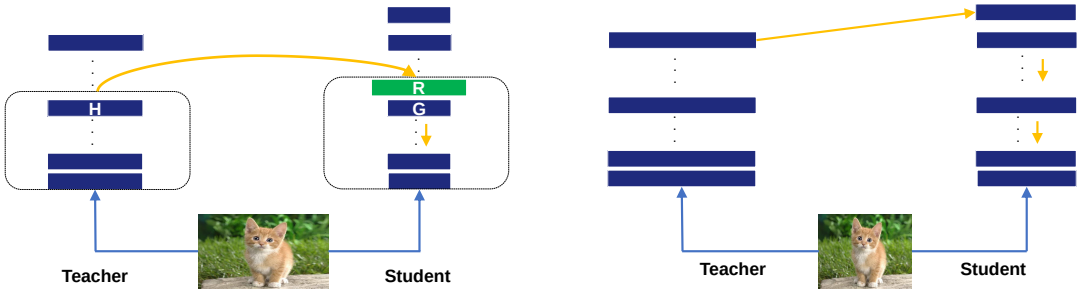
We review the related work in knowledge distillation, intermediate knowledge distillation, and knowledge distillation with the help of auxiliary classifiers. We also explain their strengths and weaknesses.

3.1 Knowledge Distillation

As discussed in Section 2.2.3, one can categorize KD variants based on the definition of teacher-student models and the source of knowledge. In this Section, we briefly describe some of the well-known KD frameworks.

Canonical KD [24] uses the teacher's soft final probabilities to train the student. Some distillation frameworks improve the conventional framework and address some of its barriers by defining new sources of knowledge for distillation.

FitNets [46] trains a deep and thin student model using a wide and shallow pre-trained teacher in two consecutive steps. First, a fraction of the student up to an intermediate *guided* layer is optimized to generate feature maps similar to the teacher's *hint* layer. This step might require an auxiliary regressor on top of the guided layer



(a) Stage 1 (b) Stage 2

Figure 3.1: FitNets- In first stage Stage (a), a sub-model up to *guided* layer (G) is optimized by the help of *regressor* (R)–if needed– and L_2 loss objective. In the second stage (b), the whole student is trained using teacher's soft probabilities.

to match the incompatible representations. Given a model S with a regressor r, the loss objective for this stage of training is:

$$\mathcal{L}_{\text{s}}(\mathbf{W}_{Guided}, \mathbf{W}_r) = \frac{1}{2}\Big|\Big|F_h(\mathbf{x}, \mathbf{W}_{Hint}) - F_r(F_g(\mathbf{x}, \mathbf{W}_{Guided}))\Big|\Big|_2, \tag{3.1}$$

where F_h, F_g, and F_r are the sequential functions up to hint, guided, and regressor layers, respectively. This stage tries to minimize the Euclidean distance (i.e., L_2 *norm*) between two intermediate representations. The second stage of training is canonical KD between the whole teacher and the student models. Figure 3.1 depicts the general training pipeline in the FitNets.

FitNets outperforms canonical KD occasionally because its performance depends on many new hyperparameters; e.g., the ideal structure for the regressor or the optimal hint-guided layer pair. Naive selection of the regressor could lead to information loss during distillation. Finally, while the final probabilities could be evaluated using the ground truth labels, there is no verified benchmark for intermediate representations, and this would be problematic since even the most accurate teacher models could infer wrongly.

Attention distillation (AT) [70] trains the student by using teacher's *attention maps*, which are the activation maps in an intermediate layer, averaged along the channel dimension. Distilling the teacher's attention map to the student's equivalent layer helps the student layer focus on similar areas in the input sample as the teacher layer does. AT improves the student since the powerful pre-trained teacher layer knows what areas in the input image are more important to the student model (i.e., which areas contain more discriminative features). This framework partially solves the dimensionality mismatch; however, the teacher and the student still should have the same height and width in their intermediate target layers.

Consider a student model S, the teacher model T, and i that denotes the indices for attention maps distillation between the two models, the total loss objective is:

$$\mathcal{L}_{\texttt{S}}(\mathrm{S}|\mathrm{T},\mathbf{x}) = \mathcal{L}_{\texttt{CE}}(\mathrm{S}|\mathbf{x}, y_{\texttt{true}}) + \frac{\beta}{2}\left|\left|\sum_{j\in i}\left(\frac{Q^j_{\mathrm{S}}}{\left|\left|Q^j_{\mathrm{S}}\right|\right|_2} - \frac{Q^j_{\mathrm{T}}}{\left|\left|Q^j_{\mathrm{T}}\right|\right|_2}\right)\right|\right|_p, \tag{3.2}$$

where Q_{S} and Q_{T} are teacher's and student's attention maps, respectively[1].

Yim et al. [65] define the information flow between the layers as the source of knowledge for distillation. The information flow is the inner product between the representations of two selected layers. For training the student S by using N information flow matrices from the trained teacher T, the loss objective is:

$$L_{FSP}(\mathbf{W}_{\mathrm{T}}, \mathbf{W}_{\mathrm{S}}) = \frac{1}{\mathrm{N}}\sum_{x}\sum_{i=1}^{\mathrm{N}}\lambda_i\left|\left|(\mathbf{G}^{\mathrm{T}}_i(\mathbf{x}, \mathbf{W}_{\mathrm{S}}) - \mathbf{G}^{S}_i(\mathbf{x}, \mathbf{W}_{\mathrm{S}})\right|\right|^2_2, \tag{3.3}$$

where it is the Euclidean distance between matrices $\mathbf{G}$, the information flow matrix between two layers in either teacher or student. It is worth noting that the information flow matrices of the student and the teacher should have the same height and width;

[1] $||\mathbf{x}||_p$ refers to the p-norm of the vector $\mathbf{x}$. Given the vector $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, $||\mathbf{x}||_p = (\sum_{i=1}^{n} |\mathbf{x}_i|^p)^{\frac{1}{p}}$

therefore dimensionality mismatch problem limits this framework as well.

Heo et al. [23] use a pre-trained teacher's *activation boundaries (AB)* as knowledge source. In contrast with the previous work (e.g., [46, 65, 70]), this framework transfers the intermediate neurons' activation status (i.e., whether a neuron is activated or not). The paper shows that the canonical KD neglects small intermediate neurons' responses. These small activations are crucial since they present complex input samples. Using L_1 norm, *activation transfer loss* (Eq. 3.4) aims to amplify the neuron responses near the activation boundaries for each input $\mathbf{x}$ by applying an element-wise indicator on each neuron, and transfer the amplified response to the student:

$$\rho(\mathbf{x}) = \begin{cases} 1, & \text{if} \quad \mathbf{x} > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{3.4}$$

$$L(I) = \Big|\Big|\rho(\mathrm{T}(I)) - \rho(\mathrm{S}(I))\Big|\Big|_1 \tag{3.5}$$

One drawback of this framework is that the AB transfer loss is not differentiable, and other functions should approximate it in the backward propagation. Moreover, similar to AT, AB is only applicable to intermediate layers with the same height and width. Finally, AB is only effective in ReLU-based neural networks.

Fu et al. [16] propose a distillation framework in which, instead of transferring the teacher's intermediate knowledge to the student, it directly replaces the student's intermediate blocks with the teacher's trained blocks.

Similar to [16], *progressive grafting training* [50] swaps student's blocks with the teacher's pre-trained blocks in two stages. The motivation behind this approach is the shortage of labeled training data which fails the regular CE. First, both teacher and the student are decomposed into the same number of blocks. Then, each student

block is replaced with the equivalent teacher's block, goes through training. Then, when all of the student's blocks are individually trained, we progressively graft the trained students' blocks with the teacher's until the full swap of two models.

In all of the distillation approaches mentioned earlier, the student model is trained deterministically. In contrast, some distillation frameworks use concepts related to adversarial training [22, 60, 61]. Wang et al. [62] propose a distillation framework based on generative adversarial networks (GANs). In this approach, the student is the small-size generator that learns to generate predictions similar to the pre-trained teacher model. The discriminator, known as the *teacher assistant*, is responsible for recognizing whether a received representation is from the student or the teacher. The student and the teacher assistant simultaneously learn how to generate more teacher-alike probabilities and more accurate discrimination, respectively.

Some other KD variants have improved the student's robustness by noise injection in the teacher's outputs [49] or aggregating multiple teachers' knowledge by voting mechanisms [67]. Although these frameworks positively contribute to training a more accurate student, they are not directly related to our approach.

Regardless of the knowledge definition or distillation procedure, a fully-trained teacher is necessary among all of the mentioned frameworks. Offline KD has shown significant improvements in training student models. However, the need for an already trained teacher limits its use-cases since a pre-trained teacher is not always cheaply available. Online KD has raised attention due to the mentioned barriers.

Co-distillation [3] replaces the expensive ensembles and canonical KD training with an online distillation approach. Co-distillation trains a group of models with identical architecture, which only vary in their initialization. Each model is trained by using a subset of a dataset. At some checkpoints, models exchange their computed weights to reproduce each others' predictions. Weight exchanging is beneficial since

weights are more robust against expiration, i.e., only the predictions of a small subset of training data can be dramatically changed during the training. However, this approach could only train identical models, which is a limiting factor. Besides, it forces each model in the ensemble to generate similar probability distribution, limiting the diversity of the ensemble.

Hierarchical distillation (HD) [47] addresses the limited diversity in ensembles by establishing a *hierarchical neural ensemble (HNE)*. This framework's motivation is *any time inference* applications, where the available resources during the inference time are not predictable in advance. Models in HNE share a subset of parameters, which leads to a lighter ensemble for inference time. HD uses the whole ensemble as the teacher for a subset of models in the ensemble. Therefore, HD optimizes each student while not forcing the student to produce similar outputs to every other model in the ensemble, which preserves the diversity in the ensemble.

Deep mutual learning (DML) [74] extends the concept of online KD to *cohorts* of various models, identical or heterogeneous, that co-teach each other. Consider a cohort of K models, the model S is optimized with the loss objective:

$$\mathcal{L}_{\mathrm{S}}(\mathrm{S}, \mathbf{x}) = \mathcal{L}_{\mathtt{CE}}(\mathrm{S}|\mathbf{x}, y_{\mathtt{true}}) + \frac{1}{\mathrm{K}-1} \sum_{l=1, l\neq \mathrm{S}}^{\mathrm{K}} D_{KL}(\mathbf{p}_l \,||\, \mathbf{p}_{\mathrm{S}}), \tag{3.6}$$

where $\mathrm{K}-1$ teachers collaborate in the training the target peer S in the cohort. According to the experiments, multiple teachers can train more accurate students than the ensemble of all the teachers. These experiments show that ensembling might erase some useful inter-class information which could be beneficial for distillation.

Collaborative teacher-student learning via multiple knowledge transfer (CTSL-MKT) [52] combines DML with self-distillation and relation-based knowledge transfer. Self-distillation [68] refers to KD using the student-generated outputs, i.e., when the

teacher and student refer to the same model. Self-distillation can stabilize the model and mitigates the negative impact of other peers' wrong signals because the peers could generate less confident outputs during DML. CTSL-MKT uses the *angle-wise* and *distance-wise* loss objectives to transfer relation-based knowledge to the student, i.e., the student learns to map input instances to a high-dimensional space in a way that the distance and the angle between these mappings are similar to the teacher's mappings of the same input samples.

3.1.1 Knowledge Distillation vs Capacity Gap

Canonical KD trains a student model using a powerful pre-trained teacher network. Usually, The student trained via KD preserves the teacher's generalization power and has a higher testing accuracy than the student trained regularly from ground truth labels. However, when the teacher's and the student's model complexity are different (*capacity gap* [39]), KD loses its efficiency and trains weaker student models. In such cases, the student can not mimic the teacher's behavior, i.e., the teacher's knowledge is complicated for the small student, and KD acts as a powerful regularizer for the student.

One can improve KD's efficiency by bridging the capacity gap. Mirzade et al. [39] mitigate the negative impact of the huge capacity gap using multiple *teacher assistant* models between the teacher and the target student model (TAKD). The medium-power intermediate teacher assistant model can mimic the teacher's complicated representations while generating digestible knowledge for the small student (see Figure 3.2). Depending on the size of the capacity gap, one can establish a chain of teacher assistant models. Each teacher assistant is optimized by the teacher's hints (or previously trained teacher assistant) and trains the next student (weaker teacher assistant or the primary student). TAKD improves the final student's accuracy as it

provides more understandable representations for the final student.

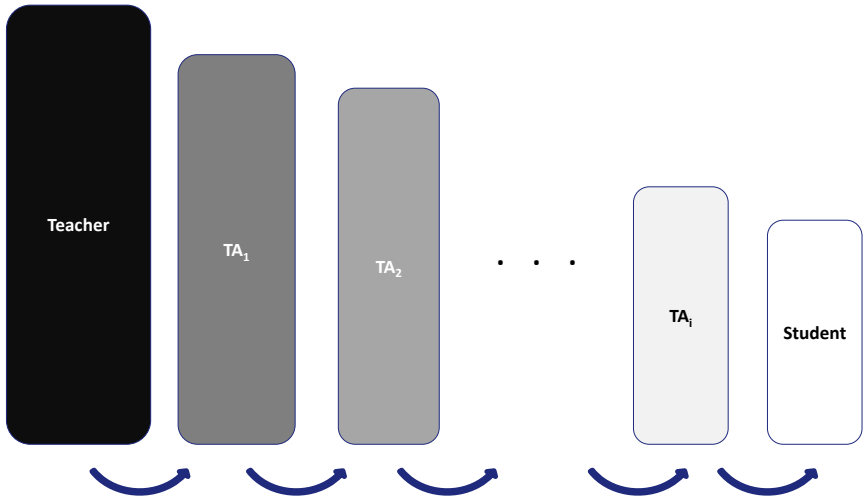


Figure 3.2: Training a student model with the help of i teacher assistant models.

Passalis et al. [42] introduce a distillation framework by combining the *auxiliary teacher* model and *critical learning*. Critical learning indicates that the most influential period in training a deep neural model is the first few epochs since the neural paths are established in this period, and after that, new paths are barely created.

Residual error-based knowledge distillation (RKD) [18] defines the teacher assistant models differently. Since the last fully connected layers are identical in both teacher and the student, RKD bridges the capacity gap in intermediate layers. The teacher assistant model fills the capacity gap by matching the feature representations and simultaneously learns the residual error between the teacher and the student. In the inference time, the combination of both student and the teacher assistant does the classification.

In contrast with TAKD and RKD, some frameworks directly use the teacher model to mitigate the capacity gap's negative impact, e.g., RCO [30] uses multiple replicas of partially-optimized teacher model to create a learning route tutorial for the student.

Since the teacher in these replicas has not reached its highest performance, the hints are more digestible for the student. In a similar approach, *early-stopping knowledge distillation (ES-KD)* [11] lessens the capacity gap impact by training the powerful teacher model for a shorter time than the regular.

The capacity gap could also hurt peers' accuracy in an online distillation setting. *Evolutionary knowledge distillation* [71] proposes an online distillation framework, in which peers not only teach each other with their final outputs but also they use intermediate classifiers to resemble each others' intermediate representations as well.

Aguilar et al. [1] use intermediate distillation for transformer-based linguistic models. These models are cumbersome, and canonical KD cannot transfer all the important learned linguistic properties to the student. This framework uses KL-divergence and cosine loss to distill every two teacher's blocks into a single student's block.

Although these approaches have one common motivation as this thesis, i.e., dealing with the capacity gap's negative impact on KD, they have not addressed some challenging issues, e.g., TAKD [39] the process of consecutively training teacher assistants could be expensive, and also the effectiveness of the whole process depends on many hyperparameters. Bridging the capacity gap using partially-trained teachers (ES-KD [11]) still misses the teacher's intermediate knowledge. Besides, it requires training the teacher in advance and storing multiple partially trained replicas. Similarly, RCO [30] requires re-training the teacher model for recording the anchor points, which is not efficient when the teacher is already trained.

3.1.2 Relation to This Thesis

Our proposed approach (EKD) is built on the strengths of FitNets [46] and distillation with teaching assistants [39]. Like FitNets, EKD leverages the teacher's intermediate representations to supervise the student during training. EKD solves the incompatible

dimensionality issue by deploying intermediate classifier heads. As with teaching assistants, EKD deploys medium-capacity models to facilitate KD from large-capacity teachers to small-capacity students. However, these medium-capacity models are computationally cheaper than independent teacher assistants as they could be easily built and quickly trained by recycling the pre-trained teacher.

3.2 Multi-Classifier Heads (MCH)

Auxiliary intermediate classifier heads have been extensively used in deep learning for various applications. *Inception* [55] uses multiple auxiliary intermediate classifiers as regularizers. The added classifiers amplify the gradient magnitudes in backward propagation, which decreases the vanishing gradient effect, making it possible to train deeper models more accurately.

Multi-scale dense networks (MSDNet) [29] establishes multiple intermediate classifiers for faster testing time. This approach borrows the concept of *feature reusing* from deeply supervised nets (DSN) [34] and uses it with auxiliary classifiers. Each classifier (either the main classifier or the added ones) can efficiently participate at inference time based on the input image's difficulty, i.e., the shallower classifiers predict easier input examples without wasting more computation resources while deeper classifiers classify complex samples. Given an input image, The image will be passed through the classifiers from the shallowest one to the main one at the end; as soon as a classifier's prediction reaches the confidence threshold, the model would ignore deeper layers.

Similar to MSDNet [29], *Conditional Deep Learning (CDL)* [41] decreases the inference time by using a cascade of linear layers attached to each convolutional layer in the model. As soon as one of the intermediate classifiers satisfies the confidence

threshold, the model could skip propagation to the deeper layers. In this way, the inference computation energy and time would be proportional to the difficulty of the input sample.

Early-Exiting Framework (ELF) [15] benefits from intermediate classifier branches in both training and inference time. More specifically, this approach tries to address *long-tailed* data distribution classification, where except for a few classes, most of the remaining are in the minority in the dataset. Given a model with K exit classifiers at various depths denoted by $\{C_1, C_2, ...C_{\mathrm{K}}\}$, If the classifier $j \in [1, \mathrm{K}]$ correctly classified input sample $\mathbf{x}$ with a confidence higher than a threshold in less than a predefined time slot, the model would no longer propagate the input through the network. The model would follow the same process in the inference time.

BranchyNet [57] is a multi-exit classifier similar to [15, 29] that learns the optimal confidence threshold. The confidence metric in BranchyNet is the *entropy*.

Phuong et al. [44] train the shallow classifiers using canonical KD to improve the intermediate classifiers' accuracy in a multi-exits-classifier model. *Multi-Self-Distillation learning (MSD)* [38] uses the intermediate classifiers and KD to train a more accurate *any-time inference* model.

Alain and Bengio [2] used internal classifiers (known as *probes*) as debugging tools for very deep neural models. Probes could be useful when conventional loss or accuracy values do not help debug the model. They interestingly analyzed the correlation of the intermediate heads to each other in terms of *mutual information* from the information theory points of view.

Intermediate classifiers could also prevent the network from misclassification. *Shallow Deep Networks (SDN)* [25] has been introduced with this motivation that mapping simple input data to complex high-dimensional representation could lead to misclassification, i.e., *overthinking*. SDN is a set of intermediate classifiers attached at different

model depths. In Chapter 5, We observe that a simple input image could be correctly classified by the shallowest classifier head, while the most powerful head, i.e., the main classifier, cannot predict correctly.

One can benefit from intermediate classifier heads to improve canonical KD's performance. In the following Section, we review some KD variants that use intermediate classifiers.

3.2.1 Knowledge Distillation with Internal classifiers

One way to address the barriers in distilling teacher's intermediate knowledge is using intermediate classifier heads. Intermediate classifiers map the teacher's intermediate knowledge to probability distribution among the task's classes, which is understandable for the student. Moreover, shallower classifiers in the teacher model could play the teacher assistant's role, mitigating the negative impacts of the large capacity gap on KD. In general, we could divide the KD frameworks with intermediate classifiers into four general categories:

(1) One-to-One. In this category, each teacher's classifier (either auxiliary or the primary one) transfers its knowledge to the same-stage classifier in the student model. Regardless of the defined loss objective and the architecture for auxiliary classifier modules, this approach requires adding the same number of auxiliary modules to both teachers and students. Task-oriented feature distillation (TOFD) [72] is an instance of this category. In TOFD, the student's classifiers are trained with canonical KD (*logit distillation loss*), regular CE (*task loss*), L_2 loss to match student's intermediate representations before each classifier with the equivalent teacher's representation (*features distillation loss*), and last but not least, *orthogonal loss* in feature resizing layer to reduce information loss.

Patient Knowledge Distillation (PKD) [53] uses a similar approach but in natural language processing. More specifically, PKD transfers the teacher's intermediate knowledge between every k internal layer from a BERT teacher to the equivalent layers in a BERT student model.

This category requires more computation and time since both models should be equipped with intermediate classifiers. The symmetric nature of this group also increases the number of hyperparameters. Last but not least, one-to-one approaches could not properly address the large capacity issue.

(2) Many-to-one. EKD is an instance of many-to-one distillation. Many-to-one approaches can distill the teacher's intermediate knowledge from as many intermediate layers as possible to the student's primary classifier. This category usually has less overhead than many-to-many or one-to-one since there is no need to decide on the architecture and location of auxiliary classifiers in the student model. This characteristic makes distillation possible between heterogeneous pairs. Besides, the teacher's shallower classifier could bridge the capacity gap between the teacher and the student.

(3) One-to-Many. In this category, one classifier on the teacher side supervises multiple classifier heads on the student side. Phuong et al. [44] propose a distillation-based training framework for multi-exit classifier models. The proposed framework targets multi-exit classifier models, useful when the time and computational budget are not clear in advance. This category does not address the main motivations of this thesis, i.e., capacity gap issue and improving canonical KD by using the teacher's intermediate representations.

(4) Many-to-Many. This category refers to frameworks in which both teacher and the student have multiple classifiers. Knowledge Transfer via *Dense Cross-Layer Mutual-Distillation (DCM)* [64] is an online KD approach that outperforms the canon-

ical deep mutual learning (DML) by using auxiliary classifiers in all the models in the cohort. Each classifier is optimized using regular CE and KD loss between the equivalent layer's classifier (*same-staged bidirectional KD*) and different layers' classifiers (*cross-layer bidirectional KD*). Figure 3.3 illustrates the DCM training in a cohort of two models.

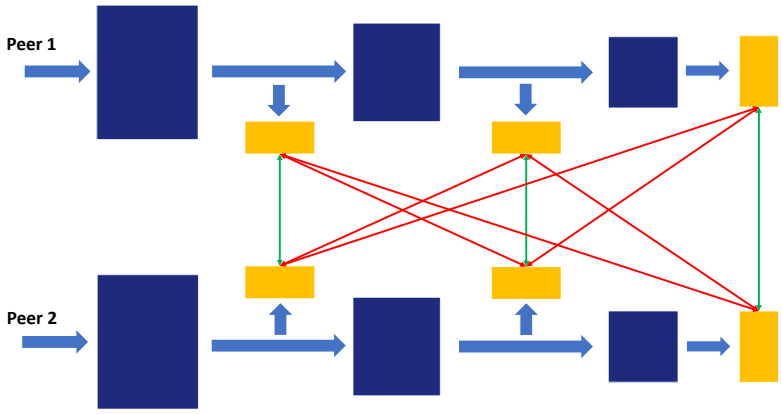


Figure 3.3: Dense Cross-Layer Mutual-Distillation (DCM).

Although naively comparing offline and online KD is not practical due to their different characteristics, motivations, and applications, DCM is a time-consuming training approach. According to the paper, DCM is almost 50% slower than regular DML, which is a negative factor. Besides, similar to many-to-many approaches, this category requires tuning many new hyperparameters that increase the overall training overhead.

3.2.2 Relation to This Thesis

Enhanced Knowledge Distillation by Auxiliary Classifiers (EKD) is a many-to-one distillation framework that improves student's accuracy by exploiting the teacher's

intermediate knowledge using multiple intermediate classifier heads. The mounted classifier heads, in addition to the primary classifier, collaboratively train the student. This simultaneous collaboration provides more intuition about the class similarities in the training dataset for the student. Besides, shallower classifiers bridge large capacity gaps between the teacher and student. Also, EKD could distill the knowledge between completely heterogeneous models since it does not require symmetric auxiliary classifiers in both models.

3.3 Curriculum Learning

Curriculum learning [6] refers to the training scheme that benefits from training samples ordered according to their difficulty. Education systems provide progressive curricula from fundamental concepts to more advanced material. Bengio et al. [6] show that ordering the training samples from simple to complex could improve the student's accuracy and decrease the training time.

Curriculum learning has been the intuition behind many distillation frameworks. RCO [30] uses teacher replicas, from partially trained teachers to the fully optimized replica. Partially teacher replicas behave similar to the final student, providing easier hints for the distillation. ES-KD [11] provides more digestible hints for the student by early-stopping the teacher model during the training.

3.3.1 Relation to This Thesis

Our work can loosely be considered as a type of curriculum learning. Instead of ordering the input data based on their difficulty, we simultaneously provide both easy and complicated representations via multiple intermediate classifier heads to the student. These representations with different levels of complexity allow the student

to learn from the different teachers with diverse representation capacities.

Chapter 4

Approach

In this Chapter, we present the main contributions of this thesis. The primary motivations behind this work are: (1) introducing a distillation framework that can improve the student using the teacher's rich intermediate representations, and (2) improving KD when there is a large capacity gap between the teacher and the student. In this chapter, we first define the problems and drawbacks of currently used distillation frameworks, then we explain our proposed approach, *Enhanced Knowledge Distillation by Auxiliary Classifiers (EKD)*.

4.1 Problem Statement

Consider a classification task between M classes with a training set of N pairs of ($\mathbf{x}$,y), i.e., $X = [\mathbf{x}_i]_{i=1}^{N}$, and $Y = [y_i]_{i=1}^{N} \in [1, 2, ..., M]$. Given a fully-trained teacher model T and a student model S, one can improve student S with the help of the teacher's smooth outputs. In canonical KD, the teacher's softened probabilities act as an extra hint for the student model to improve its accuracy. Although this approach has shown better performance than the regular CE, KD can not efficiently distill all the available

teacher's knowledge to the student. For this purpose, various approaches have tried to increase the canonical KD's efficiency using different techniques or defining new sources of knowledge. A group of these approaches uses the teacher's intermediate representations as a source of knowledge for distillation.

However, these intermediate KD frameworks suffer from multiple issues. First of all, there is no benchmark for intermediate representations, i.e., in contrast with canonical KD and CE that use ground truth labels for evaluating the final probabilities, teacher's intermediate representations are not easily interpretable.

Besides, the student could not understand the teacher's intermediate representation if their dimensionality differs from the student. In canonical KD, distillation happens between two probability distributions among the same number of classes, i.e., both models generate final outputs in the same semantic space. While for incompatible intermediate representations, first, one should match the dimensions before distillation. This could happen by adding an extra regressor module [46] for resizing the feature maps or averaging along one dimension [70]. However, these mapping techniques usually lead to information loss, reducing KD's efficiency. Besides, mapping's performance highly depends on properly setting multiple hyperparameters, e.g., the architecture of the added module, leading to fragile functionality.

Moreover, most intermediate KD approaches do distillation symmetrically, i.e., layer-wise distillation [46, 65, 70, 72]. This limits the possible teacher-student pair options, e.g., distillation between heterogeneous teacher-student pairs would be impossible or at least not practical.

Finally, canonical KD could hurt the student's accuracy when the teacher and student model complexity are different, i.e., a massive capacity gap [39] between the two models. In this case, the student finds the teacher's hints very complicated.

4.2 Preliminaries

This Section describes knowledge distillation and intermediate classifier heads, which are the main components of our proposed framework.

4.2.1 Knowledge Distillation

Given a classification dataset containing N input samples and M classes, i.e., $X = \{\mathbf{x}_i\}_{i=1}^{N}$, and $Y = \{y_i\}_{i=1}^{N} \in \{1, 2, ..., M\}$, one can train a small student model S with the help of a powerful pre-optimized teacher model T, that has been trained on the same task, in advance.

Given a random training sample $\mathbf{x}_i$, the student classifies the input image belonging to class i among M classes in the task using the softmax activation function:

$$P_{\mathbf{S}}^{\tau}(\mathbf{x})_i = \frac{\frac{e^{z_i}}{\tau}}{\sum_{j=1}^{M} \frac{e^{z_j}}{\tau}} \tag{4.1}$$

where the z_j is the student's output vector representation of input image $\mathbf{x}_i$ (*logits*). In the regular supervised setting, the model is optimized using ground truth labels and CE loss objective. By doing so, the model is penalized for wrong predictions:

$$L_{CE}(\mathrm{S}|X, Y) = \mathcal{H}(Y \mid\mid P_{\mathbf{S}}) = -\frac{1}{N} \sum_{i}^{N} y_i \log P_{\mathrm{S}}^{\tau}(\mathbf{x}_i) \tag{4.2}$$

The hyperparameter temperature (τ) is responsible for adjusting the model's confidence, i.e., the higher the temperature is, the less confident the student would be. During the regular CE training $\tau = 1$ (see Eq. 4.2).

When a pre-trained powerful teacher model T is available, one can benefit from the teacher's learned knowledge to improve the student's accuracy. The teacher's supervision provides valuable information about the input samples and how the teacher

classifies them. By applying higher temperature values, these intra-class relationships will be more detectable for the student. Given the pre-trained teacher T, and temperature τ, the student is trained by optimizing the following loss objective:

$$L_{KD}(\mathrm{S}|\mathrm{T}, X) = \tau^2 \mathcal{H}(P_{\mathrm{T}}^{\tau} \mid\mid P_{\mathrm{S}}^{\tau}) = -\tau^2 \frac{1}{N} \sum_{i}^{N} P_{\mathrm{T}}^{\tau}(\mathbf{x}_i) \log P_{\mathrm{S}}^{\tau}(\mathbf{x}_i) \tag{4.3}$$

where $\mathcal{H}(P_{\mathrm{S}}^{\tau} \mid\mid P_{\mathrm{T}}^{\tau})$ is the cross-entropy between the softened teacher and student probabilities. τ^2 takes care of the increased temperature in the backward propagation and gradients calculation. It is worth noting that in many distillation frameworks, $\mathcal{H}(P_{\mathrm{T}}^{\tau} \mid\mid P_{\mathrm{S}}^{\tau})$ has been replaced with the KL-divergence distance $D_{KL}(P_{\mathrm{T}}^{\tau} \mid\mid P_{\mathrm{S}}^{\tau})$. KL-divergence and CE can be used interchangeably because:

$$\mathcal{H}(P_{\mathrm{T}}^{\tau} \mid\mid P_{\mathrm{S}}^{\tau}) = D_{KL}(P_{\mathrm{T}}^{\tau} \mid\mid P_{\mathrm{S}}^{\tau}) + \mathcal{H}(P_{\mathrm{T}}^{\tau}) \tag{4.4}$$

where $\mathcal{H}(P_{\mathrm{T}}^{\tau})$ is the teacher's entropy and is independent of the student's probability distribution. One can benefit from both *hard labels* (ground truth labels) and *soft labels* to train a more accurate student:

$$L_{total}(\mathrm{S}|\mathrm{T}, X, Y) = \alpha L_{CE}(\mathrm{S}|X, Y) + (1 - \alpha) L_{KD}(\mathrm{S}|\mathrm{T}, X) \tag{4.5}$$

where α is a balancing weight between two loss functions.

4.2.2 Intermediate Classifier Heads

Intermediate classifier heads refer to the auxiliary modules that generate the class probabilities from the model's intermediate representations. As it is discussed in Chapter 3, these intermediate classifiers have numerous usages, such as debugging the deep neural model [2], dynamic inference time [44, 57], regularization [54], and knowledge

distillation [59, 64, 72]. We use intermediate classifier heads to improve the canonical KD, especially when there is a massive capacity gap between the teacher and the student.

The auxiliary classifier heads map the teacher's high dimensional intermediate representations, a rich source of data features and class similarities, to the probability distribution among the classes, which is understandable for student S. Each classifier head generates a different representation of the same input sample. This provides a diverse curriculum (easy and complex) of inter-class similarities. Also, shallower classifiers could address the large capacity gap issue by generating easier hints that the small student could understand.

4.3 Enhanced Knowledge Distillation by Auxiliary Classifiers

Our proposed approach combines canonical KD and intermediate classifier heads. This framework is an offline distillation framework (i.e., the teacher model is already fully trained). Figure 4.1 depicts the general scheme of EKD.

In a classification task over M classes, consider a fully-optimized teacher T and a student S. First, the teacher is decomposed into $\mathrm{K}+1$ separate modules, each containing a couple of teacher's trained layers. Then, K classifier heads are mounted on top of the first K separated modules at different depths (the last module already contains the main classifier). These mounted intermediate classifier heads are denoted as $\{C_i\}_{i=1}^{\mathrm{K}}$. At the end of this step, the teacher model has $(\mathrm{K}+1)$ classifier heads.

The number of added intermediate heads could be different based on the available training budget and the teacher's architecture. For this thesis, we use the most common deep neural models in computer vision that follow a modular architecture,

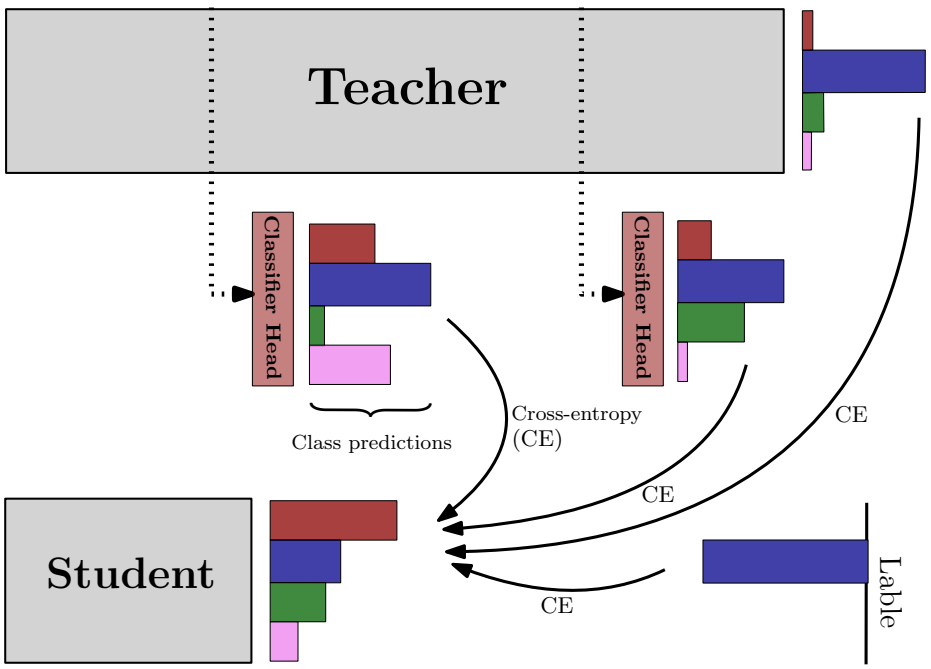


Figure 4.1: The proposed Enhanced Knowledge Distillation by Auxiliary Classifiers (EKD) framework. The teacher is equipped with multiple intermediate classifier heads at various depths. These new classifier heads are trained while the backbone teacher is frozen. A cohort of classifiers, including all the intermediate classifiers and the original teacher, simultaneously supervises the student.

e.g., residual networks [19], wide-ResNets [69], and VGG models [51]. We add an intermediate head after each teacher's module. Since the main teacher is already fully optimized, there is no need to retrain this gigantic model; therefore, we freeze the teacher model during intermediate heads' fine-tuning. Fine-tuning would not be costly because the number of learnable parameters in the added classifiers is not considerable. Given a teacher T with K fine-tuned intermediate classifiers, the student S is optimized using the diverse hints from $\mathcal{T} \in \{C_1, C_2, ..., C_{k+1}\}$:

$$\mathcal{L}_{\text{EKD}}(\text{S}|\mathcal{T}, \mathbf{x}) = \frac{1}{K+1} \sum_{V \in \mathcal{T}} \mathcal{L}_{\text{KD}}(\text{S}|\text{V}, \mathbf{x}), \tag{4.6}$$

where the KD loss $\mathcal{L}_{\text{KD}}(S|M, \mathbf{x})$ is computed by Eq. 2.3. This loss is the average of KD

loss values between student S and teachers in cohort $\mathcal{T}$ (including the original teacher and K intermediate heads). One can interpret the canonical KD as a particular case of EKD, where there are no mounted intermediate classifiers, i.e., $K = 0$. Given input $\mathbf{x}$, the total loss objective for student S would be:

$$\mathcal{L}_{\mathtt{S}}(\mathrm{S}|\mathcal{T}, \mathbf{x}) = \alpha \mathcal{L}_{\mathtt{EKD}}(\mathrm{S}|\mathcal{T}, \mathbf{x}) + (1 - \alpha)\mathcal{L}_{\mathtt{CE}}(\mathrm{S}|\mathbf{x}, y_{\mathtt{true}}), \tag{4.7}$$

where $\mathcal{L}_{\mathtt{CE}}(S|\mathbf{x}, y_{\mathtt{true}})$ is a conventional CE loss between student S's probabilities and the true label $y_{\mathtt{true}}$, and α is the trade-off weight between the two loss objectives.

Chapter 5

Experiments

This Chapter provides the reports and interpretations of various experiments that we have done to evaluate Enhanced Knowledge Distillation by Auxiliary Classifiers (EKD). All the evaluations have been done using the PyTorch framework [43] on a GeForce GTX 1080 Ti GPU. The implementation codes and the used hyperparameters are available in our GitHub repository. [1]

5.1 Evaluation Datasets

We report the obtained results on three well-known image classification datasets: CIFAR-10, CIFAR-100 [32],[2] and Tiny-ImageNet.[3] Table 5.1 provides the general statistics of the mentioned datasets. CIFAR-10 contains $60,000$ (5000 training, and 1000 testing samples per class) 32x32 RGB images for 10 classes. CIFAR-100 includes the same-sized samples as CIFAR-10 while providing 100 classes, with 500 training samples and 100 testing examples per class.

In order to evaluate our proposed framework on a more challenging dataset, we

[1] https://github.com/aryanasadianuoit/Distilling-Knowledge-via-Intermediate-Classifiers
[2] https://www.cs.toronto.edu/ kriz/cifar.html
[3] https://www.kaggle.com/c/tiny-imagenet

Dataset	Spatial Size	# Classes	# Train	# Test
CIFAR-10	32x32	10	$50,000$	$10,000$
CIFAR-100	32x32	100	$50,000$	$10,000$
Tiny-ImageNet	64x64	200	$100,000$	$10,000$

Table 5.1: The general statistics of the used datasets.

were eager to use ImageNet dataset [13]; however, we have not been permitted to access this dataset at the time of writing. Instead, we used Tiny-ImageNet, a down-sampled subset of ImageNet, containing $110K$ RGB images in the size of 64x64 categorized into 200 classes.

5.1.1 Preprocessing

For CIFAR-10 and CIFAR-100 [32], we followed standard data augmentation and pre-processing steps in [39, 70, 74], which includes horizontal flips, 4 pixels padding with the reflection of the original image, random crops, and lastly, normalization by the mean, standard deviation of the dataset.

We did not find a common set of pre-processing techniques as CIFAR10 and CIFAR-100 for Tiny-ImageNet; therefore, we followed a similar data augmentation technique for Tiny-ImageNet.

5.1.2 Evaluation Metrics

In all of the experiments, test accuracy has been considered as the primary metric for evaluation. Besides, we have also defined *capacity ratio*, which refers to the teacher's total number of learnable parameters divided by the student's total number of learnable parameters.

5.1.3 Hyperparameters Setting

For all the experiments, we use stochastic gradient descent (SGD) as the optimizer with Nesterov and momentum of 0.9, the initial learning rate of 0.1 multiplied to 0.2 at epochs $60, 120$, and 180, and weight decay $5e-4$. The models are trained for 200 epochs using batches of size 128.

We did extensive hyperparameter tuning for canonical KD and EKD. Our optimal hyperparameters are similar to [24, 39, 46, 70] where $\tau \in [2.5, 5]$ and $\alpha = 0.1$. In our experiments, regardless of the used distillation framework and dataset $\alpha = 0.1$ and $\tau = 5$, except for KD on CIFAR-100 where $\tau = 4$.

5.2 Comparison Benchmarks

We have used various state-of-the-art KD frameworks to evaluate the effectiveness of our proposed approach. In addition to regular CE and canonical KD [24], the following approaches have been used for comparison:

- FitNets [46] FitNets is a two-stage distillation framework in which the sub-teacher model (up to intermediate *hint* layer) optimizes the student up to intermediate *guided* layer. In the second stage, The whole teacher trains the whole student using canonical KD and soft labels.

- TAKD [39] Knowledge distillation via teacher assistant models bridges the large capacity gap between a powerful pre-trained teacher and small student networks using a chain of teacher assistant models.

- AT [70] AT transfers the teacher's intermediate attention map (channel-wise average of activation map) to the student's equivalent layer.

- TOFD [72] Like our approach, TOFD benefits from intermediate classifier modules to improve the student model. However, in contrast to EKD, TOFD uses very deep and complex classifier modules. Also, both teacher and the student are symmetrically equipped with intermediate heads. TOFD uses four types of loss objective: regular CE, canonical KD using soft probabilities generated by the teacher's same-stage classifier, L_2 loss objective to match same-stage intermediate representations, and the orthogonal loss for information loss reduction(only applied to feature resizing layers).

- MHKD [59] MHKD is another distillation approach that uses intermediate classifiers. However, in MHKD, similar to TOFD [72], both teacher and the student are equipped with multiple classifier heads. MHKD uses a fixed architecture as the classifier module, containing two convolutional layers with batch normalization and ReLU, followed by two fully connected layers. In contrast, EKD benefits from a simpler classifier module by using fully connected layers. MHKD optimizes the student's classifier heads using regular CE and canonical KD with same-stage teacher classifier's soft labels.

- CRD [58] Which improves canonical KD using *contrastive learning*. The loss objective maximizes the teacher-student mutual information's lower bound. Using this framework, the student learns to generate feature maps close to each other for positive sample pairs and increases the distance between the representations for negative pairs.

	CIFAR-10				CIFAR-100				Tiny-ImageNet			
Teacher	CE	KD	EKD	Imp.	CE	KD	EKD	Imp.	CE	KD	EKD	Imp.
WR28-2	88.19	*88.82*	**89.89**	1.07	60.47	60.78	**63.32**	2.54	40.45	*40.70*	**43.89**	3.19
Res110	88.19	*89.30*	**89.44**	0.14	60.47	*62.31*	**63.36**	1.05	40.45	*40.47*	**42.25**	1.78
VGG11	88.19	*88.41*	**89.91**	1.50	60.47	*61.10*	**63.79**	2.69	40.45	*40.76*	**43.78**	3.02
Res34	88.19	*89.26*	**90.00**	0.74	60.47	*61.68*	**63.06**	1.38	*40.45*	40.01	**43.00**	2.55

Table 5.2: Test accuracy (%) of ResNet-8 student network on various teachers and datasets. The student is trained by EKD (ours), canonical KD, or regular cross-entropy (CE). Imp. stands for improvement between the best (in bold) and the second-best (in italics). Average over three runs. For datasets with higher number of classes, the improvement of EKD is higher.

5.3 Performance Comparison

Table 5.2 reports the ResNet-8 student's test accuracy that has been trained using CE, KD, and proposed EKD on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets. We use four different teachers with various model complexity for KD and EKD.

As it is observed, EKD strongly outperforms regular CE and canonical KD regardless of the used dataset. The diverse set of teachers illustrates the generalizability of EKD. As we mentioned earlier, one of the main motivations of EKD is dealing with large capacity gaps between teacher and student, which could underperform KD's performance.

We clearly observe that the capacity gap has underperformed KD compared to CE for Res34-Res8 teacher-student pair using Tiny-ImageNet dataset. In contrast, EKD surpasses both canonical KD and regular CE regardless of the capacity gap. More interestingly, we can detect a relationship between the task's difficulty and the magnitude of improvement, i.e., the more the dataset is challenging (i.e., higher input resolutions, more training samples, or classes), the more significant the improvement would be. On average, EKD improves the student models by 0.86% for CIFAR-10, while this improvement is 1.91% and 2.63% for CIFAR-100 and Tiny-

Teacher	Res20	Res34	WR28-2	Res110	Res34	VGG11	Res34
Student	Res8	WR28-2	Res8	Res8	Res20	Res8	Res8
Capacity Ratio	3.50	15.86	18.50	21.75	76.14	115.80	266.50
CE	60.47	70.08	60.47	60.47	69.30	60.47	60.47
KD [24]	61.22	69.84	60.78	62.31	69.10	61.10	61.68
FitNets [46]	61.69↑	70.81↑	60.74↓	61.55↓	67.20↓	61.37↑	61.53↓
TAKD [39]	61.52↑	68.47↓	61.05↑	61.41↓	69.22↑	60.99↓	61.73↑
TOFD [72]	61.44↑	72.14↑	61.86↑	61.47↓	69.55↑	61.26↑	61.76↑
AT [70]	60.85↓	70.97↑	60.34↓	61.05↓	69.39↑	54.91↓	60.36↓
MHKD [59]	60.03↓	70.39↑	60.94↑	61.16↓	71.00↑	60.54↓	60.66↓
CRD [58]	61.62↑	76.49↑	61.60↑	61.04↓	70.48↑	60.66↓	60.79↓
EKD (ours)	63.11↑	71.41↑	63.32↑	63.36↑	70.66↑	63.79↑	63.06↑

Table 5.3: Test accuracy (%) of various student-teacher pairs on CIFAR-100 datasets. The best, second, and third are shown with gold, silver, and bronze backgrounds, respectively. ↑ and ↓ denote better and worse than KD. Capacity Ratio is the ratio of the number of parameters in the teacher model to the number of the student's parameters EKD (ours) is the only method that consistently has outperformed KD for all teacher-student pairs with any capacity ratio. Average over three runs.

ImageNet datasets.

We also compare EKD with some of the state-of-the-art distillation frameworks using diverse teacher-student model pairs: FitNets [46], AT [70], TAKD [39], TOFD [72], MHKD [59], and CRD [58]. Note that two of these benchmarks (TOFD and MHKD) benefit from intermediate classifiers similar to ours, and TAKD's main motivation is addressing the capacity gap issue. We have defined *capacity ratio* as an indicator of the capacity gap between the teacher and student. Capacity ratio is the total number of learnable parameters in the teacher model, divided by the number of student's learnable parameters. The higher the capacity ratio is, the larger the capacity gap would be, e.g., the Res34 teacher model has 266.5 times more learnable parameters than a tiny Res8 student model. Table 5.3 shows the results of this set of experiments.

EKD outperforms all the benchmarks with a considerable margin for all teacher-student pairs except two teacher-student pairs:(Res34, Res20) and (Res34, WR28-2), where EKD is the second and third best training framework, respectively. EKD is the only method that consistently outperforms canonical KD for all teacher-student pairs regardless of capacity ratio magnitude. Besides EKD, TOFD shows the highest consistency in outperforming KD's performance for all teacher-student pairs except for the Res110-Res8 pair. In comparison with TOFD and MHKD [59, 72], two KD frameworks using intermediate classifiers, EKD shows higher performance in most of the experiments. This is interesting since EKD uses a simple loss objective compared to these two frameworks. Besides, EKD only uses cheap fully-connected layers as intermediate heads only on the teacher's side, while TOFD and MHKD establish complex intermediate modules. We note that CRD could outperform KD with considerable margins for relatively small capacity ratios (e.g., < 21.75), but its performance downgrades for large capacity ratios (e.g., > 100). TAKD's performance is not consistent with capacity ratio changes, and even for some teacher-student pairs, TAKD underperforms KD. FitNets has an inconsistent performance by outperforming over KD for only three teacher-student pairs, indicating its fragility. It is worth noting that the results for TAKD and FitNets are reported after an intensive hyper-parameter tuning process to find the best teacher assistant model(s) or the best hint-guided layer match. The Res110-Res8 seems to be the most challenging teacher-student pair for distillation when none of the approaches (except EKD) could outperform the canonical KD.

Earlier in this chapter, We explained that some benchmarks [46, 72] use euclidean or L_2 loss to match the student's intermediate representations with the teacher's. As observed in Table 5.3, L_2 loss objective does not necessarily improve the student. This inconsistent behavior is due to two factors: First of all, when intermediate represen-

tations have very different dimensionality, even by using a well-designed regressor module [46] or specific loss objectives [72], a fraction of intermediate information would be lost during the feature resizing process, leading to non-effective distillation. Besides, even between compatible representations, the element-wise loss might not necessarily be useful for the student. Consider two identical models that have been trained on the same task with different initialization (e.g., different random weights at the beginning). Although these two models generalize similarly and show similar performance, they could generate very different intermediate activation maps. Naively matching the activation maps does not guarantee an effective distillation. AT partially addresses the two mentioned problems by averaging the activation maps along the channel dimension. However, the results in Table 5.3 indicate that this framework is sensitive to the experiment's setting as well.

In conclusion, these experiments prove that EKD can train more accurate student models than its counterparts. It is worth noting that some of these approaches, e.g., TAKD, specifically, have been proposed to address the significant capacity gap problem in canonical KD. Moreover, EKD surpasses two distillation frameworks that use intermediate classifier modules, while it uses much simpler classifiers with cheaper fine-tuning. Also, EKD establishes a very straightforward loss objective compared to the complex loss in TOFD. EKD addresses the dimensionality mismatch problem very neatly. It also benefits from ground truth labels as the benchmark for evaluating the teacher's intermediate knowledge rather than naively forcing the student model to follow the teacher, even when the teacher infers wrongly.

5.4 EKD vs. Capacity Gap

This Section investigates how EKD can address the capacity gap problem in distillation and compares its results with TAKD.

It has been proven [39] that whenever the model complexity of teachers and the students are very different (i.e., there is a capacity gap between two models), canonical KD would hurt the student's accuracy. One common hypothesis [39] is that the teacher's hints are too complicated for the student to learn. Mirzade et al. [39] proposed the concept of knowledge distillation via teacher assistant models. A chain of teacher assistants bridges the capacity gap between the powerful teacher and the small student. Each teacher assistant learns the knowledge of the primary teacher or previously trained teacher assistant by canonical KD and transfers this information to the weaker teacher assistant or the final student. Sequential KD using medium-size teacher assistants converts the teacher's complicated hints to understandable information for the student.

Although this approach could alleviate the capacity gap issue, it creates multiple new hyperparameters, e.g., the number of teacher assistants, the teacher assistants' architecture, and their training setting. Naively setting the mentioned hyperparameters could significantly harm the final student's performance. Besides, This approach could be very time and resource-consuming, especially when the capacity gap between the teacher and the student is massive. We show that EKD could efficiently address the capacity gap problem without numerous hyperparameters.

Also, EKD benefits from the knowledge of the fully-trained teacher model by adding classifiers to the teacher's optimized layers. For a fair comparison, we did not limit the number of teacher assistants to compare EKD with the most powerful version of TAKD. Each teacher assistant has been trained similarly to the teacher

Framework	Teacher Assistants	Test Acc.
KD	NA	61.68
TAKD	$Res-18$	60.82
TAKD	$Res-18 \rightarrow Res110$	60.73
TAKD	$Res18 \rightarrow Res110 \rightarrow Res56$	61.01
TAKD	$Res18 \rightarrow Res110 \rightarrow Res56 \rightarrow Res32$	61.41
TAKD	$Res18 \rightarrow Res110 \rightarrow Res56 \rightarrow Res32 \rightarrow Res20$	61.60
TAKD	$Res18 \rightarrow Res110 \rightarrow Res56 \rightarrow Res32 \rightarrow Res20 \rightarrow Res14$	61.73
EKD	NA	**63.06**

Table 5.4: A ResNet-8 student model trained under the supervision of ResNet-34 teacher on CIFAR-100 dataset. The teacher assistants have been trained sequentially from left to right.

model. We used the Res34-Res8 teacher-student pair with a massive capacity gap and trained the student using canonical KD, EKD, and all possible scenarios for TAKD, i.e., every possible combination of teacher assistants. Table 5.4 reports the results of these experiments. We can see that a poor selection of teacher assistant models could underperform canonical KD. After multiple resource-consuming experiments, we found that only the chain of six teacher assistants outperforms the canonical KD, while EKD surpasses the canonical KD and TAKD with a less computation overhead.

5.5 Hyperparameter Sensitivity

We observed that EKD not only requires fewer hyperparameters it is also very robust against changes in hyperparameter values (see Sections 5.4 and 5.3). This characteristic makes EKD an even better choice for training gigantic models. Huge models contain billions of parameters, and sometimes even validating these models is a costly task. We experimented the need for delicate hyper-parameter tuning for AT [70], FitNets [46], TAKD [39], and MHKD [59]. This Section can be viewed as the following of the previous Section and TAKD experiments (see Section 5.4). However, in this

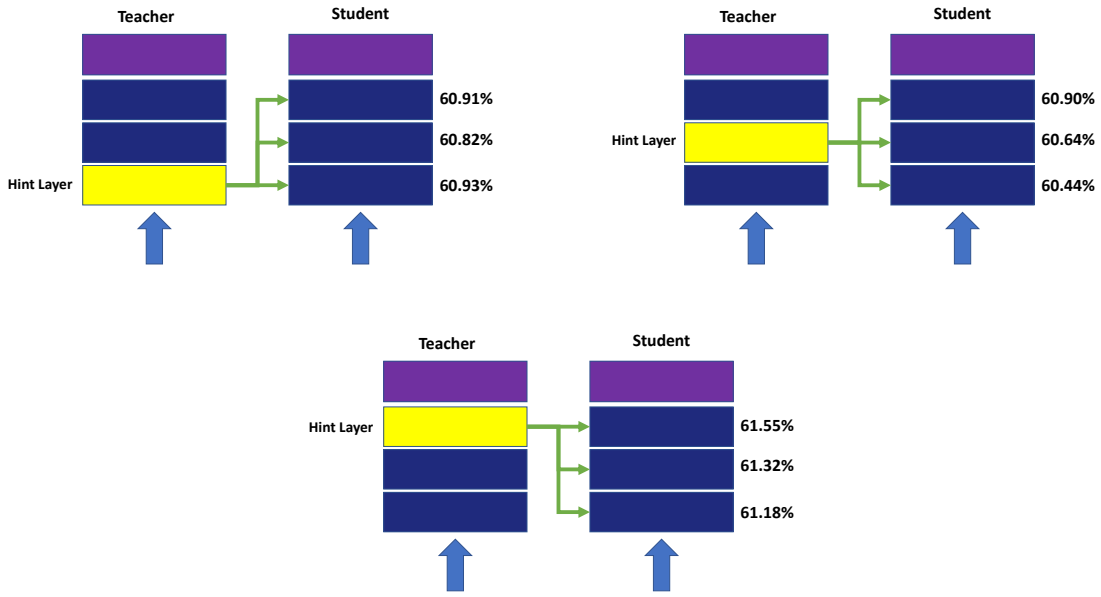


Figure 5.1: A ResNet-8 student model trained under the supervision of the ResNet-110 teacher model using FitNets. Both models generate the same size intermediate representations; therefore, there is no need for a regressor for distillation between equivalent layers. In general, nine different hint-guided layer pairs have been tested. The reported results are the test accuracy after training the student for two stages of FitNets. EKD's test accuracy for the same pair is 63.36%.

Section, we mainly focus on intermediate KD and FitNets. This part compares EKD with FitNets using a ResNet-110 teacher and ResNet-8 student model on CIFAR-100 dataset. The mentioned models comprise three residual blocks with the same dimensionality. Hence the regressor is only required for cross-stage distillation. Figure 5.1 shows nine possible hint-guided layer pairs between the two models. As the figure illustrates, none of these nine possible pairs have led to higher test accuracy than canonical KD. Some of these pairs (those that transfer knowledge between cross-stage layers also require an additional regressor to address dimensionality mismatch). In contrast, EKD surpasses both canonical KD and FitNets with a considerable margin.

The experiments in 5.4 and 5.5 show that EKD generalizes to more models and

scenarios than some of the state-of-the-art KD frameworks. Moreover, it does not require massive hyperparameter tuning.

5.6 Why EKD outperforms Canonical KD?

This Section investigates the intuitions behind EKD, i.e., why EKD works better than canonical KD, while it follows almost the same approach? For this purpose, we analyze EKD in two different aspects: the negative impact of *overthinking* (see Section 5.6.1) on KD and the relationship between EKD and *information entropy* (Section 5.6.2).

5.6.1 EKD and Overthinking

Deep neural models are general approximators [26]; this means a sufficiently powerful deep neural model could solve any mapping function. Since AlexNet [33], there has been a common practice about neural models:"the deeper, the better", i.e., *depth* is a strong indicator of the model's power. Ba et al. [5] show that depth is not necessarily the only factor of power in neural models. A shallow model could reach comparable results similar to its deeper counterparts if it contains an approximately similar number of learnable parameters. However, the community still seeks deeper networks for more precise predictions. Residual architecture [19] eased the creation of very deep neural networks using skip connections. However, although these deep neural models surpass their shallower counterparts, they still suffer from some weaknesses. One of these challenges is *overthinking* [25]. Overthinking occurs when an input sample does not require high dimensional mappings before the final inference, i.e., it can be correctly classified by passing through fewer layers. Passing the input image through extra layers is not only a waste of computation resources but can also lead to misclas-

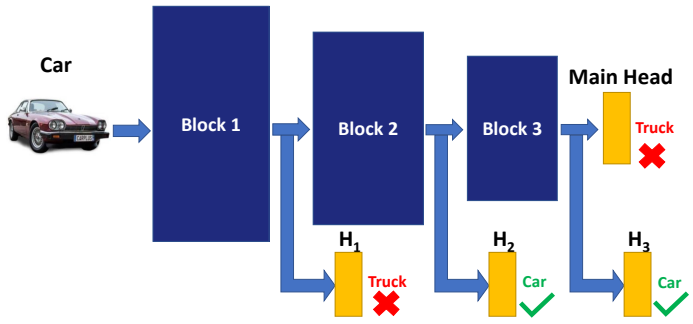


Figure 5.2: Overthinking could lead to misclassification in deep neural networks. The wrong prediction in the most powerful head (i.e., main head) is because of overthinking.

sification in the final classifier. Figure 5.2 depicts this phenomenon in a toy example, with three added intermediate heads. The intermediate heads are added to represent the power of the neural model at various depths. As we observe, the classifiers have classified the input sample differently. While one could expect the wrong prediction in head 1 because of its low model complexity, the misclassification in the main head (i.e., the most powerful classifier), especially after correct classification in two weaker heads 2 and 3, is surprising. This phenomenon motivated us to study overthinking and its impacts on KD. To the best of our knowledge, this is the first work that studies the relationship between overthinking and KD. We used a ResNet-110 teacher with three intermediate classifier heads, fully optimized on CIFAR-10 dataset. Figure 5.3 shows how each head classifies CIFAR-10 training samples. We found that some of the input samples are correctly classified by only the shallowest head (i.e., head 1), while the more powerful heads are failed to do so. According to the figure, head 1 exclusively classifies 80 input images correctly. This is very interesting since the whole ResNet-110 model is almost 21 times more powerful than the sub-model containing head 1. We observe a similar pattern for other heads, i.e., 85, 229, and 188 images are correctly

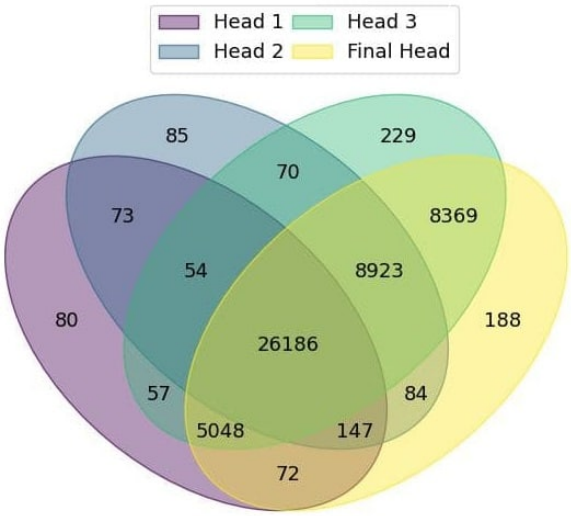


Figure 5.3: The number of correct predictions by each head in the ResNet-110 teacher on CIFAR-10 training dataset.

classified by only one of the heads 2, 3, or the main. We did the same experiment on the testing set as well as other datasets in this work. Regardless of the dataset, overthinking leads to misclassification in deeper layers. This observation emphasizes the value of multiple heads (even the shallowest one) for knowledge distillation, while the deeper heads might be more prone to overthinking. The misclassifications from confident classifier heads penalize the student and reduce its accuracy.

5.6.2 EKD and Information Entropy

We investigate how EKD differs from KD with the ensemble of heads, where the average of heads' probabilities acts as the teacher. We found that KD with ensemble teachers underperforms EKD. This finding is similar to [74], where a cohort of peers could collaboratively train each other better if they individually participate as teachers than members of a unique ensemble. Figure 5.4 shows the outputs of a ResNet-110 on a CIFAR-10 instance image of a plane. We observe that although the ensemble pre-

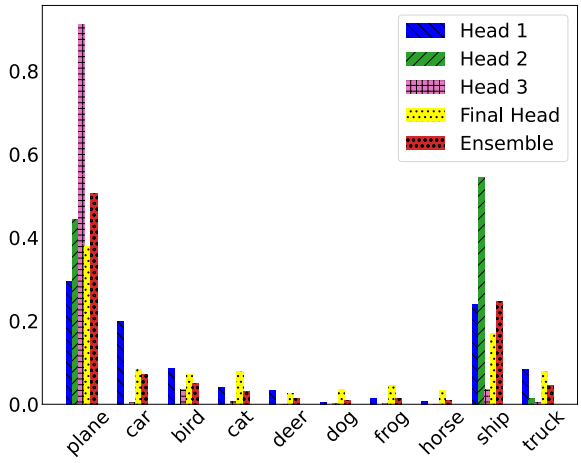


Figure 5.4: Output of Res110's heads on a CIFAR-10 image of plane.

	Head 1	Head 2	Head 3	Main
Entropy	1.785	0.272	0.330	1.837
Head 1	0	5.118	2.139	0.261
Head 2	1.040	0	1.126	0.972
Head 3	1.013	1.398	0	0.599
Main	0.282	5.492	1.422	0

Table 5.5: Average entropy and KL Divergence of heads for Res110 teacher on CIFAR-10 training images, $\tau = 5$.

serves the general inter-class similarity information, the most confident heads highly affect the ensemble, ignoring some minor similarities from weaker heads. We aim to find the impact of each head's confidence and EKD's performance. For this purpose, we defined the confidence of each head by using the definition of information entropy (see Table 5.5).

Information entropy indicates how uncertain the outcomes of a random variable are. As much as we are more certain about the outcomes of a random variable, the entropy would be lower, and vise versa., i.e., when we are sure about en event, i.e., the lower entropy value, the outcome is less informative. We can use this concept to

H_1	H_2	H_3	$Main$	Accuracy	H_1	H_2	H_3	$Main$	Accuracy
○	○	○	○	60.47	○	●	●	○	62.30
●	○	○	○	59.60	○	●	○	●	62.53
○	●	○	○	60.61	○	○	●	●	62.25
○	○	●	○	60.49	●	●	●	○	62.54
○	○	○	●	62.31	●	●	○	●	62.83
●	●	○	○	61.73	●	○	●	●	*62.89*
●	○	●	○	61.80	○	●	●	●	62.79
●	○	○	●	62.30	●	●	●	●	**63.36**

Table 5.6: Ablation study on EKD, CIFAR-100 dataset, Res110 teacher with four heads, Res8 student. Accuracy (%) is the average of three runs. The best and second best are in **bold** and *italic*, respectively. The ● and ○ indicates "on" and "off."

explain why canonical KD improves the student compared to regular CE. In regular CE, the ground truth labels have the entropy of 0, while the soft probabilities with higher entropy provide more information for the student. We were curious to investigate the relationship between the heads' entropy and their impacts on EKD. For this purpose, we calculated the averaged entropy of each head in a ResNet-110 with three added heads on CIFAR-10 dataset (see Table 5.5).

We observe that head 2 and head 3 are very confident in their predictions, where even by increasing the temperature ($\tau = 5$), their entropy is close to zero, while head 1 and main head have almost similar higher entropy values. We did an intensive ablation study to determine which head is the best teacher for distillation or even which combination of heads could improve canonical EKD. Table 5.6 reports the results of this experiment.

According to the definition of information entropy, one could expect that the combination of less confident classifiers (i.e., head 1, 2, and main) is the best teacher union for EKD. In contrast, we observe that the main head is the most influential member among all the heads for EKD. The counter-intuitive results indicate that EKD's power is not only because of confident classifiers but also due to the diverse

set of heads that do the distillation. This finding also explains the inefficiency of ensemble teachers compared to EKD. As the results show, the best teachers' union for the most effective distillation is the total combination (all classifiers) with high and low entropy.

5.7 Summary

In this chapter, we tried to illustrate the superiority of our proposed framework, Enhanced Knowledge Distillation by Auxiliary Classifiers (EKD). We used several teacher-student pairs with different capacity ratios on three well-known datasets and in comparison with multiple state-of-the-art KD approaches. We observed that EKD significantly outperforms its state-of-the-art counterparts. Moreover, we showed that EKD is far cheaper than many compared benchmarks in terms of required computation. Also, EKD shows negligible sensitivity to the used hyperparameter values, making it an affordable approach for training gigantic models. We also investigated why EKD could surpass canonical KD using concepts of overthinking and information entropy.

Chapter 6

Conclusions

In this chapter, we present conclusions and potential future work. Section 6.1 describes the conclusions, and Section 6.2 illustrates the directions for future work.

6.1 Summary

This thesis introduces a knowledge distillation framework called *Enhanced Knowledge Distillation by Auxiliary Classifiers (EKD)*, which exploits and distills the teacher's intermediate knowledge using auxiliary classifier heads. The added auxiliary classifier heads convert the teacher's high-dimensional intermediate representations to probability distributions over the classes in the task. The mapped probability distributions are easy to understand for the student. Besides, since the classifiers have been added at various teacher's layers, the shallower classifiers could bridge the teacher-student large capacity gap, translating the teacher's complicated hints to easier representations for the final student. EKD mitigates the security and privacy concerns by optimizing the small yet accurate models that can be locally deployed in offline devices.

6.2 Future Directions

EKD is a distillation framework that can improve canonical KD using teacher's intermediate representations. These intermediate feature maps are mapped to probability distributions among the classes as a mutual semantic space. EKD has shown remarkable performance in dealing with large capacity gaps between the teacher and the student. This framework has been proposed as a generic framework, which can be used in various applications and domains. Some potential future work after this thesis is presented in the following.

6.2.1 Dynamic Intermediate classifier Architecture Design

In our proposed approach, we used fully connected layers as the simplest architecture for intermediate classifiers. One could expect that using a more delicate architecture for intermediate classifiers can lead to better improvements. However, we surprisingly observed that two of our benchmarks, TOFD, and MHKD [59, 72], with deeper classifiers comprising multiple convolutional, batch normalization, and pooling layers failed to surpass EKD in multiple settings. In our opinion, not paying attention to the backbone model's architecture and inability to address large capacity gaps have weakened the two mentioned frameworks. One interesting topic for future studies could be a dynamic architecture design module using neural architecture search and reinforcement learning, which can consider the base model's structure and the capacity gap between two models.

6.2.2 EKD and Threshold Mechanism

In the current EKD framework, we use all the teacher's classifiers for distillation. We already observed in experiments (see Chapter 5) that we could not decide about

each head's usefulness based on its entropy. However, one could improve canonical EKD by defining a threshold for the confidence of each classifier or mutual prediction between the majority of heads, e.g., if a fraction of classifiers higher than a predefined threshold has the same top k predictions, the student should ignore the rest of the classifiers. We believe that this direction helps to find more intuition about EKD and, more generally, KD, but it can also lead to a more efficient EKD framework.

6.2.3 EKD and Online KD

EKD has been proposed as an offline KD framework, which requires a pre-trained teacher model. However, we are curious to know how we can train multiple peer models using a variant of EKD in the online setting. The capacity gap's negative impacts could still be problematic in the online KD. However, the shallower intermediate classifiers in more powerful peers could act as medium-power peers, reducing the complexity of the powerful peer's hints for the rest of the cohort.

Bibliography

[1] Aguilar, G., Ling, Y., Zhang, Y., Yao, B., Fan, X., and Guo, C. Knowledge distillation from internal representations. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020* (2020).

[2] Alain, G., and Bengio, Y. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, (ICLR-2017)* (2017), OpenReview.net.

[3] Anil, R., Pereyra, G., Passos, A., Ormándi, R., Dahl, G. E., and Hinton, G. E. Large scale distributed neural network training through online distillation. In *6th International Conference on Learning Representations, (ICLR-2018)* (2018).

[4] Asadian, A., and Salehi-Abari, A. Distilling knowledge via intermediate classifier heads. *arXiv preprint arXiv:2103.00497* (2021).

[5] Ba, J., and Caruana, R. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems* (2014).

[6] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (2009).

[7] Blalock, D. W., Ortiz, J. J. G., Frankle, J., and Guttag, J. V. What is the state of neural network pruning? In *Proceedings of Machine Learning and Systems 2020 (MLSys 2020)* (2020), mlsys.org.

[8] Bochkovskiy, A., Wang, C., and Liao, H. M. Yolov4: Optimal speed and accuracy of object detection.

[9] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020* (2020).

[10] Buciluǎ, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006).

[11] Cho, J. H., and Hariharan, B. On the efficacy of knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV-2019)* (2019).

[12] Courbariaux, M., Bengio, Y., and David, J. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015* (2015).

[13] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (2009), Ieee.

[14] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019* (2019).

[15] Duggal, R., Freitas, S., Dhamnani, S., Chau, D. H., and Sun, J. ELF: an early-exiting framework for long-tailed classification. *CoRR* (2020).

[16] Fu, S., Li, Z., Liu, Z., and Yang, X. Interactive knowledge distillation for image classification. *Neurocomputing* (2021).

[17] Gao, M., Shen, Y., Li, Q., and Loy, C. C. Residual knowledge distillation. *CoRR* (2020).

[18] Gao, M., Wang, Y., and Wan, L. Residual error based knowledge distillation. *Neurocomputing* (2021).

[19] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, June 27-30, 2016* (2016).

[20] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2016)* (2016).

[21] He, Y., Zhang, X., and Sun, J. Channel pruning for accelerating very deep neural networks. In *IEEE International Conference on Computer Vision, (ICCV-2017)* (2017), IEEE Computer Society.

[22] Heo, B., Lee, M., Yun, S., and Choi, J. Y. Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI-2019)* (2019).

[23] Heo, B., Lee, M., Yun, S., and Choi, J. Y. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *The Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-2019)* (2019), AAAI Press.

[24] Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS-2015)* (2015).

[25] Hong, S., and Dumitras, T. Shallow-deep networks: Understanding and mitigating network overthinking. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019* (2019), Proceedings of Machine Learning Research, PMLR.

[26] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks* (1989).

[27] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR* (2017).

[28] Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, June 18-22, 2018* (2018), IEEE Computer Society.

[29] Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., and Weinberger, K. Q. Multi-scale dense networks for resource efficient image classification. In *6th International Conference on Learning Representations, ICLR 2018, April 30 - May 3, 2018, Conference Track Proceedings* (2018).

[30] Jin, X., Peng, B., Wu, Y., Liu, Y., Liu, J., Liang, D., Yan, J., and Hu, X. Knowledge distillation via route constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV-2019)* (2019).

[31] Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *16th European Conference of Computer Vision - (ECCV-2020)* (2020).

[32] Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images.

[33] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012* (2012).

[34] Lee, C., Xie, S., Gallagher, P. W., Zhang, Z., and Tu, Z. Deeply-supervised nets. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, May 9-12, 2015* (2015).

[35] Leng, C., Dou, Z., Li, H., Zhu, S., and Jin, R. Extremely low bit neural network: Squeeze the last bit out with ADMM. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), February 2-7, 2018* (2018).

[36] Li, F., and Liu, B. Ternary weight networks. *CoRR* (2016).

[37] Lin, M., Chen, Q., and Yan, S. Network in network. In *2nd International Conference on Learning Representations, ICLR 2014, April 14-16, 2014, Conference Track Proceedings* (2014).

[38] Luan, Y., Zhao, H., Yang, Z., and Dai, Y. MSD: multi-self-distillation learning via multi-classifiers within deep neural networks. *CoRR* (2019).

[39] Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., and Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In *Proceedings of the 34th AAAI Conference on Airtificial Intelligence (AAAI-2020)* (2020).

[40] Müller, R., Kornblith, S., and Hinton, G. E. Subclass distillation. *CoRR* (2020).

[41] Panda, P., Sengupta, A., and Roy, K. Conditional deep learning for energy-efficient and enhanced pattern recognition. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (2016).

[42] Passalis, N., Tzelepi, M., and Tefas, A. Heterogeneous knowledge distillation using information flow modeling. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR-2020)* (2020).

[43] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. 2019.

[44] Phuong, M., and Lampert, C. Distillation-based training for multi-exit architectures. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, October 27 - November 2, 2019* (2019).

[45] Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Computer Vision - ECCV 2016 - 14th European Conference, , October 11-14, 2016, Proceedings, Part IV* (2016), Springer.

[46] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR-2015)* (2015).

[47] Ruiz, A., and Verbeek, J. Anytime inference with distilled hierarchical neural ensembles.

[48] Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, June 18-22, 2018* (2018).

[49] Sau, B. B., and Balasubramanian, V. N. Deep model compression: Distilling knowledge from noisy teachers. *CoRR* (2016).

[50] Shen, C., Wang, X., Yin, Y., Song, J., Luo, S., and Song, M. Progressive network grafting for few-shot knowledge distillation. *CoRR* (2020).

[51] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations, (ICLR-2015)* (2015).

[52] Sun, L., Gou, J., Du, L., and Tao, D. Collaborative teacher-student learning via multiple knowledge transfer.

[53] Sun, S., Cheng, Y., Gan, Z., and Liu, J. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, November 3-7, 2019* (2019), Association for Computational Linguistics.

[54] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015* (2015).

[55] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[56] Tan, M., and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML-2019)* (2019).

[57] Teerapittayanon, S., McDanel, B., and Kung, H. T. Branchynet: Fast inference via early exiting from deep neural networks.

[58] Tian, Y., Krishnan, D., and Isola, P. Contrastive representation distillation. In *8th International Conference on Learning Representations, ICLR 2020, April 26-30, 2020* (2020).

[59] Wang, H., Lohit, S., Jones, M., and Fu, Y. Multi-head knowledge distillation for model compression. *CoRR* (2020).

[60] Wang, X., Zhang, R., Sun, Y., and Qi, J. Kdgan: Knowledge distillation with generative adversarial networks. In *31th Advances in Neural Information Processing Systems (NeurIPS-2018)* (2018).

[61] Wang, Y., Xu, C., Xu, C., and Tao, D. Adversarial learning of portable student networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)* (2018).

[62] Wang, Y., Xu, C., Xu, C., and Tao, D. Adversarial learning of portable student networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)* (2018).

[63] Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J., and Hua, X.-s. Quantization networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2019)* (2019).

[64] Yao, A., and Sun, D. Knowledge transfer via dense cross-layer mutual-distillation. In *Computer Vision - ECCV 2020 - 16th European Conference,*

August 23-28, 2020, Proceedings, Part XV (2020), Lecture Notes in Computer Science.

[65] Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, (CVPR-2017)* (2017), IEEE Computer Society.

[66] Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, (CVPR-2017)* (2017).

[67] You, S., Xu, C., Xu, C., and Tao, D. Learning from multiple teacher networks.

[68] Yuan, L., Tay, F. E. H., Li, G., Wang, T., and Feng, J. Revisiting knowledge distillation via label smoothing regularization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, June 13-19, 2020* (2020).

[69] Zagoruyko, S., and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, September 19-22, 2016* (2016).

[70] Zagoruyko, S., and Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Proceedings of the 5th International Conference on Learning Representations, (ICLR-2017)* (2017).

[71] Zhang, K., Zhang, C., Li, S., Zeng, D., and Ge, S. Student network learning via evolutionary knowledge distillation. *arXiv preprint arXiv:2103.13811* (2021).

[72] Zhang, L., Shi, Y., Shi, Z., Ma, K., and Bao, C. Task-oriented feature distillation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020* (2020).

[73] Zhang, X., Zhou, X., Lin, M., and Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CoRR* (2017).

[74] Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2018)* (2018).