

Improving Knowledge Distillation by Training Teachers to maximize Their Conditional Mutual Information

by

Linfeng Ye

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

© Linfeng Ye 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Knowledge distillation (KD) and its variants, as effective model compression methods, have attracted tremendous attention from both academia and industry. These methods usually use the pretrained teacher models’ outputs and the ground truth labels as supervision signals to train the lightweight student model, which can improve student performance in terms of accuracy.

One aspect of KD that has rarely been explored in the literature is how the behavior of the teacher models affects the students’ performance. Specifically, in most existing KD projects, teacher models are usually trained to optimize their own accuracy. However, recent studies have shown that a teacher with higher accuracy does not always lead to a student with higher accuracy [12, 68]. To explain the aforementioned counter-intuitive observations and advance the understanding of the role of teacher models in KD, the following research problem naturally arises:

How can a teacher model be trained to further improve student’s accuracy in scenarios where the teacher is willing to allow its knowledge to be transferred to the student in whatever form?

In this thesis, we assert that the role of the teacher model is to provide contextual information to the student model during the KD process. In order to increase the contextual information captured by the teacher model, this thesis proposes a novel regularization term called Maximum Conditional Mutual Information (MCMI). Specifically, when a teacher model is trained by conventional cross-entropy loss plus MCMI, its log-likelihood and conditional mutual information (CMI) are simultaneously maximized. The new Class Activation Mapping (CAM) algorithm further verified that maximizing the teacher’s CMI value allows it to capture more contextual information in an image cluster. Via conducting a thorough set of experiments, we show that by employing a teacher trained by CE plus MCMI rather than one trained CE in various state-of-the-art KD frameworks, student’s classification accuracy consistently increases, with a gain of up to 3.32%. In addition, we show that such improvements in the student’s accuracy are more drastic in zero-shot and few-shot settings. Notably, when 5% of the training samples are available to the student (few-shot), the student’s accuracy increases with the gain of up to 5.72%, and increases from 0% to as high as 84% for an omitted class (zero-shot).

Acknowledgements

I would like to express my deepest gratitude to Prof. En-hui Yang, my supervisor, for his altruistic guidance, endless support, and invaluable mentorship. His exemplary approach to doing high-quality research teaches us to think analytically and critically, which has been instrumental in my academic and professional growth. I could not have reached this point in my career without his support.

I would like to thank my coauthors and best friends (in no particular order) Ahmed Hussein Salamah, Renhao Tan, Shayan Mohajer Hamidi, and Kaixiang Zheng. I deeply appreciate the valuable advice and support they provided for my research, as well as the enjoyable times and fascinating discussions we shared. Additionally, I am grateful to Likhil Babu Pallati for the conversations on various topics and the memorable moments we had together.

I thank my family members for their unwavering support throughout the journey. I am incredibly fortunate to have them as my family.

Dedication

This is dedicated to my family members for their endless love and support.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 Knowledge Distillation	1
1.2 The Role of Teacher Model in KD process	2
1.3 Contributions	2
2 Notation and Preliminaries	3
2.1 Notation	3
2.2 Conditional Mutual Information of a model	4
2.3 How CMI Affects the Output Space of the Model	5

3	KD-Amplifying Deep Learning	7
3.1	Capture Contextual Information by CMI	7
3.2	Methodology	9
3.2.1	Optimization Problem Formulation	9
3.2.2	Algorithm for Solving the Optimization in Eq. 3.3	10
3.2.3	Visualization of contextual captured by teacher models.	11
3.3	Experiments	15
3.3.1	MCMI Teacher in KD Methods	15
3.3.2	Zero-shot Classification in KD	18
3.3.3	Few-Shot classification	19
3.3.4	Comparison between MCMI and early-stopped teacher in KD	20
3.3.5	Why larger teachers harm KD	20
4	Conclusion	22
	References	23
A	Appendix	33
A.1	Accuracy variances	33
A.2	Comparing Teacher Trained by Three Regularization Methods: CMI, Entropy, Label Smoothing	35
A.3	knowledge distillation algorithm setup	36
A.4	Real-world examples for zero-shot KD	37
A.4.1	More Zero-shot KD results	38
A.5	Proof of Theorem 1	45
A.6	Proof of Proposition 1	46

List of Figures

2.1	The Markov chain of a classification task. The label space $[C]$ maps to the input space X , and from the input space to the output space $P[C]$ of a model. We use caricatures to depict label and input spaces, where each of the three instances in the label space is mapped to two instances in input space according to $P_{X Y}(\cdot y_i)$, $\forall i \in \{1, 2, 3\}$. While, the output space is obtained from a pretrained ResNet56 model on CIFAR-100, the output probability vectors corresponding to all validation sample instances from those three classes are projected over the two-dimensional probability simplex.	4
2.2	Visualization of the projected clusters of probability distributions corresponding to three randomly selected labels from CIFAR-100 for two models with (a) high CMI value, (b) low CMI value.	6
3.1	prototype and natural images of (i) top: goldfish; (ii) bottom: bee-eater.	8
3.2	The general structure of image classifiers.	11
3.3	The workflow of the C-CAM algorithm.	12
3.4	C-CAM for MLL teacher (first row), and MCMI teacher (second row).	12
3.5	The teacher's CMI value (red curve, right axis) along with the student's accuracy (blue bars, left axis) Vs. the teacher's temperature in conventional KD for three different teacher-student pairs.	13
3.6	The evolution of CMI and LL values for a teacher trained by MLL during the training.	14
3.7	The student's confusion matrices when it is trained through the MLL teacher (left), and by MCMI teacher (right).	18
3.8	The effect of teacher's size on the (i) student's accuracy, (ii) teacher's CMI, and (iii) teacher's LL for both MLL and MCI teachers. The student model is ResNet8, and the dataset is CIFAR-100.	21

A.1	The effect of various regularization methods and MCMI method on (i) student's accuracy, (ii) teacher's entropy, (iii) teacher's CMI.	35
A.2	Zero-shot KD when one of the classes is entirely removed for the student. .	39
A.3	Zero-shot KD when two classes are entirely removed for the student.	40
A.4	Zero-shot KD when three classes are entirely removed for the student. . . .	41
A.5	Zero-shot KD when four classes are entirely removed for the student. . . .	42
A.6	Zero-shot KD when five classes are entirely removed for the student.	43
A.7	Zero-shot KD when six classes are entirely removed for the student.	44

List of Tables

3.1	Top-1 and Top-5 student’s test accuracy (%) on ImageNet validation set for 4 different KD methods using MLL and MCMI teachers (RN and MN stand for ResNet and MobileNet, respectively).	15
3.2	The test accuracy (%) of student networks on CIFAR-100 (averaged over 5 runs), with teacher-student pairs of the same architectures. The subscript denotes the improvement achieved by replacing MLL teacher with MCMI teacher. We use bold numbers and asterisk (*) to denote the best results and to identify the results reproduced on our local machines, respectively. .	16
3.3	The test accuracy (%) of student networks on CIFAR-100 (averaged over 5 runs), with teacher-student pairs of the different architectures. The subscript denotes the improvement achieved by replacing MLL teacher with MCMI teacher. We use bold numbers and asterisk (*) to denote the best results and to identify the results reproduced on our local machines, respectively.	17
3.4	Top-1 accuracy (%) of the student under few-shot setting (averaged over 5 runs).	19
3.5	Student’s accuracy (%) on ImageNet when it is trained by MLL, ES, and CMI teachers.	20
A.1	The variance in accuracy for the results in Table 3.2, where the teacher and student have the same architectures.	33
A.2	The variance in accuracy for the results in Table 3.3, where the teacher and student have different architectures.	34

A.3	Student accuracy of zero-shot learning in CIFAR-100, where teacher is WRN-40-2, and student is WRN-16-2. Numbers in red indicate the accuracy of omitted classes when training students, those in blue represent that of preserved classes. The reported results are averaged over 5 different random seeds.	38
-----	--	----

Chapter 1

Introduction

1.1 Knowledge Distillation

Following the great success of AlexNet [37]. Deep neural network-based machine learning algorithms have revolutionized various field such as medicine [63], natural language processing [15, 61, 6], computer vision (CV) [23, 17], and mass production [75]. It is believed that the crux of their success lies in the benefits of over-parameterization, which enlarge models' representation space [32]. However, those over-parameterized models, always containing millions or even billions of parameters, are characterized by their vast scale and complexity. This makes it challenging to deploy them on devices with limited resources for real-time applications.

To address the challenge mentioned above, as an efficient model compression method, knowledge distillation (KD) [7] has emerged as an efficient model compression method, bridging the performance gap between models of different sizes. In KD, a lightweight model is generally supervised by a larger teacher model. The main idea is that the student model mimics the teacher model in order to obtain a competitive or even superior performance. After KD has been rediscovered and popularized by [25], numerous researchers have attempted to improve the performance of KD by introducing new knowledge transfer algorithms [62, 4, 56], and to understand why distillation works [60, 52, 85, 3, 50, 14]. However, one aspect of KD that has received relatively limited attention is the role of the teacher model and its training process, which will be discussed in the next section.

1.2 The Role of Teacher Model in KD process

To further improve student performance by customizing the teacher model, few works in the literature aim to discuss the role of the teacher model in the knowledge distillation process. [80] argues that the teacher model is designed to be tolerant to help students learn inter-class relations and prevent overfitting. [12, 74] regularized the teacher by utilizing early stopping and checkpoints to achieve a model with large mutual information. [69] trained the teachers to have more dispersed soft probabilities. Additionally, [16] stated that a model exhibiting local Lipschitz continuity around training inputs can serve as a student-oriented teacher which can be a better Bayes conditional probability distribution estimator. All the works mentioned above with the advancement of knowledge distillation methods have greatly improved the distilled students' performance. In this thesis distinguishes itself from all other works in the literature as it argues that the gain of student accuracy comes from the contextual information present in the teacher's output.

1.3 Contributions

In this section, I summarize the main contributions of this thesis by reviewing the key contributions of each individual work.

1. We argue that the so-called dark knowledge in KD process is indeed the contextual information of the input, which can be quantified by CMI value.
2. Equipped with the concept of CMI, we can now justify a batch of observations in knowledge distillation that were previously difficult to justify.
3. We propose a new loss function dubbed MCMI which maximize the log-likelihood and CMI value simultaneously, which allow the teacher model provide more contextual information to the student.
4. By conducting a through set of experiments, by simply replacing the teacher trained by CE with the teacher trained by MCMI in existing state-of-the-art KD methods, student's accuracy consistently increases across the board. Noticeably, the improvements are more significant in both few-shot and zero-shot setting.

Chapter 2

Notation and Preliminaries

2.1 Notation

The set of real numbers is denoted by $\mathbb{R}$. Vectors are denoted by bold-face letters (e.g., $\mathbf{w}$). The i -th element of vector $\mathbf{w}$ is denoted by $\mathbf{w}[i]$. For a natural number K , let $[K] \triangleq \{1, \dots, K\}$. Assume that there are C class labels with $[C]$ as the set of class labels. Let $\mathcal{P}([C])$ denote the set of all C dimensional probability distributions. For any two probability distribution $P_1, P_2 \in \mathcal{P}([C])$, the cross-entropy and Kullback-Leibler (KL) divergence between P_1 and P_2 are denoted by $H(P_1, P_2)$ and $KL(P_1, P_2)$, respectively. For any $y \in [C]$ and $P \in \mathcal{P}([C])$, write the CE of the one-hot probability distribution corresponding to y and P as $H(y, P)$. An $N \times M$ Markov matrix is denoted by $\mathfrak{M}_c^{[N, M]}$.

$\nabla_{\mathbf{w}} f(\cdot)$ denotes the gradient vector of function $f(\cdot)$ w.r.t. vector $\mathbf{w}$. The directional derivative of $f(\cdot)$ in direction of $\boldsymbol{\gamma}$ is denoted by $D_{\boldsymbol{\gamma}}(\cdot)$

For any pair of random variables (X, Y) , denote its joint probability distribution by $P_{X,Y}(x, y)$ or simply $P(x, y)$ whenever there is no ambiguity, the marginal distribution of Y by $P_Y(y)$, and the conditional distribution of Y given $X = x$ by $P_{Y|X}(\cdot|x)$. The mutual information between two random variables X and Y is denoted by $I(X, Y)$, and the CMI of X and Y given a third random variable Z is $I(X, Y|Z)$.

We regard a classification model as a mapping from raw data $x \in \mathbb{R}^d$ to a probability distribution $q_x \in \mathcal{P}([C])$. Given a student model: $x \in \mathbb{R}^d \rightarrow q_x$, let θ denote its weight vector consisting of all its connection weights; whenever there is no ambiguity, we also write q_x as $q_{x,\theta}$.

2.2 Conditional Mutual Information of a model

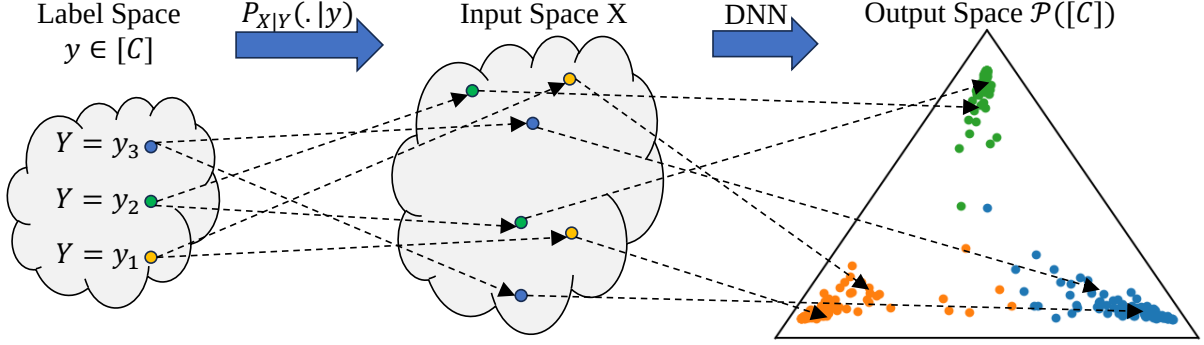


Figure 2.1: The Markov chain of a classification task. The label space $[C]$ maps to the input space X , and from the input space to the output space $\mathcal{P}[C]$ of a model. We use caricatures to depict label and input spaces, where each of the three instances in the label space is mapped to two instances in input space according to $P_{X|Y}(\cdot | y_i)$, $\forall i \in \{1, 2, 3\}$. While, the output space is obtained from a pretrained ResNet56 model on CIFAR-100, the output probability vectors corresponding to all validation sample instances from those three classes are projected over the two-dimensional probability simplex.

In a multi-class classification task, each data point x with same label $y \in [C]$ is sampled from the distribution $P(X|Y = y)$ by observation. In practice, a deep neural network (DNN) maps these samples to the output probability space $\mathcal{P}[C]$ and let $\hat{Y}$ be the label predicted by the DNN with probability $\mathcal{P}[\hat{Y}]$ in respond to the input X . This mapping process can be regarded as a Markov chain as depicted in Fig. 2.1.

As showed in [82], the conditional mutual information (CMI) between X and $\hat{Y}$, given it's ground truth label $Y = y$ can be calculated as:

$$I(X; \hat{Y} | Y = y) = \sum_x P_{X|Y}(x|y) \left[\sum_{i=1}^C P(\hat{Y} = i | x, y) \times \ln \frac{P(\hat{Y} = i | x, y)}{P_{\hat{Y}|Y}(\hat{Y} = i | Y = y)} \right] \quad (2.1)$$

$$= \mathbb{E}_{X|Y} \left[\left(\sum_{i=1}^C q_X[i] \ln \frac{q_X[i]}{P_{\hat{Y}|Y}(\hat{Y} = i | Y = y)} \right) | Y = y \right] = \mathbb{E}_{X|Y} [\text{KL}(q_X, s_y) | Y = y], \quad (2.2)$$

where $s_y = P_{\hat{Y}|Y}(\cdot|y) = \mathbb{E}_{X|Y} [q_X|Y = y]$. $I(X; \hat{Y}|Y = y)$ measures the concentration of the cluster corresponding to label $y \in [C]$. Averaging over all clusters corresponding to all labels y , we get

$$I(X; \hat{Y}|Y) = \sum_{y \in [C]} P_Y(y) I(X; \hat{Y}|Y = y) = \mathbb{E}_{XY} [\text{KL}(q_X, s_Y)]. \quad (2.3)$$

$I(X; \hat{Y}|Y)$ measures the average concentration across all clusters.

For a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ drawn from an unknown distribution $P_{X,Y}$, we can approximate the CMI of f by its empirical value. To this end, let $\mathcal{D}_y = \{1 \leq j \leq N : y_j = y\}$. Denote the size of $\mathcal{D}_y$ by $|\mathcal{D}_y|$. The empirical values of each label specific CMI and the CMI can be calculated as follows

$$I^{emp}(X; \hat{Y}|Y = y) = \frac{1}{|\mathcal{D}_y|} \sum_{i \in \mathcal{D}_y} \text{KL}(q_{x_i}, s_y^{emp}), \quad (2.4)$$

$$I^{emp}(X; \hat{Y}|Y) = \frac{1}{N} \sum_{i=1}^N \text{KL}(q_{x_i}, s_{y_i}^{emp}), \quad (2.5)$$

$$\text{where } s_y^{emp} = \frac{1}{|\mathcal{D}_y|} \sum_{i \in \mathcal{D}_y} q_{x_i}, \forall y \in [C]. \quad (2.6)$$

In next section, we investigated the CMI's effect on the output space of the model by visualizing the clusters of output probability vector.

2.3 How CMI Affects the Output Space of the Model

The outputs of the DNN of all data points belonging to a specific class $Y = y$, where $y \in [C]$, can be considered as a cluster in the DNN's output space. The centroid of this cluster corresponds to the average of q_x with respect to the conditional distribution $P(X | Y = y)$, which is defined as s_y^{emp} in Eq. 2.6.

As Eq. 2.5 indicate, the CMI value is equal to the average KL-divergence between the probability vector and the empirical centroid s_y^{emp} , as though, a smaller CMI($|Y = y$) is the more concentrated cluster around the centroid s_y^{emp} .

To visualize it, two DNN with different CMI values are trained over Cifar-100 dataset. Then we use the visualization method proposed in [82] which project the output probability

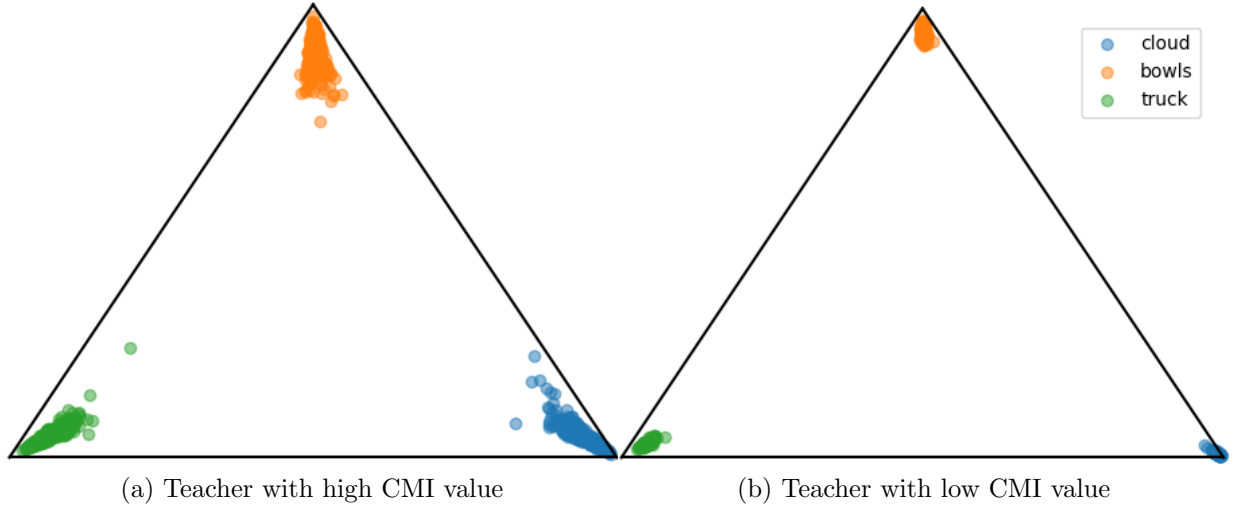


Figure 2.2: Visualization of the projected clusters of probability distributions corresponding to three randomly selected labels from CIFAR-100 for two models with (a) high CMI value, (b) low CMI value.

associated with three randomly selected classes to a two-dimensional simplex. The results are depicted in Fig. 2.2. As seen, the cluster of the model with low CMI value is indeed more concentrated.

Chapter 3

KD-Amplifying Deep Learning

In this chapter, we aim to answer the following question:

How can a teacher model be trained to further improve student’s accuracy in the scenarios where the teacher is in fully cooperative mode and willing to allow its knowledge to be transferred to the student in whatever form?

In [53], researchers first observed that when the teacher is trained by maximizing log-likelihood plus label smoothing (LS), it groups the output probabilities of data points from the same class in tight clusters, which prevents its knowledge from distilling to the student model. Similar findings were reported by [18, 85]; i.e, if the output probability cluster of the teacher model is so compact that almost all output probability vectors with the same ground truth label are collapsed to a single point, in another words the model’s CMI value is low, then the distilled student can not outperforms the one trained using CE plus LS. In fact, when the teacher model eliminates all contextual information from the input, KD essentially reduces to LS. However, in the conventional KD framework, the teacher model is trained by maximize log-likelihood, which solely aims at the prototype information and ignore the contextual information of each cluster. To enable the teacher model to capture both contextual information, in next section we discuss the relation between CMI and contextual information.

3.1 Capture Contextual Information by CMI

Let’s give an in-depth examination of the sampling process, as discussed in Sec. 2.2 each data point with the class y is sampled from the distribution $P(X|Y = y)$. Intuitively, in

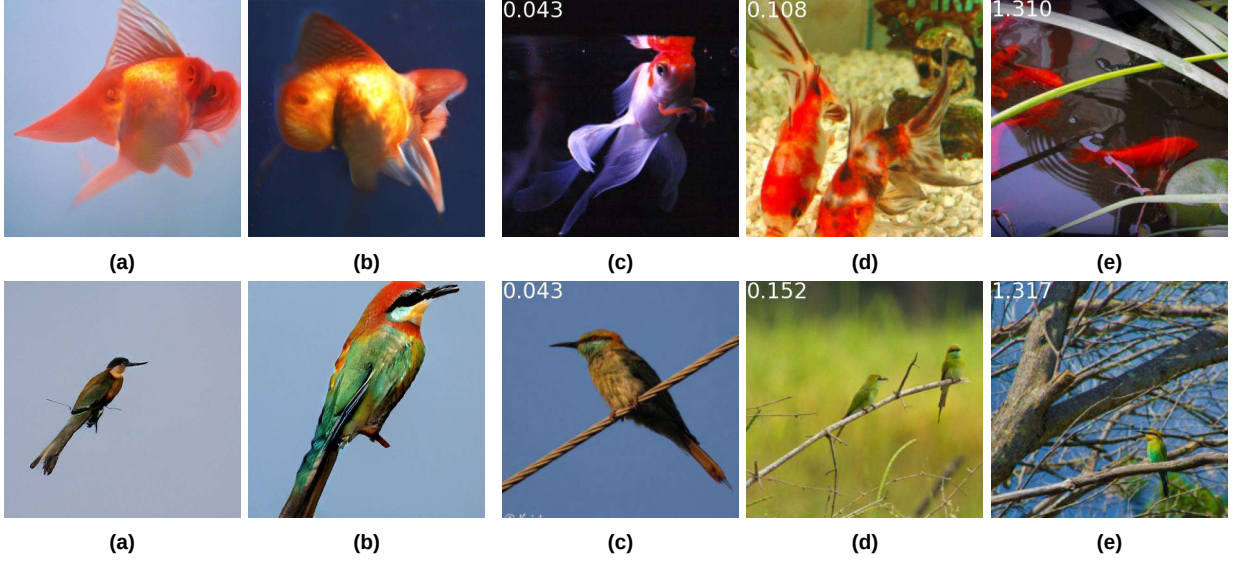


Figure 3.1: prototype and natural images of (i) top: goldfish; (ii) bottom: bee-eater.

a multi-class classification task, each of the class has a distinct *prototype*. As such, all the data points within the class can be regarded as different manifestations of a common *prototype*, each with its own context. For example, the prototype for the last three columns images is goldfish (top), and bee-eater (bottom) (**prototype information**), but they have different background and each goldfish (bee-eater) has its own posture (**contextual information**), which is dominated by the unknown distribution $P(X|Y = y)$. As such, each data point (manifestation) contains two types of information: (i) its *prototype*; and (ii) some contextual information on top its *prototype*.

Let the $\hat{Y}$ be the label predicted by the model with probability $P_X[\hat{Y}]$ corresponding to the input X . That is for any input sample $x \in R^d$ and any $\hat{y} \in [C]$; $Pr(\hat{Y} = \hat{y} | X = x) = P_x[\hat{y}]$. The intricate non-linear function between input and prediction of a model f can be captured by the $CMI(f) = I(X; \hat{Y} | Y)$, which is the quantity of information shared between X and $\hat{Y}$ given Y .

On another hand, we can rewrite the CMI as $I(X; \hat{Y} | Y) = H(X | Y) - H(X | \hat{Y}, Y)$, which can be justified as the difference between the uncertainty of X when Y is known and that of X when both Y and $\hat{Y}$ are known. As such, maximizing $I(X; \hat{Y} | Y)$ is equivalent to minimizing $H(X | \hat{Y}, Y)$, as $H(X | Y)$ is a constant which is determined by the dataset, which, in turn, force $\hat{Y}$ to have as much information as possible about X , given its prototype.

To facilitate the understanding of the semantic meaning attached to CMI, we further visualized the prototypes of goldfish and bee-eater in the eye of the DNN. To this end, an inverse mapping algorithm is developed, which maps the empirical centroid s_y^{emp} associated with the target class y as per Eq.2.5 into a synthetic image. We use a ResNet-34 pretrained over ImageNet as the classification model, and using the BigGANs [5] as a estimator of prior of natural images. By slightly modifying the mapping algorithm proposed in [76], two synthetic images are generated which can be mapped to the empirical centroid s_y^{emp} of each of the “goldfish” and “bee-eater”, which are depicted in the columns (a) and (b) of Fig.3.1. As observed, the synthetic images feature plain backgrounds, and the target objects (goldfish and bee-eater) has similar postures since the absence of contextual information.

3.2 Methodology

3.2.1 Optimization Problem Formulation

Based on previous discussion, a output probability of a good teacher should contains both prototype and contextual information. Hence, we maximize the following loss function to train the teacher model

$$\ell_{\text{MCMI}}(f) = -\text{LL}_{\text{emp}}(f) - \lambda \text{CMI}_{\text{emp}}(f), \quad (3.1)$$

where λ is a predetermined positive hyper-parameter and LL represents the log-likelihood. We note that the objective function in Eq.3.1 is not amenable to parallel computation via GPU due to the dependency of CMI_{emp} on the centrooid s_y of each cluster to $Y = y$. To overcome this, we first convert it into a double minimization problem by introducing a dummy distribution $Q_y \in P[C]$ for each $y \in [C]$, as shown in the following theorem, which has been proved in the Appendix A.5.

Theorem 1 For any $\lambda > 0$

$$\min_{\theta} \left\{ E_{X,Y} \mathbf{H}(y, P_x) - \lambda I(X; \hat{Y} | Y) \right\} \quad (3.2)$$

$$= \min_{\theta} \min_{Q(X|\hat{Y}=i, Y=y)} \left\{ E_{X,Y} \mathbf{H}(y, P_x) - \lambda \sum_{X,Y} P(X, Y) \sum_{i=1}^C P_X(i) \ln \frac{Q(X | \hat{Y} = i, Y = y)}{P_{X|Y}(x|y)} \right\} \quad (3.3)$$

3.2.2 Algorithm for Solving the Optimization in Eq. 3.3

Algorithm 1 The proposed alternating algorithm for solving the optimization problem in Eq. 3.3.

Input: The training set $\mathcal{T}$, all mini-batches $\{\mathcal{B}_b\}_{b \in [B]}$, number of epochs T , λ , and β .

1: **Initialization:**

 Initialize θ^0 and $\{Q_c^0\}_{c \in [C]}$.

2: **for** $t = 1$ to T **do**

3: **for** $b = 1$ to B **do**

4: [Updating θ]:

 Fix $\{Q_{c,b-1}^t\}_{c \in [C]}$.

 Update θ_{b-1}^t to θ_b^t by using (stochastic) batch gradient descent over the loss function $l_{\mathcal{B}_b}(\lambda, \theta_{b-1}^t, \{Q_{c,b-1}^t\}_{c \in [C]})$.

5: [Updating $\{Q_c\}_{c \in [C]}$]:

 Fix θ_b^t .

 Construct mini-batches $\{\mathfrak{B}_c\}_{c \in [C]}$ from $\mathcal{T}$.

 Update $Q_{c,b-1}^t$ to $Q_{c,b}^t$, $\forall c \in [C]$, according to Eq. A.8:

$$P_{\hat{Y}|Y} = \frac{\sum_{\mathfrak{B}_c} P_x}{|\mathfrak{B}_c|} \quad (3.4)$$

$$Q_{c,b}^t(x | i, c) = \frac{P_{x_i}}{|\mathfrak{B}_c| P_{\hat{Y}|Y}} \quad (3.5)$$

6: **end for**

7: **end for**

8: **return** Model parameters θ^T .

By reformulating the single maximization problem into a double maximization problem, we are now ready to present an algorithm for solving Eq. 3.3. In fact, the reformulated formulation lends us an alternating algorithm, which optimize θ and $Q(X | \hat{Y} = i, Y = y)$ alternatively to maximize the Eq. 3.3.

Given $Q(X | \hat{Y} = i, Y = y)$, θ can be updated using standard deep learning process through stochastic gradient descent, in which for each epoch, the training set is randomly partitioned in to B mini-batches $\{\mathcal{B}_b\}_{b \in [B]}$ with each batch of size $|\mathcal{B}|$. Given θ , $Q(X | \hat{Y} = i, Y = y)$ can be updated according to Eq. A.8.

$$P_{\hat{Y}|Y}(i | y) \leftarrow \sum_X P_{X|Y}(x | y) \quad (3.6)$$

$$q(X | \hat{Y} = i, Y = y) \leftarrow \frac{P_{X|Y}(x | y) P_{x,\theta}(i)}{P_{\hat{Y}|Y}(i | y)} \quad (3.7)$$

The procedure for solving the Eq. 3.3 is summarized in Algorithm 1.

3.2.3 Visualization of contextual captured by teacher models.

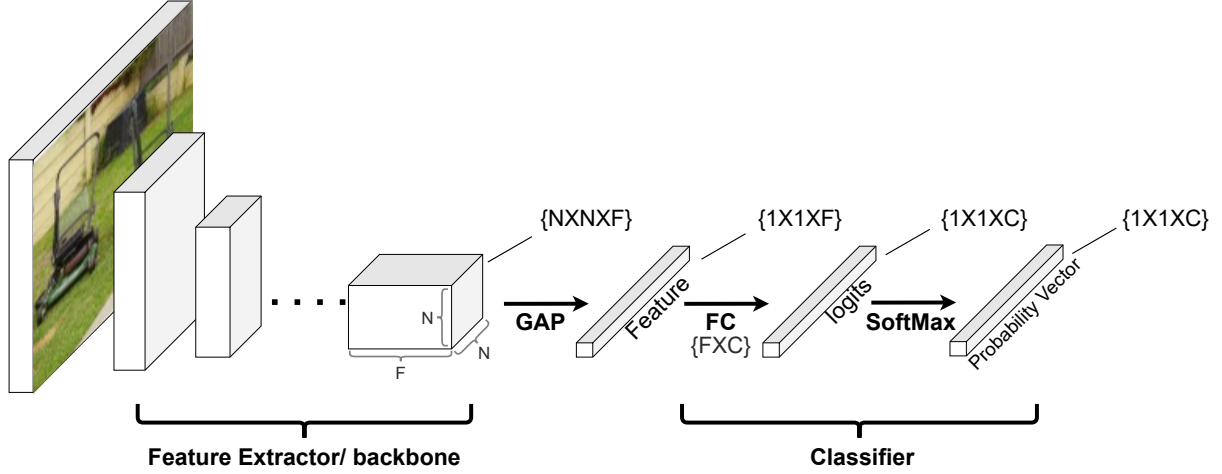


Figure 3.2: The general structure of image classifiers.

In this subsection, we introduce a new CAM algorithm named Contextual-CAM (C-CAM) to visualize and compare the feature maps extracted by MCMI and MLL teachers. Specifically, for a classification model (as depicted in Fig. 3.2), the feature extractor maps input samples to an F dimensional feature map with a spatial resolution of $N \times N$. Next, Global Average Pooling (GAP) layer averages the values of each feature map, converting them into an F dimensional feature vector. Finally, the fully connected layer and softmax function map the feature vector to a C dimensional probability vector.

In C-CAM, the centroid S_c of the target class C is first estimated using Eq. 2.6. Next, the GAP layer is removed, allowing the fully connected (FC) layer to map the feature map of each input sample to $N \times N$ probability vectors. Finally, the KL-divergence between each of these probability vectors and S_c is calculated, resulting in the C-CAM visualization.

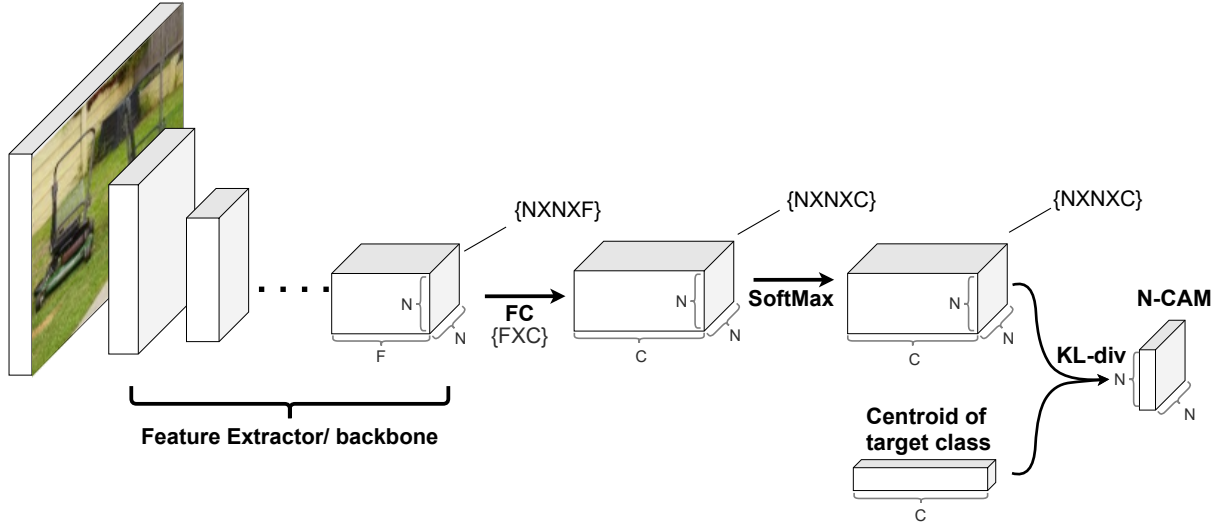


Figure 3.3: The workflow of the C-CAM algorithm.

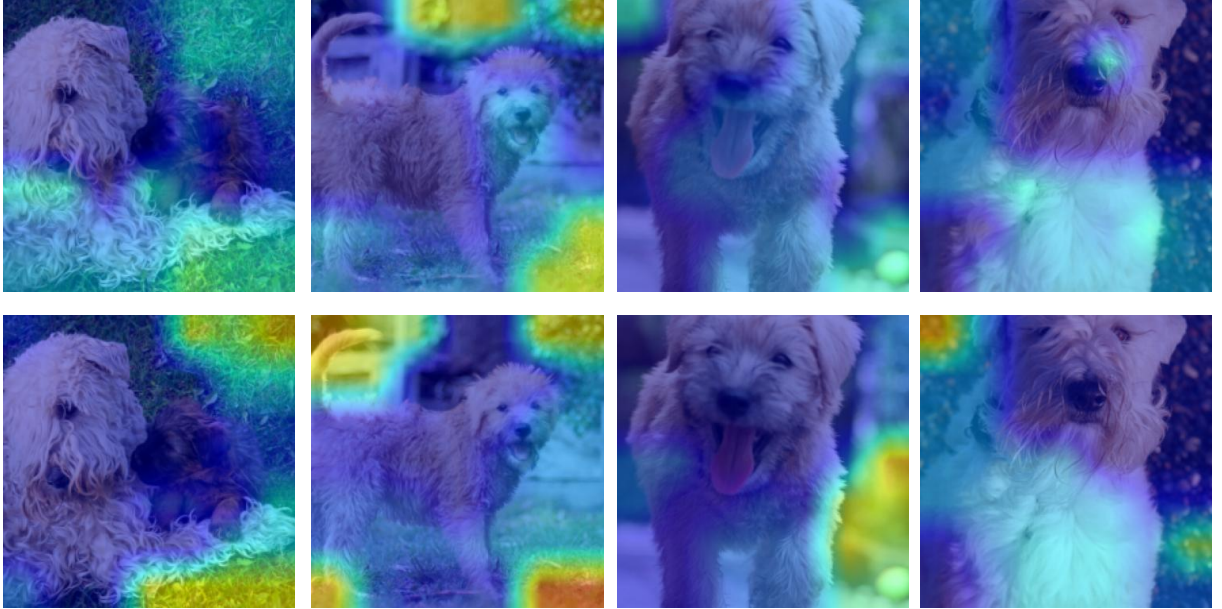


Figure 3.4: C-CAM for MLL teacher (first row), and MCMI teacher (second row).

We randomly selected four images from the "Soft-coated Wheaten Terrier" class in the ImageNet dataset and depict the respective MCMI and MLL teacher's C-CAM for

them (see Fig. 3.4). As shown, the activation maps from the MLL teacher barely highlight any object in the images. In contrast, the activation maps from the MCMI teacher as a whole capture more contextual information, including the dog’s tail, background and other contextual information.

Remark 1: The Role of Temperature in KD

The crux of the [25]’s success lies in applying temperature scaling to the logit before the softmax layer. Although the some works in the literature try to figure out the optimal temperature value by grid search or regard it as a hyper-parameter of the training algorithm [42, 43, 28, 68], the real physical meaning has yet to be elucidated.

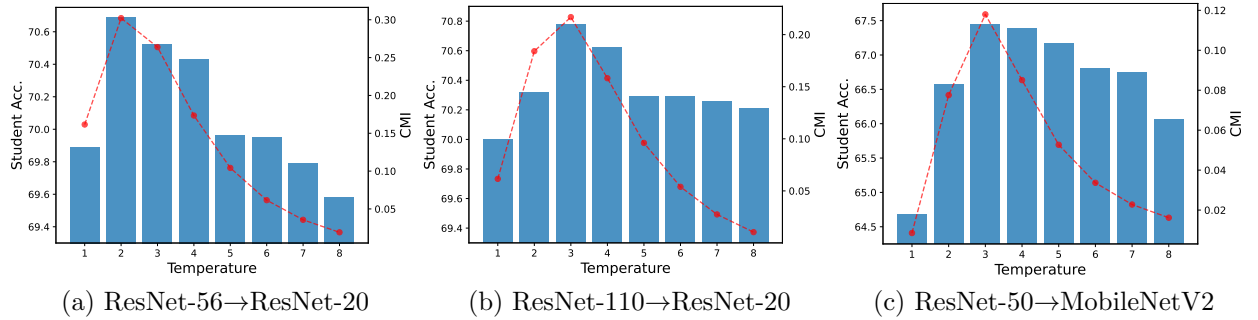


Figure 3.5: The teacher’s CMI value (red curve, right axis) along with the student’s accuracy (blue bars, left axis) Vs. the teacher’s temperature in conventional KD for three different teacher-student pairs.

We argue that the role of temperature is adjusting the teacher’s CMI value by some extent. To illustrate this, following [25] we pick $T = \{1, 2, \dots, 8\}$ and apply the same temperature to both teacher and student for three different teacher-student pairs over Cifar-100 dataset. Fig.3.5, depict the teacher’s CMI value (right vertical axis) with student accuracy (left vertical axis) Vs. the teacher’s temperature (horizontal axis). Let us take Fig.3.5a, for an example, the teacher’s CMI value initially increases reaching its maximum value at $T^* = 2$, and then it monotonically drop, while the student accuracy reaches its maximum value at the same temperature. Similar conclusions can be drawn from other two figures.

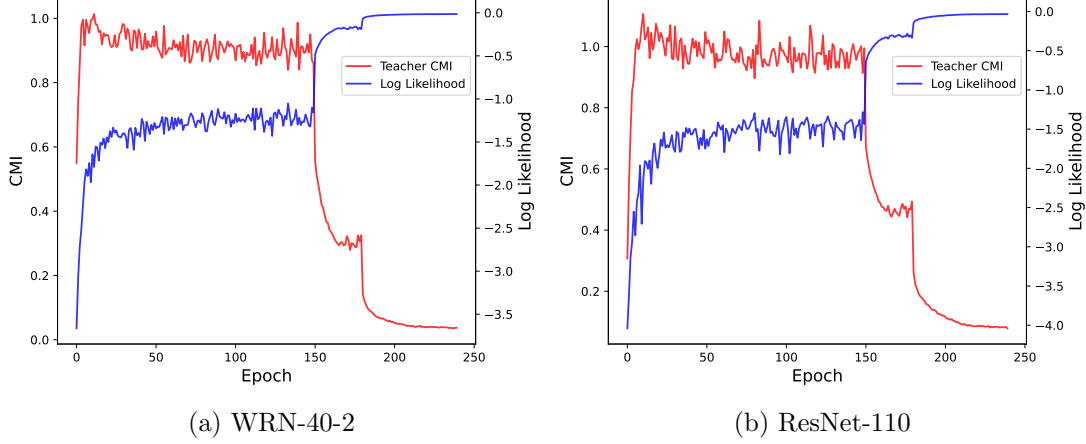


Figure 3.6: The evolution of CMI and LL values for a teacher trained by MLL during the training.

Remark 2: Early Stopping Teacher

One may wonder why an early-stopping model (a teacher model trained by CE while not fully converged. In implementation, the teacher at epoch 100 is picked as early-stopping model on Cifar-100 dataset [36].) is a better teacher comparing with the fully converged one [12]. To answer this question, we plot the curve of CMI and log-likelihood value for two models namely Wide ResNet (WRN)-40-2 [87] and ResNet-110 respectively over Cifar-100 dataset by maximizing log-likelihood in Fig.3.6. As viewed, the CMI value drops dramatically at epoch 150 for both models. Consequently, using an early-stopping model is beneficial since it has larger CMI value compared with the fully converged one. Yet, the problem of early-stopping model is low log-likelihood value, which does not exist in the model trained by MCMI.¹

Proposition 1 *Let f_X be an intermediate feature map of a model corresponding to the input X , then, $I(X; \hat{Y} | Y) \leq I(X; f_X | Y)$.*²

Based on the Proposition 1, maximizing the $I(X; \hat{Y} | Y)$ would also have the effect of increasing $I(X; f_X | Y)$. Therefore, no matter what kind of KD method is used, logit-based or feature-based, teacher models are enabled to provide more contextual information to the student model by maximizing Eq. 3.1.

¹For a comparison between MCMI teacher and early-stopping teacher please refer to Sec. 3.3.4

²See Appendix A.6 for the proof.

3.3 Experiments

We conclude this chapter by conducting extensive experiments to demonstrate the superior effectiveness of MCMI teacher compared to its MLL counterpart ³. Specifically, in Sec 3.3.1 we show that by simply replacing MLL teachers by their MCMI counterparts in the state-of-the-art KD methods, students accuracy consistently increase across the board.

Plug and Play nature of MCMI teacher: For all experiments conduct in this section, to show the effectiveness of our MCMI teacher, we do not tune any hyper-parameter for the underlying KD methods, which use the same hyper-parameter with their benchmarks.

3.3.1 MCMI Teacher in KD Methods

Table 3.1: Top-1 and Top-5 student’s test accuracy (%) on ImageNet validation set for 4 different KD methods using MLL and MCMI teachers (RN and MN stand for ResNet and MobileNet, respectively).

Teacher-Student	Teacher Performance				KD		DKD		ReviewKD		CRD		
		Top1	Top5	CMI	LL	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
RN34-R18	MLL	73.31	91.42	0.7203	-0.5600	71.03	90.05	71.70	90.41	71.61	90.51	71.17	90.13
	MCMI	71.62	90.67	0.8650	-0.6262	71.54	90.63	72.06	91.12	71.98	90.86	71.50	90.20
RN50-MNV2	MLL	76.13	92.86	0.6002	-0.4492	70.50	89.80	72.05	91.05	72.56	91.00	71.37	90.41
	MCMI	74.94	91.03	0.7150	-0.4943	71.10	90.16	72.61	91.26	73.00	92.19	71.63	90.53

In the following, we conduct a set of experiments over Cifar-100 [36] and ImageNet [64] dataset, showing the MCMI teachers’ effectiveness.

• **Cifar-100.** This dataset consists of 60000 color images, each with a resolution of 32×32 pixels. These images are divided into 100 different classes, with each class containing 600 images. The dataset is split into two subsets: 50000 images for training and 10000 images for testing. Following the settings of CRD [71], we conducted our experiments using 12 teacher-student pairs: 6 pairs with identical network architectures and 6 pairs with differing network architectures (see Tab. 3.2 and Tab. 3.3). We perform each experiment across

³For the sake of brevity, hereafter, we refer to the teacher trained by MCMI and maximizing log-likelihood as MCMI and MLL teacher, respectively.

Table 3.2: The test accuracy (%) of student networks on CIFAR-100 (averaged over 5 runs), with teacher-student pairs of the same architectures. The subscript denotes the improvement achieved by replacing MLL teacher with MCMI teacher. We use **bold** numbers and asterisk (*) to denote the best results and to identify the results reproduced on our local machines, respectively.

Teachers and students with the same architectures.												
Teacher	ResNet-56		ResNet-110		ResNet-110		WRN-40-2		WRN-40-2		VGG-13	
	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI
Accuracy	72.34	72.09	74.31	73.60	74.31	73.60	75.61	75.21	75.61	75.21	74.64	73.96
LL	-0.101	-0.344	-0.033	-0.246	-0.033	-0.246	-0.052	-0.132	-0.052	-0.132	-0.007	-0.076
CMI	0.1583	0.4428	0.0605	0.3474	0.0605	0.3474	0.0255	0.1951	0.0255	0.1951	0.0152	0.1298
Student	ResNet-20		ResNet-20		ResNet-32		WRN-16-2		WRN-40-1		VGG-8	
Accuracy	69.06		69.06		71.14		73.26		71.98		70.36	
KD	70.66	70.84 +0.18	70.67	70.85 +0.18	73.08	73.48 +0.40	74.92	75.42 +0.50	73.54	74.53 +0.99	72.98	73.83 +0.85
AT	70.55	70.89 +0.34	70.22	70.68 +0.46	72.31	73.96 +1.65	74.08	74.49 +0.41	72.77	73.25 +0.48	71.43	71.76 +0.33
PKT	70.34	70.96 +0.62	70.25	71.03 +0.78	72.61	72.92 +0.31	74.54	75.01 +0.47	73.45	74.15 +0.70	72.88	73.35 +0.47
SP	69.67	70.98 +1.31	70.04	70.83 +0.79	72.69	73.34 +0.65	73.83	74.60 +0.77	72.43	73.60 +1.17	72.68	73.29 +0.61
CC	69.63	69.98 +0.35	69.48	70.02 +0.54	71.48	71.71 +0.23	73.56	74.00 +0.44	72.21	72.50 +0.29	70.71	71.02 +0.31
RKD	69.61	70.68 +1.07	69.25	70.24 +0.99	71.82	72.65 +0.83	73.35	73.97 +0.62	72.22	72.66 +0.44	71.48	72.03 +0.55
VID	70.38	70.64 +0.26	70.16	70.69 +0.53	72.61	73.10 +0.49	74.11	74.44 +0.33	73.30	73.58 +0.28	71.23	71.93 +0.70
CRD	71.16	71.40 +0.24	71.46	71.93 +0.47	73.48	74.03 +0.55	75.48	75.82 +0.34	74.14	74.86 +0.72	73.94	74.23 +0.29
DKD	71.97	72.31 +0.34	71.51*	71.83 +0.32	74.11	74.36 +0.25	76.24	76.66 +0.42	74.81	75.63 +0.82	74.68	74.87 +0.19
REVIEW	71.89	72.31 +0.42	71.65*	72.11 +0.46	73.89	74.01 +0.12	76.12	76.29 +0.17	75.09	75.47 +0.38	74.84	74.96 +0.12
HSAKD	72.58	72.70 +0.12	72.64*	73.15 +0.51	74.97*	75.71 +0.74	77.20	77.36 +0.16	77.00	77.55 +0.55	75.42*	75.86 +0.44

5 independent runs, and report the average accuracy (for the accuracy variances, refer to Appendix A.1).

For comprehensive comparisons, we evaluate an MCMI teacher versus an MLL teacher across various state-of-the-art distillation methods, including KD [25], AT [86], PKT [58], SP [73], CC [59], RKD [57], VID [2], CRD [71], DKD [89], REVIEWKD [8], and HSAKD [81]. All training setups, including the fine-tuning of MCMI teachers, student training, and λ values, as well as a study on the effect of λ on MCMI teachers, are detailed in A.3.

As shown in Tab. 3.2 and Tab. 3.3, replacing the MLL teacher with the MCMI one

Table 3.3: The test accuracy (%) of student networks on CIFAR-100 (averaged over 5 runs), with teacher-student pairs of the different architectures. The subscript denotes the improvement achieved by replacing MLL teacher with MCMI teacher. We use **bold** numbers and asterisk (*) to denote the best results and to identify the results reproduced on our local machines, respectively.

Teachers and students with different architectures.												
Teacher	ResNet-50		ResNet-50		ResNet-32×4		ResNet-32×4		WRN-40-2		VGG-13	
	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI
Accuracy	79.34	78.45	79.34	78.45	79.41	78.70	79.41	78.70	75.61	75.21	74.64	73.96
LL	-0.004	-0.083	-0.004	-0.083	-0.004	-0.035	-0.004	-0.035	-0.052	-0.132	-0.007	-0.076
CMI	0.0085	0.1065	0.0085	0.1065	0.0059	0.0586	0.0059	0.0586	0.0255	0.1951	0.0152	0.1298
Student	MobileNetV2		VGG-8		ShuffleNetV1		ShuffleNetV2		ShuffleNetV1		MobileNetV2	
	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI
Accuracy	64.60		70.36		70.50		71.82		70.50		64.60	
KD	67.35	70.23 _{+2.88}	73.81	74.59 _{+0.78}	74.07	75.90 _{+1.83}	74.45	76.32 _{+1.87}	74.83	76.45 _{+1.62}	67.37	69.14 _{+1.77}
AT	58.58	60.03 _{+1.45}	71.84	72.19 _{+0.35}	71.73	75.05 _{+3.32}	72.73	75.21 _{+2.48}	73.32	75.61 _{+2.29}	59.40	62.07 _{+2.67}
PKT	66.52	67.42 _{+0.90}	73.10	73.43 _{+0.33}	74.10	75.21 _{+1.11}	74.69	76.34 _{+1.65}	73.89	75.39 _{+1.50}	67.13	68.37 _{+1.24}
SP	68.08	69.07 _{+0.99}	73.34	74.14 _{+0.80}	73.48	76.56 _{+3.08}	74.56	76.70 _{+2.14}	74.52	76.82 _{+2.30}	66.30	67.83 _{+1.53}
CC	65.43	66.76 _{+1.33}	70.25	70.90 _{+0.65}	71.14	71.77 _{+0.63}	71.29	73.02 _{+1.73}	71.38	71.80 _{+0.42}	64.86	65.45 _{+0.59}
RKD	64.43	65.11 _{+0.68}	71.50	72.10 _{+0.60}	72.28	73.59 _{+1.31}	73.21	74.67 _{+1.46}	72.21	74.26 _{+2.05}	64.52	65.37 _{+0.85}
VID	67.57	67.61 _{+0.04}	70.30	70.69 _{+0.39}	73.38	74.58 _{+1.20}	73.40	74.67 _{+1.27}	73.61	75.03 _{+1.42}	65.56	65.77 _{+0.21}
CRD	69.11	69.70 _{+0.59}	74.30	74.86 _{+0.56}	75.11	76.82 _{+1.71}	75.65	77.54 _{+1.89}	76.05	76.62 _{+0.57}	69.70	69.98 _{+0.28}
DKD	70.35	71.70 _{+1.35}	73.94*	75.35 _{+1.41}	76.45	77.21 _{+0.76}	77.07	77.66 _{+0.59}	76.70	77.42 _{+0.72}	69.71	70.35 _{+0.64}
REVIEW	69.89	70.63 _{+0.74}	73.43*	74.20 _{+0.77}	77.45	77.78 _{+0.33}	77.78	78.23 _{+0.45}	77.14	77.56 _{+0.42}	70.37	71.70 _{+1.33}
HSKD	71.83*	72.98 _{+1.15}	75.87*	76.45 _{+0.58}	79.51*	79.77 _{+0.26}	79.93	80.01 _{+0.08}	78.51	78.87 _{+0.36}	71.09*	72.80 _{+1.71}

improves the student’s accuracy across all scenarios, regardless of whether the teacher and student architectures are the same or different. This improvement is consistent across all tested KD methods and can be as high as 3.32%. Notably, the gain is more pronounced when the teacher and student have different architectures. 3.2 and Tab. 3.3 also presents the accuracies of both MLL and MCMI teachers on the test set, and their respective LL and CMI values on the training set. It is evident that the MCMI teacher consistently has lower test accuracy compared to the MLL teacher. This highlights that training a teacher for knowledge distillation purposes is a different task than training it solely for its own classification accuracy.

• **ImageNet.** This dataset is a comprehensive visual database designed for visual object recognition research, containing 1.2 million of labeled training images and 56K labeled validation images across one thousand of categories. In line with the settings from [71, 83], we conducted our experiments using two popular teacher-student pairs, as shown in Tab. 3.1. We observed that, across all knowledge transfer methods reported in Tab. 3.1, replacing the MLL teacher with the MCMI teacher consistently enhances student accuracy. For instance, with the teacher-student pairs ResNet34-ResNet18 and ResNet50-MobileNetV2, the student’s Top-1 accuracy in KD increased by 0.51% and 0.60%, respectively.

3.3.2 Zero-shot Classification in KD

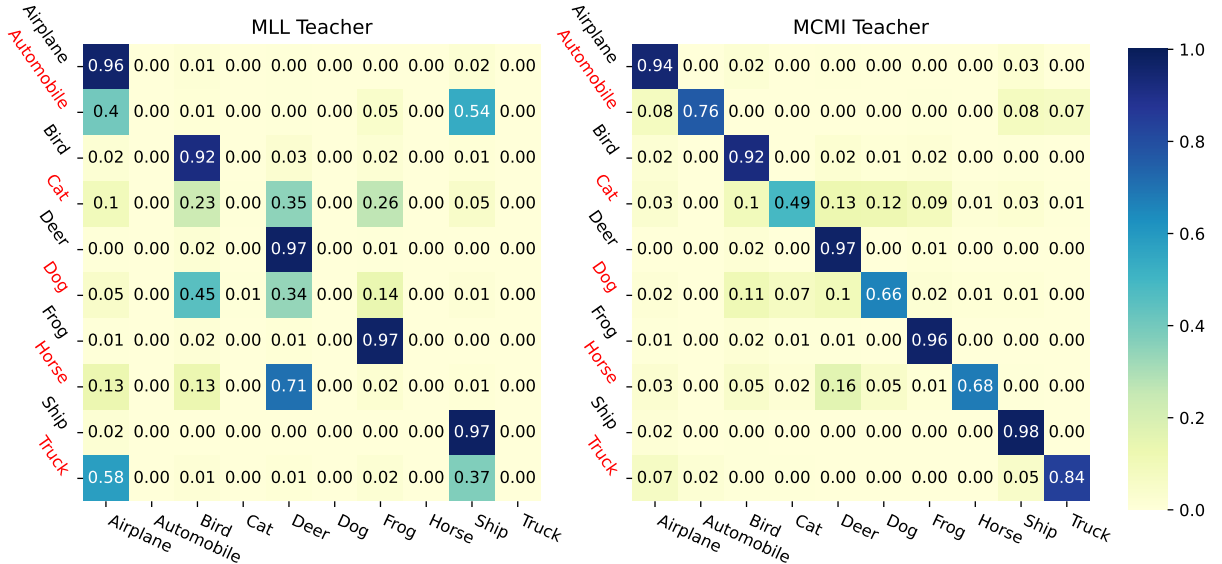


Figure 3.7: The student’s confusion matrices when it is trained through the MLL teacher (left), and by MCMI teacher (right).

In the concept of zero-shot classification in KD, the teacher is trained on the entire dataset, but the samples for some of the classes are completely omitted for the student during the distillation. Then, the student is tested against the samples for the whole dataset (both seen and unseen classes). Some real-world scenarios of this setup is elaborated in Appendix A.4. To show the effectiveness of using MCMI teacher in KD for zero-shot classification, here we report an example of such scenario on CIFAR-10 dataset, and further

results including those for CIFAR-100 are reported in Appendix A.4.1. We use ResNet56-ResNet20 as the teacher-student pair, and train the teacher via MLL and MCMI methods on the whole dataset. Then, we entirely omit five classes from the student’s training set, and train it once using MLL teacher and once using MCMI teacher. The student’s confusion matrices⁴ for both scenarios are depicted in 3.7 where the dropped classes are highlighted in red. As seen, the student’s accuracies for the dropped classes are always zero when using the MLL teacher. On the other hand, these accuracies are substantially increased, by as much as 84%, when using the MCMI teacher. We conducted additional tests in which we dropped varying numbers of classes for the student, as elaborated in Appendix A.4.1.

3.3.3 Few-Shot classification

Table 3.4: Top-1 accuracy (%) of the student under few-shot setting (averaged over 5 runs).

α	5		10		15		25		35		50		75	
Teacher	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI	MLL	MCMI
KD	52.30	58.02 +5.72	60.13	63.75 +3.62	63.52	66.20 +2.68	66.78	68.10 +1.32	68.28	69.34 +1.06	69.52	70.28 +0.76	70.44	70.59 +0.15
CRD	47.60	51.40 +3.80	54.60	56.80 +2.20	58.90	60.02 +1.12	63.82	64.7 +0.88	66.70	67.22 +0.52	68.84	69.15 +0.31	70.35	70.40 +0.05

In few-shot classification, the models are provided with an α percent of the instances in each class [10]. Translating this concept into the context of KD, the teacher is trained on the full dataset, yet an α percent of the samples in each class is used to train the student during the distillation process. Here, we show the effectiveness of MCMI teacher over MLL teacher using the following experiments on CIFAR-100. We use KD and CRD with ResNet-56 and ResNet-20 as the teacher and student models. We conducted experiments for different values of α , namely $\{5, 10, 15, 25, 35, 50, 75\}$. The results are summarized in Tab. 3.4, where for a fair comparison, we employ an identical partition of the training set for each few-shot setting. As seen, the improvement in the student’s accuracy is notable by using an MCMI teacher, which is particularly more pronounced for smaller α values (for instance, when $\alpha = 5$, the improvement in the student’s accuracy is 5.72% and 3.8% when KD and CRD are used, respectively).

⁴Please refer to Appendix A.4.1 for a detailed explanation of how to interpret the confusion matrix.

3.3.4 Comparison between MCMI and early-stopped teacher in KD

Table 3.5: Student’s accuracy (%) on ImageNet when it is trained by MLL, ES, and CMI teachers.

Teacher-Student	R34-R18				R50-MNV2			
	Teacher’s Acc.	CMI	LL	Student’s Acc.	Teacher’s Acc.	CMI	LL	Student’s Acc.
MLL	73.31	0.7203	-0.5600	71.03	76.13	0.6002	-0.4492	70.50
ES	68.68	0.9958	-0.9217	71.25	70.14	0.9598	-0.8840	70.76
MCMI	71.62	0.8650	-0.6262	71.54	74.94	0.7150	-0.4943	71.10

In this subsection, we compare the effectiveness of using MCMI teacher Vs. early-stopped (ES) one in KD. In particular, we conduct experiments on ImageNet dataset, where the teacher-student pairs are ResNet34-ResNet18, and ResNet50-MobileNetV2. The setup for training MCMI teacher is the same as that used for generating the results in Tab. 3.2 and Tab. 3.3. For the ES teacher, following [12], we train ResNet34 for 50 epochs, and ResNet50 for 35 epochs. The results are summarized in Tab. 3.5.

As seen, the student’s accuracy is higher when it is trained by MCMI teacher compared to when it is trained by the other counterpart teachers. In fact, the problem with ES and MLL teachers are that (i) the ES teacher has a low LL value and a relatively high CMI value (to see why an ES teacher has a higher CMI value compared to MLL teacher, please refer to [82]), and (ii) the MLL teacher has a high LL value, yet a low CMI value. Therefore, none-of these methods can establish a fair trade-off between MLL and LL, and consequently they cannot properly estimate the true P_X^* . This issue is alleviated for the MCMI teachers.

3.3.5 Why larger teachers harm KD

It is known that using a large model as the teacher would considerably degrade the student performance [44]. Many papers attribute the cause of this issue to the *capacity gap* between the teachers and the students, and to that the small-scale model struggles to grasp the intricate, high-level semantics extracted by the larger model [12, 51]. Nevertheless, these explanations lack persuasiveness. In fact, we can answer to this phenomenon by observing the LL and CMI values for teachers of different sizes.

In particular, consider Fig. 3.8a, where KD method is used to train ResNet8 using MLL and MCMI teachers of different sizes. As observed, for both MLL and MCMI teachers,

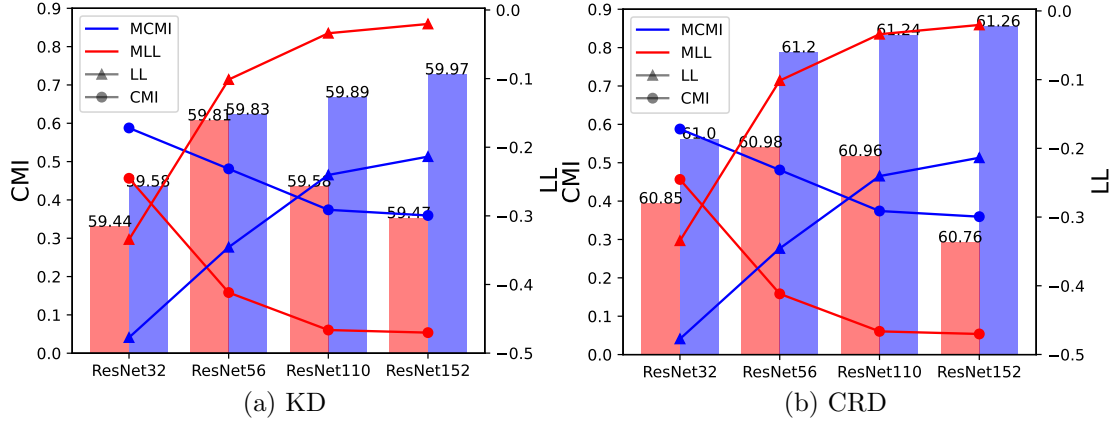


Figure 3.8: The effect of teacher's size on the (i) student's accuracy, (ii) teacher's CMI, and (iii) teacher's LL for both MLL and MCI teachers. The student model is ResNet8, and the dataset is CIFAR-100.

as the model size increases, on the one hand, the teacher's LL value increases (beneficial), and on the other hand, the teacher's CMI decreases (detrimental). However, for MLL teachers with larger size, the CMI values are rather small, although their LL values are quite big. This is the reason that why larger MLL teachers are not often hurt the student's accuracy (as seen the student's accuracy starts decreasing as the MLL teacher model size goes beyond ResNet56).

However, this issue is mitigated when using MCMI teachers, in that the student's accuracy always rises as the teacher's size increases. This is due to the fact that compared to the MLL teacher, the CMI value for the MCMI teacher does not drop significantly as the model size increases. Therefore, an MCMI teacher with a large size can establish a better trade-off between its LL and CMI values yielding a better BCPD estimate.

Chapter 4

Conclusion

In conclusion, this thesis has introduced a novel approach, the maximum conditional mutual information (MCMI) method, to enhance knowledge distillation (KD) by providing more contextual information in student training. Unlike conventional methods that rely on maximum log-likelihood (MLL) estimation, MCMI simultaneously maximizes both log-likelihood and conditional mutual information (CMI) during teacher training. Extensive experiments across various state-of-the-art KD frameworks have demonstrated that utilizing a teacher trained with MCMI leads to a consistent increase in student’s accuracy.

References

- [1] Improved knowledge distillation via teacher assistant. 34:5191–5198, Apr. 2020.
- [2] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9163–9171, 2019.
- [3] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.
- [4] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.
- [5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [7] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

- [8] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2021.
- [9] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2021.
- [10] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2018.
- [11] Anda Cheng and Jian Cheng. Apgp: Accuracy-preserving generative perturbation for defending against model cloning attacks. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.
- [12] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019.
- [13] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.
- [14] Tri Dao, Govinda M Kamath, Vasilis Syrgkanis, and Lester Mackey. Knowledge distillation as semiparametric inference. In *International Conference on Learning Representations*, 2020.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Tamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [16] Chengyu Dong, Liyuan Liu, and Jingbo Shang. Soteacher: Toward student-oriented teacher network training for knowledge distillation, 2023.
- [17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words:

Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

- [18] Yang En-hui and Ye Linfeng. Markov knowledge distillation: Make nasty teachers trained by self-undermining knowledge distillation fully distillable. In *European Conference on Computer Vision*. Springer, 2024.
- [19] Yang En-Hui, Mohajer Hamidi Shayan, Ye Linfeng, Tan Renhao, and Yang Beverly. Conditional mutual information constrained deep learning: Framework and preliminary results. In *IEEE International Symposium on Information Theory*, 2024.
- [20] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, Massachusetts, 1994.
- [21] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. Dual graph enhanced embedding neural network for ctr prediction. KDD '21, page 496–504, New York, NY, USA, 2021. Association for Computing Machinery.
- [22] Shayan Mohajer Hamidi. Training neural networks on remote edge devices for unseen class classification. *IEEE Signal Processing Letters*, pages 1–5, 2024.
- [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [26] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher. *arXiv preprint arXiv:2205.10536*, 2022.
- [27] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [28] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. Annealing knowledge distillation. *arXiv preprint arXiv:2104.07163*, 2021.

- [29] Surgan Jandial, Yash Khasbage, Arghya Pal, Vineeth N. Balasubramanian, and Balaji Krishnamurthy. Distilling the undistillable: Learning from a nasty teacher. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 587–603, Cham, 2022. Springer Nature Switzerland.
- [30] Ying Jin, Jiaqi Wang, and Dahua Lin. Multi-level logit distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24276–24285, 2023.
- [31] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima Jr, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence*, 3(6):473–484, 2021.
- [32] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [33] Reyhan Kevser Keser and Behcet Ugur Toreyin. Averager student: Distillation from undistillable teacher, 2023.
- [34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [35] Donald Knuth. *The T_EXbook*. Addison-Wesley, Reading, Massachusetts, 1986.
- [36] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *University of Toronto*, 2012.
- [37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [38] Souvik Kundu, Qirui Sun, Yao Fu, Massoud Pedram, and Peter Beerel. Analyzing the confidentiality of undistillable teachers in knowledge distillation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 9181–9192. Curran Associates, Inc., 2021.

- [39] Leslie Lamport. *TEX — A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.
- [40] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.
- [41] Taesung Lee, Benjamin Edwards, Ian Molloy, and Dong Su. Defending against neural network model stealing attacks using deceptive perturbations. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 43–49, 2019.
- [42] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1504–1512, 2023.
- [43] Jihao Liu, Boxiao Liu, Hongsheng Li, and Yu Liu. Meta knowledge distillation. *arXiv preprint arXiv:2202.07940*, 2022.
- [44] Li Liu, Qingle Huang, Sihao Lin, Hongwei Xie, Bing Wang, Xiaojun Chang, and Xiaodan Liang. Exploring inter-channel correlation for diversity-preserved knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8271–8280, 2021.
- [45] Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. Undistillable: Making a nasty teacher that {cannot} teach students. In *International Conference on Learning Representations*, 2021.
- [46] Haoyu Ma, Yifan Huang, Tianlong Chen, Hao Tang, Chenyu You, Zhangyang Wang, and Xiaohui Xie. Stingy teacher: Sparse logits suffice to fail knowledge distillation, 2022.
- [47] Haoyu Ma, Yifan Huang, Tianlong Chen, Hao Tang, Chenyu You, Zhangyang Wang, and Xiaohui Xie. Stingy teacher: Sparse logits suffice to fail knowledge distillation, 2022.
- [48] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.
- [49] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

- [50] Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In *International Conference on Machine Learning*, pages 7632–7642. PMLR, 2021.
- [51] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198, 2020.
- [52] Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.
- [53] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.
- [54] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4954–4963, 2019.
- [55] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Prediction poisoning: Towards defenses against dnn model stealing attacks. In *International Conference on Learning Representations*, 2020.
- [56] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.
- [57] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976, 2019.
- [58] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018.
- [59] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5007–5016, 2019.

- [60] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International conference on machine learning*, pages 5142–5151. PMLR, 2019.
- [61] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [62] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.
- [63] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- [64] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.
- [65] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [66] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.
- [67] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- [68] Samuel Don Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work? In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
- [69] Chao Tan and Jie Liu. Improving knowledge distillation with a customized teacher. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

- [70] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532*, 2020.
- [71] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019.
- [72] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.
- [73] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1365–1374, 2019.
- [74] Chaofei Wang, Qisen Yang, Rui Huang, Shiji Song, and Gao Huang. Efficient knowledge distillation from model checkpoints. *Advances in Neural Information Processing Systems*, 35:607–619, 2022.
- [75] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X. Gao, and Dazhong Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48:144–156, 2018. Special Issue on Smart Manufacturing.
- [76] Yulin Wang, Gao Huang, Shiji Song, Xuran Pan, Yitong Xia, and Cheng Wu. Regularizing deep networks with semantic data augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3733–3748, 2022.
- [77] Zi Wang, Chengcheng Li, and Husheng Li. Adversarial training of anti-distilled neural network with semantic regulation of class confidence. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3576–3580, 2022.
- [78] Bingzhe Wu, Shiwan Zhao, Guangyu Sun, Xiaolu Zhang, Zhong Su, Caihong Zeng, and Zhihong Liu. P3sgd: Patient privacy preserving sgd for regularizing deep cnns in pathological image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2099–2108, 2019.
- [79] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.
- [80] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan L Yuille. Training deep neural networks in generations: A more tolerant teacher educates better students. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5628–5635, 2019.

- [81] Chuanguang Yang, Zhulin An, Linhang Cai, and Yongjun Xu. Hierarchical self-supervised augmented knowledge distillation. *arXiv preprint arXiv:2107.13715*, 2021.
- [82] En-Hui Yang, Shayan Mohajer Hamidi, Linfeng Ye, Renhao Tan, and Beverly Yang. Conditional mutual information constrained deep learning for classification. *arXiv preprint arXiv:2309.09123*, 2023.
- [83] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2020.
- [84] Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17185–17194, 2023.
- [85] Linfeng Ye, Shayan Mohajer Hamidi, Renhao Tan, and EN-HUI YANG. Bayes conditional distribution estimation for knowledge distillation based on conditional mutual information. In *The Twelfth International Conference on Learning Representations*, 2024.
- [86] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2016.
- [87] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.
- [88] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia conference on computer and communications security*, pages 159–172, 2018.
- [89] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022.
- [90] Kaixiang Zheng and En-Hui Yang. Knowledge distillation based on transformed teacher matching. In *The Twelfth International Conference on Learning Representations*, 2024.

- [91] Alexander Ziller, Dmitrii Usynin, Rickmer Braren, Marcus Makowski, Daniel Rueckert, and Georgios Kaissis. Medical imaging deep learning with differential privacy. *Scientific Reports*, 11(1):13524, 2021.

Appendix A

Appendix

A.1 Accuracy variances

Table A.1: The variance in accuracy for the results in Table 3.2, where the teacher and student have the **same** architectures.

Teacher	ResNet-56	ResNet-110	ResNet-110	WRN-40-2	WRN-40-2	VGG-13
Student	ResNet-20	ResNet-20	ResNet-32	WRN-16-2	WRN-40-1	VGG-8
KD	70.84 ± 0.10	70.85 ± 0.20	73.48 ± 0.13	75.42 ± 0.12	74.53 ± 0.34	73.83 ± 0.22
AT	70.89 ± 0.28	70.68 ± 0.32	73.96 ± 0.18	74.49 ± 0.15	73.25 ± 0.28	71.76 ± 0.16
PKT	70.96 ± 0.38	71.03 ± 0.12	72.92 ± 0.14	75.01 ± 0.15	74.15 ± 0.28	73.35 ± 0.20
SP	70.98 ± 0.07	70.83 ± 0.26	73.34 ± 0.18	74.60 ± 0.27	73.60 ± 0.16	73.29 ± 0.14
CC	69.98 ± 0.47	70.02 ± 0.18	71.71 ± 0.23	74.00 ± 0.24	72.50 ± 0.16	71.02 ± 0.29
RKD	70.68 ± 0.13	70.24 ± 0.17	72.65 ± 0.24	73.97 ± 0.15	72.66 ± 0.21	72.03 ± 0.28
VID	70.64 ± 0.25	70.69 ± 0.09	73.10 ± 0.13	74.44 ± 0.25	73.58 ± 0.28	71.93 ± 0.10
CRD	71.40 ± 0.13	71.93 ± 0.17	74.03 ± 0.08	75.82 ± 0.15	74.86 ± 0.11	74.23 ± 0.05
DKD	72.31 ± 0.12	71.83 ± 0.18	74.36 ± 0.14	76.66 ± 0.22	75.63 ± 0.09	74.87 ± 0.11
REVIEWKD	72.31 ± 0.26	72.11 ± 0.30	74.01 ± 0.43	76.29 ± 0.16	75.47 ± 0.14	74.96 ± 0.18
HSAKD	72.70 ± 0.28	73.15 ± 0.16	75.71 ± 0.13	77.36 ± 0.17	77.55 ± 0.22	75.86 ± 0.20

Table A.2: The variance in accuracy for the results in Table 3.3, where the teacher and student have **different** architectures.

Teacher Student	ResNet-50 MobileNetV2	ResNet-50 VGG-8	ResNet-32×4 ShuffleNetV1	ResNet-32×4 ShuffleNetV2	WRN-40-2 ShuffleNetV1	VGG-13 MobileNetV2
KD	70.23 ± 0.34	74.59 ± 0.16	75.90 ± 0.11	76.32 ± 0.47	76.45 ± 0.27	69.14 ± 0.22
AT	60.03 ± 0.21	72.19 ± 0.94	75.05 ± 0.17	75.21 ± 0.13	75.61 ± 0.25	62.07 ± 0.26
PKT	67.42 ± 0.18	73.43 ± 0.17	75.21 ± 0.27	76.34 ± 0.18	75.39 ± 0.14	68.37 ± 0.35
SP	69.07 ± 0.17	74.14 ± 0.19	76.56 ± 0.29	76.70 ± 0.18	76.82 ± 0.14	67.83 ± 0.28
CC	66.76 ± 0.42	70.90 ± 0.20	71.77 ± 0.17	73.02 ± 0.17	71.80 ± 0.19	65.45 ± 0.39
RKD	65.11 ± 0.49	72.10 ± 0.27	73.59 ± 0.18	74.67 ± 0.29	74.26 ± 0.31	65.37 ± 0.10
VID	67.61 ± 0.19	70.69 ± 0.17	74.58 ± 0.23	74.67 ± 0.25	75.03 ± 0.17	65.77 ± 0.36
CRD	69.70 ± 0.69	74.86 ± 0.17	76.82 ± 0.22	77.54 ± 0.10	76.62 ± 0.27	69.98 ± 0.13
DKD	71.70 ± 0.35	75.35 ± 0.13	77.21 ± 0.14	77.66 ± 0.19	77.42 ± 0.45	70.35 ± 0.08
REVIEWKD	70.63 ± 0.26	74.20 ± 0.85	77.78 ± 0.31	78.23 ± 0.14	77.56 ± 0.32	71.70 ± 0.46
HSAKD	72.98 ± 0.26	76.45 ± 0.19	79.77 ± 0.09	80.01 ± 0.16	78.87 ± 0.42	72.80 ± 0.11

A.2 Comparing Teacher Trained by Three Regularization Methods: CMI, Entropy, Label Smoothing

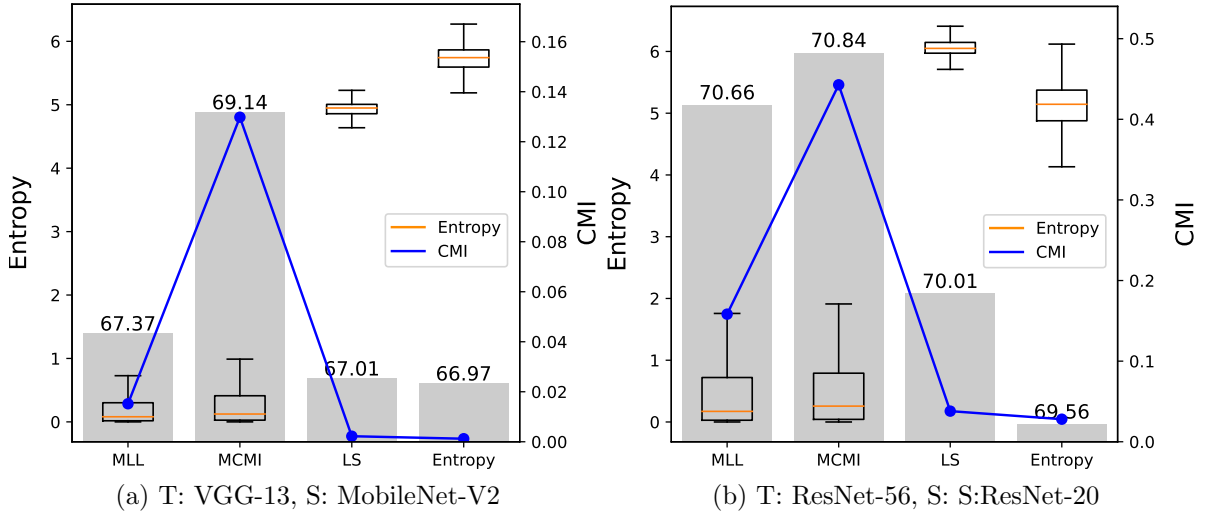


Figure A.1: The effect of various regularization methods and MCMC method on (i) student’s accuracy, (ii) teacher’s entropy, (iii) teacher’s CMI.

In this subsection, in order to unveil the reason why label smoothing hurts student’s accuracy in knowledge distillation [70], we conduct a comparative study to see how training teachers with different regularization terms, namely {CMI, entropy, label smoothing}, affect KD.

To this end, we train teachers using these regularization methods, and report the teacher’s CMI value and the average entropy of its output probability vectors. Then, we use these three teachers to train a student via KD framework, and also report the accuracy of the resulting students. The results for two teacher-student pairs are depicted in Fig. A.1, where we also depicted the results for an MLL teacher (the teacher with no regularization term).

As observed, the CMI regularization does not increase the entropy of the teacher’s output predictions. On the other hand, the teachers trained by entropy and LS regularization have low CMI value, and high output’s entropy. This shows that the CMI value is not related to the entropy of the output probability vectors. In addition, we see that to train

students with high accuracy, the teacher’s CMI value should be high, and the entropy of its output is not important.

A.3 knowledge distillation algorithm setup

For all variants of knowledge distillation and settings in this paper, SGD is applied as the optimizer.

For CIFAR-100 dataset, we train 240 epochs for all experiments with an initial learning rate of 0.05 by default, which will be decayed by factor of 0.1 at epoch 150, 180, 210. For MobileNetV2, ShuffleNetV1, and ShuffleNetV2, a smaller initial learning rate of 0.01 is used. We adopt batch size of 64.

For ImageNet dataset, we used the same training recipe as PyTorch official implementation, except train for 10 more epochs. Batch size of 256 is adopted.

In the paper, we incorporate several state-of-the-art KD variants. Specifically, we employ a linear combination of cross-entropy loss and knowledge distillation loss, as outlined below:

$$\ell = \ell_{\text{CE}} + \gamma \ell_{\text{Distill}} \quad (\text{A.1})$$

For different KD variants, we directly adopted the γ values as those reported in their original papers. The γ values and additional details for each KD variant used in the paper are outlined below:

1. AT [86]: $\gamma = 1000$;
2. PKT [58]: $\gamma = 30000$;
3. SP [73]: $\gamma = 3000$;
4. CC [59]: $\gamma = 0.02$;
5. RKD [57]: $\gamma_1 = 25$ for distance, and $\gamma_2 = 50$ for angle;
6. VID [2]: $\gamma = 1$;
7. CRD [71]: $\gamma = 0.8$;

8. DKD [89]: Consistent with their official implementation, we select γ from the set $\{0.2, 1, 2, 4, 8\}$;
9. REVIEWKD [8]: Consistent with their official implementation, we select γ from the set $\{0.6, 1, 5, 8\}$;
10. HSAKD [81]: $\gamma = 1$.

For KD, we follow the original implementation in [25], set $\alpha = 0.9$, and $T = 4$.

For both zero-shot and few-shot, we follow the same training recipe as that discussed in this subsection.

A.4 Real-world examples for zero-shot KD

Zero-shot learning in the context of knowledge distillation (KD) can be used when you want a student model to learn from a teacher model while omitting certain classes. In this subsection, we will provide detailed explanations of two real-world examples that show the application of zero-shot learning within the framework of KD.

Consider the scenario that you are tasked with developing a disease classification system using medical images as input data. The central repository of these medical images is managed by the government, which possesses the teacher model designed for this purpose. However, local hospitals express interest in utilizing these images for disease diagnosis. Due to privacy concerns, sharing all the images with these hospitals is not feasible [21].

Here’s how zero-shot learning in KD with omitted classes could be applied:

1. Teacher Model: You have a teacher model that has been trained on a large medical images including those sensitive ones.
2. Omitted Classes: Your objective is to create a student model for disease diagnosis. However, you want to exclude certain sensitive images from the training dataset.
3. Knowledge Distillation: You create a student model for disease classification and initiate knowledge distillation from the teacher model.
4. Zero-Shot Learning: The student model, can still perform diagnosing effectively, even for sensitive images that were omitted during training. It achieves this through zero-shot learning, as it learns to predict diseases that have never seen directly in the training data by leveraging the knowledge from the teacher model.

In this scenario, zero-shot learning in KD allows you to build a classification system that respects **privacy** and **confidentiality** by omitting certain classes (sensitive images) while ensuring that the diagnosing quality remains high for other diseases [22].

A.4.1 More Zero-shot KD results

- **CIFAR-100:** We consider the zero-shot in KD over CIFAR-100 where the teacher is trained over the whole dataset while some number of classes are dropped during the distillation. Specifically, we consider dropping $\{5, 10, 20, 30, 40, 50\}$ classes. The teacher and student models are WRN-40-2 and WRN-16-2, respectively. We compare using the MLL and MCMI teacher in the above scenario. Then, we report the accuracy of the student for both dropped and preserved classes, separately.

The results are summarized in Tab. A.3. As the results suggest, the accuracy for the dropped classes is significantly increases by using the MCMI teacher. In addition, the accuracy for the preserved classes is almost the same when using either teacher. This suggests that the accuracy for the preserved classes is not sacrificed for a higher accuracy for the dropped classes.

Table A.3: Student accuracy of zero-shot learning in CIFAR-100, where teacher is WRN-40-2, and student is WRN-16-2. Numbers in red indicate the accuracy of omitted classes when training students, those in blue represent that of preserved classes. The reported results are averaged over 5 different random seeds.

# of dropped classes	5		10		20		30		40		50	
MLL	0.16	/ 75.02	2.70	/ 75.82	1.63	/ 76.28	1.11	/ 80.15	1.74	/ 80.42	0.87	/ 82.38
MCMI	25.64	/ 75.08	32.78	/ 75.78	30.61	/ 76.22	21.72	/ 80.27	23.74	/ 80.79	22.26	/ 82.46

- **CIFAR-10:** For this dataset, we use a 10×10 confusion matrix $\mathbf{C}$ to better observe the model performance. Each row in $\mathbf{C}$ represents the instances in an actual class, while each column represents the instances predicted by the model. In particular, the entry $\mathbf{C}_{i,i}$, $i \in [10]$, represents the probability that the instances in class i are correctly classified. On the other hand, $\mathbf{C}_{i,j}$, $i, j \in [10], i \neq j$, represents the probability of the cases where instances from class i are incorrectly classified as belonging to Class j .

Then, we consider dropping $\{1, 2, 3, 4, 5, 6\}$ classes from the student’s training set. The dropped classes are denoted by red color in the confusion matrices.

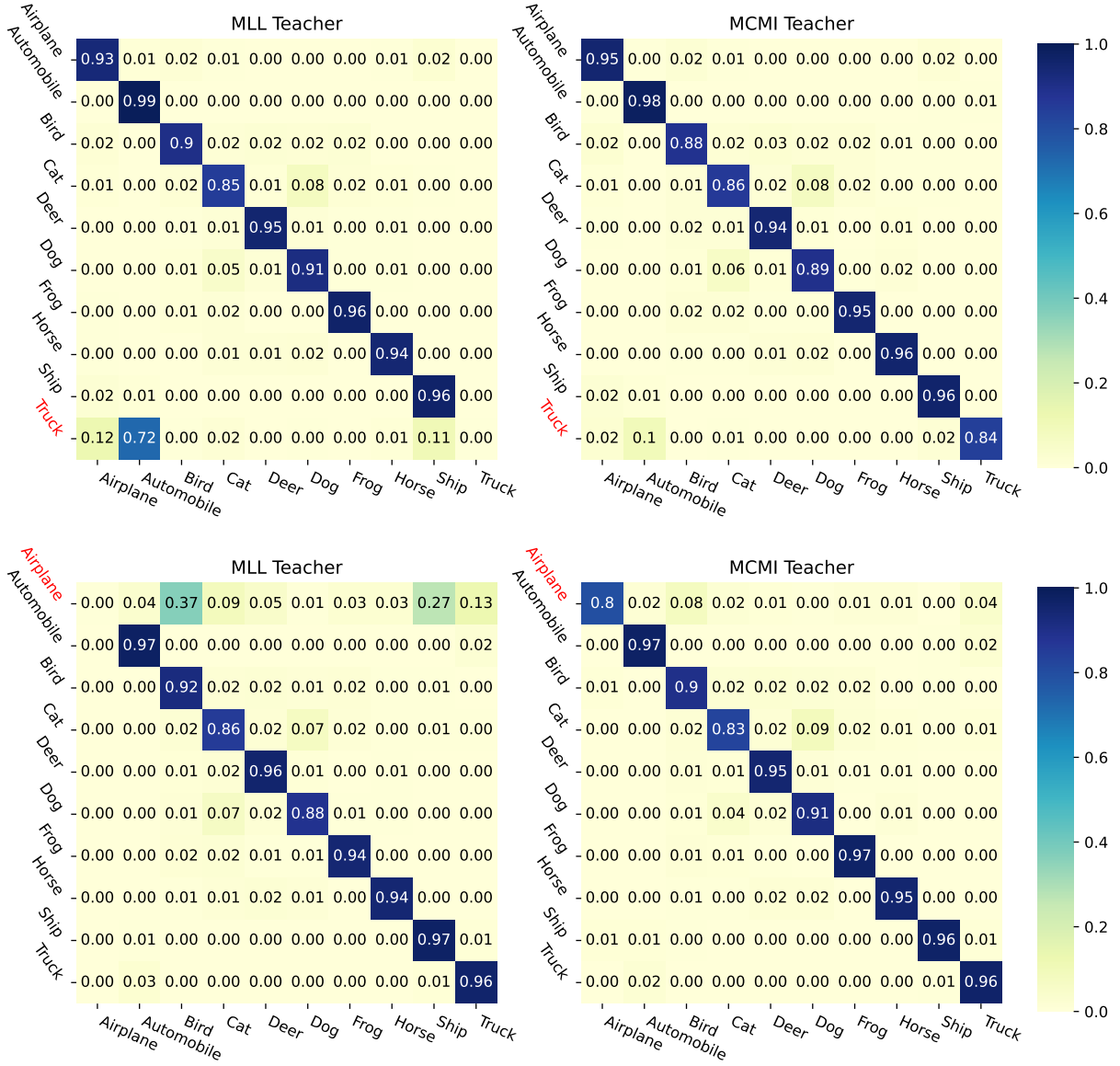


Figure A.2: Zero-shot KD when one of the classes is entirely removed for the student.

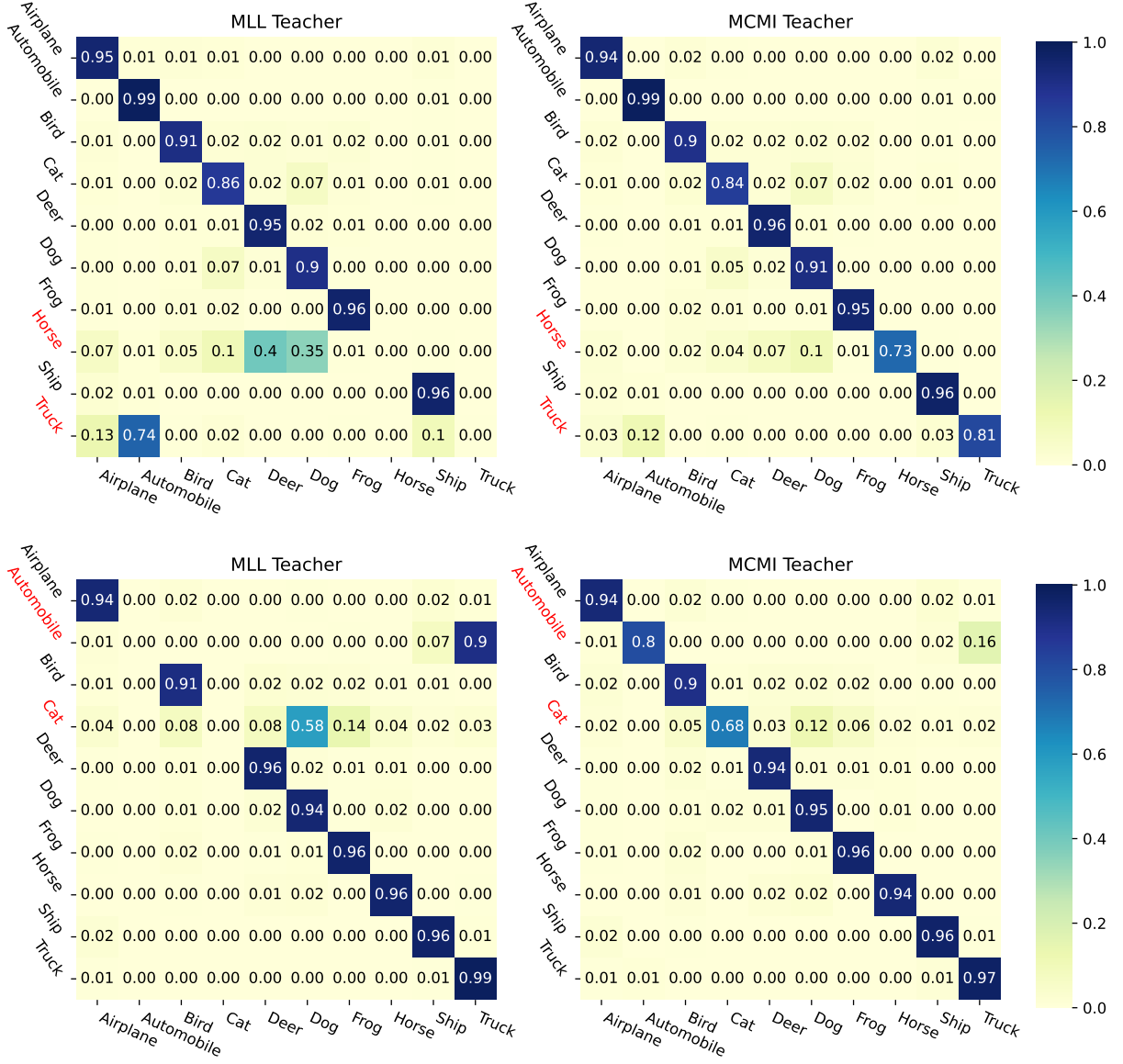


Figure A.3: Zero-shot KD when two classes are entirely removed for the student.

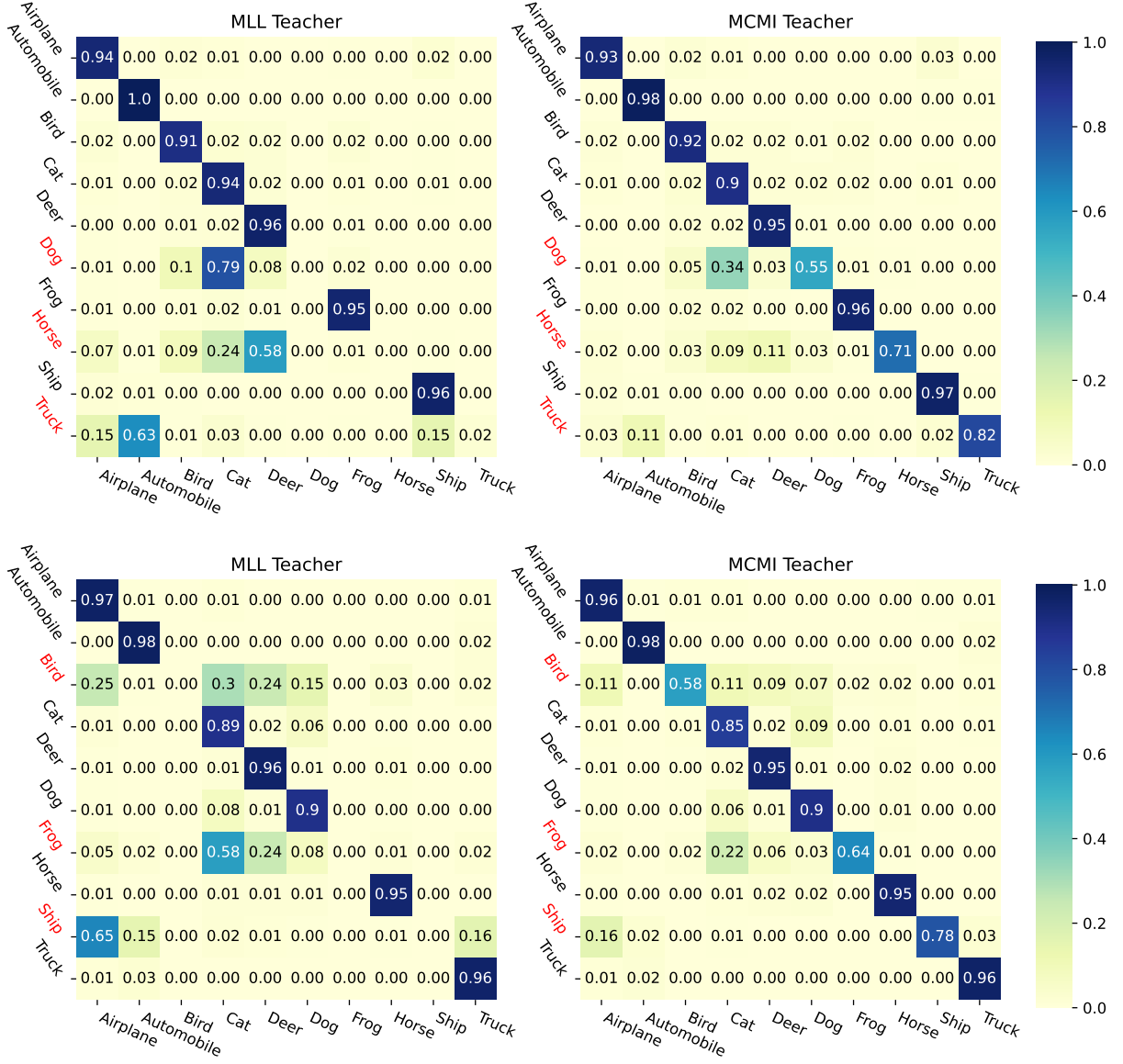


Figure A.4: Zero-shot KD when three classes are entirely removed for the student.

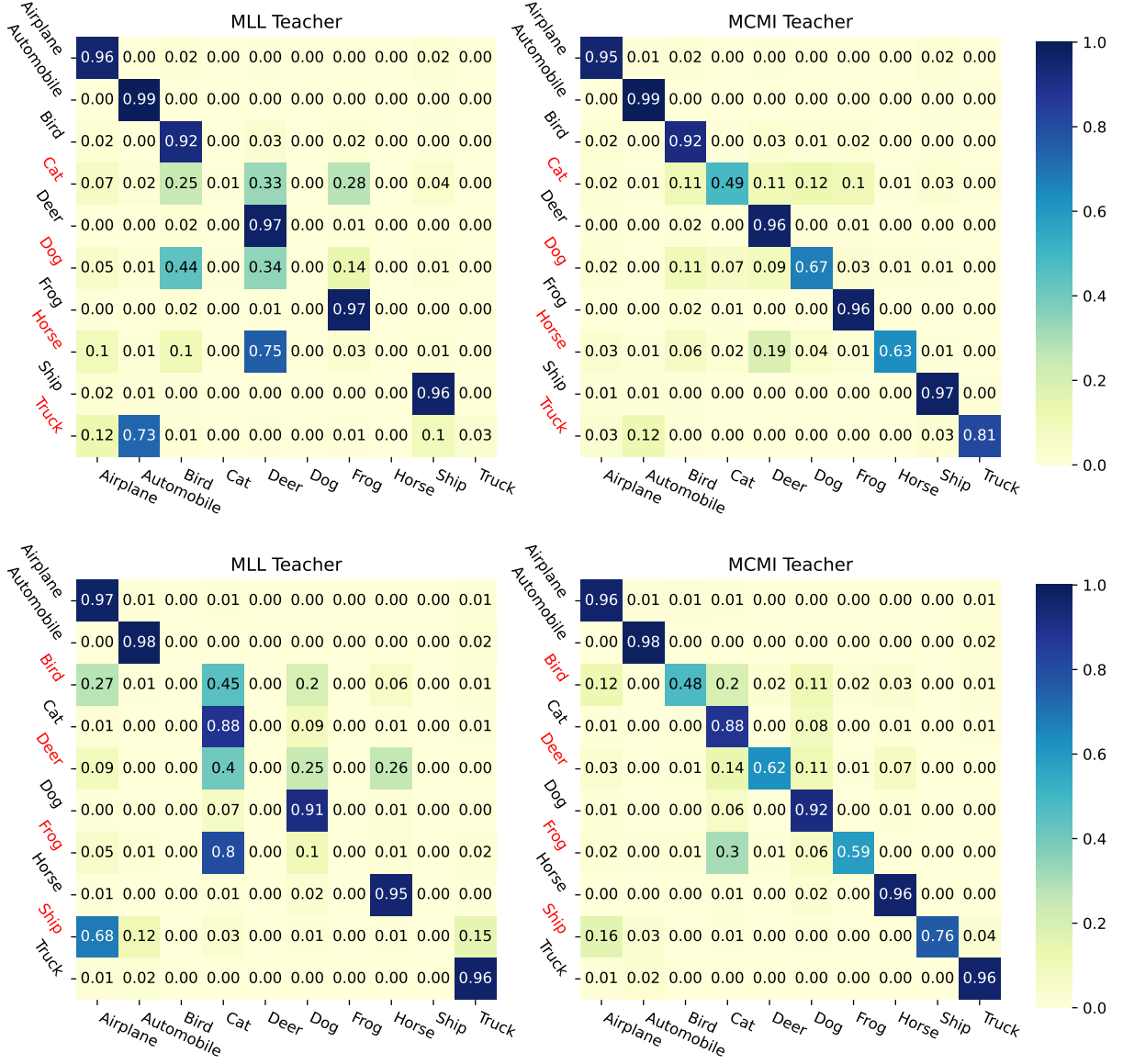


Figure A.5: Zero-shot KD when four classes are entirely removed for the student.

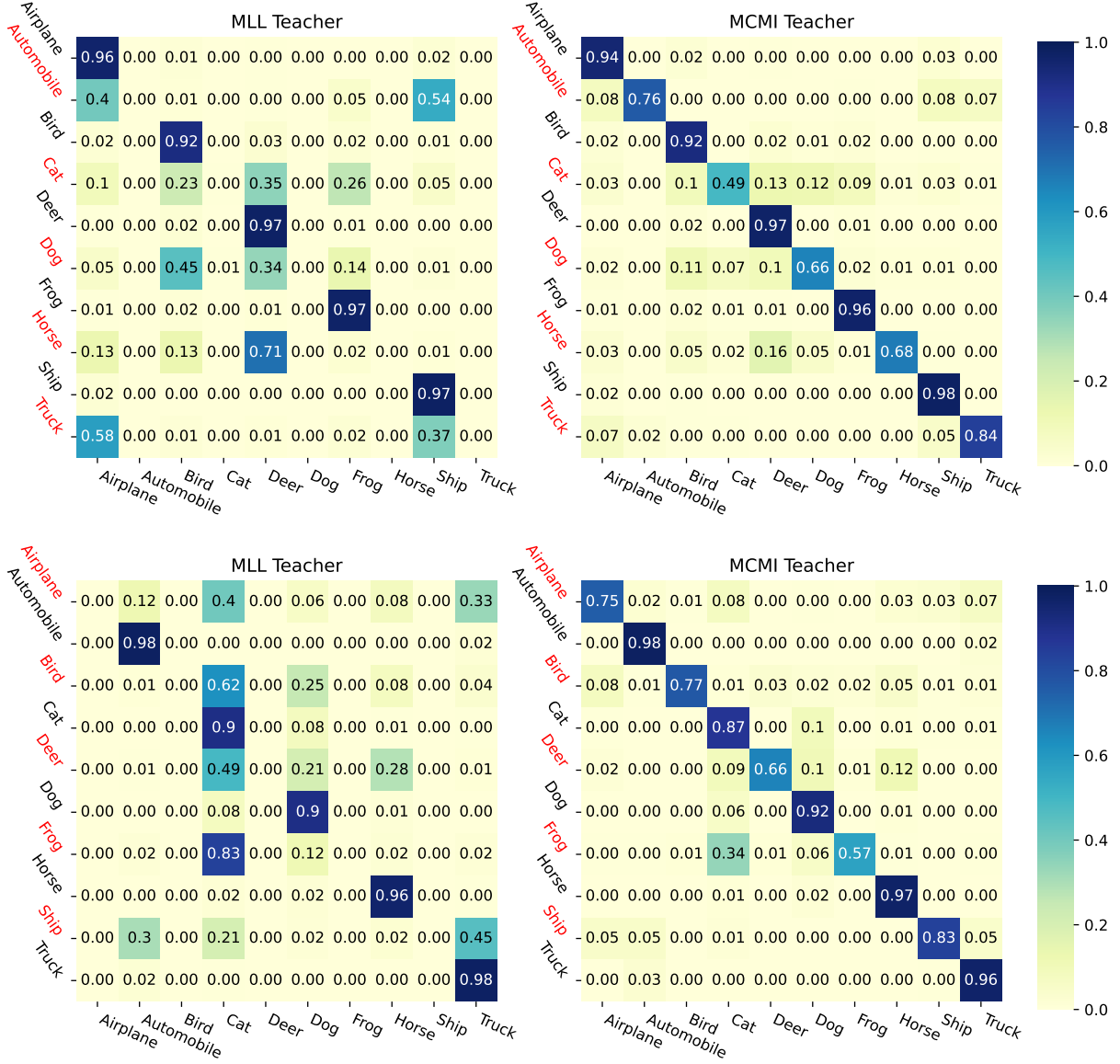


Figure A.6: Zero-shot KD when five classes are entirely removed for the student.

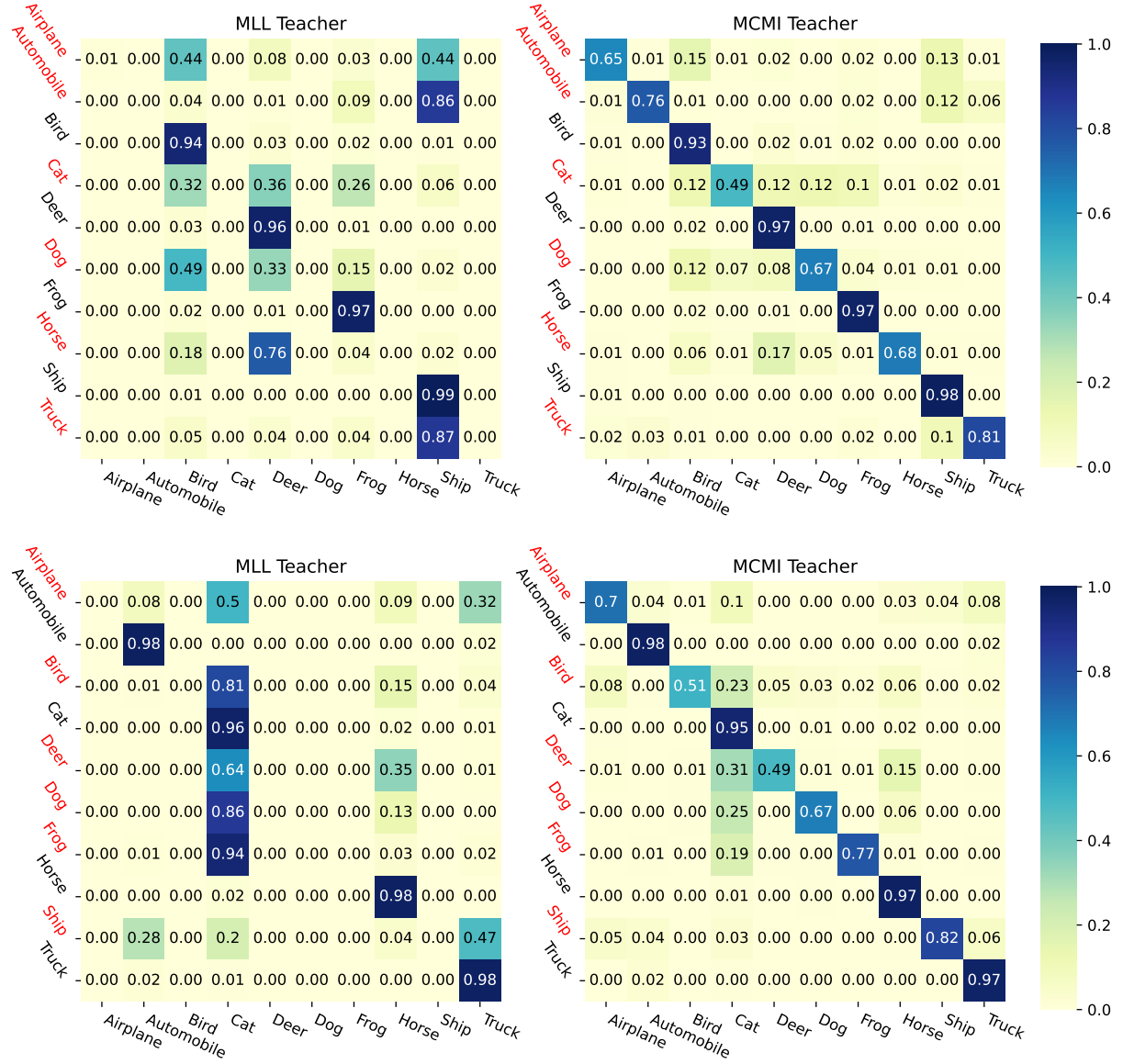


Figure A.7: Zero-shot KD when six classes are entirely removed for the student.

A.5 Proof of Theorem 1

Proof.

Since $\lambda \geq 0$, it is sufficient to show that suffices to show that

$$I(X; \hat{Y} | Y) = \max_{Q(x|\hat{Y}=i, Y=y)} \sum_{x,y} P(x,y) \sum_{i=1}^C P_x(i) \ln \frac{Q(x | \hat{Y} = i, Y = y)}{P_{X|Y}(x | y)}$$

The mappings from the label space Y to the input space X , and from the input space to the prediction $\hat{Y}$ of a classification model forms a Markov train.

$$Y \rightarrow X \rightarrow \hat{Y},$$

Then the CMI between $\hat{Y}$ and X given Y can be written as

$$I(X; \hat{Y} | Y) = \sum_{y \in [C]} P_Y(y) I(X; \hat{Y} | Y = y) \quad (\text{A.2})$$

$$= \sum_y \sum_x P(x,y) \sum_{i=1}^C P_x(i) \log \frac{P_x(i)}{P_{\hat{Y}|Y}(\hat{Y} = i | Y = y)} \quad (\text{A.3})$$

$$= \sum_y P_Y(y) \sum_x P_{X|Y}(x | y) \sum_{i=1}^C P_x(i) \log \frac{P_{X|Y}(x | y) P_x(i)}{P_{X|Y}(x | y) P_{\hat{Y}|Y}(i | y)} \quad (\text{A.4})$$

$$= \sum_y P_Y(y) \sum_{i=1}^C P_{\hat{Y}|Y}(i | y) \sum_x \frac{P_{X|Y}(x | y) P_x(i)}{P_{\hat{Y}|Y}(i | y)} \log \frac{P_{X|Y}(x | y) P_x(i)}{P_{X|Y}(x | y) P_{\hat{Y}|Y}(i | y)} \quad (\text{A.5})$$

$$= \max_{Q(x|\hat{Y}=i, Y=y)} \sum_y P_Y(y) \sum_{i=1}^C P_{\hat{Y}|Y}(i | y) \sum_x \frac{P_{X|Y}(x | y) P_x(i)}{P_{\hat{Y}|Y}(i | y)} \log \frac{Q(x | \hat{Y} = i, Y = y)}{P_{X|Y}(x | y)} \quad (\text{A.6})$$

$$= \max_{Q(x|\hat{Y}=i, Y=y)} \sum_{x,y} P(x,y) \sum_{i=1}^C P_x(i) \ln \frac{Q(x | \hat{Y} = i, Y = y)}{P_{X|Y}(x | y)} \quad (\text{A.7})$$

Note that in Eq. A.7 maximization is achieved when

$$Q(x | \hat{Y} = i, Y = y) = \frac{P_{X|Y}(x | y) P_x(i)}{P_{\hat{Y}|Y}(i | y)}. \quad (\text{A.8})$$

This completes the proof of Theorem 1.

A.6 Proof of Proposition 1

Proof. For a given model, we use f_X to represent the intermediate feature map to input X , as discussed in [82, 72], the layered structure if a model forms a Markov chain as:

$$X \rightarrow f_X \rightarrow \hat{Y}, \quad (\text{A.9})$$

where $\hat{Y}$ is the output of the model corresponding to the input X . By writing the CMI in terms of entropy, we obtain

$$I(X; \hat{Y} | Y) = H(X | Y) - H(X | \hat{Y}; Y) \quad (\text{A.10})$$

$$I(X; f_x | Y) = H(X | Y) - H(X | f_x, Y) \quad (\text{A.11})$$

On the other hand,

$$H(X | f_x; Y) = H(X | f_x; \hat{Y}, Y) \quad (\text{A.12})$$

$$\leq H(X | \hat{Y}; Y), \quad (\text{A.13})$$

where Eq. A.12 is due to the properties of Markov chain, and Eq. A.13 holds since removing a random variable from condition increase the entropy. This together with Eq. A.11 completes the proof.