Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A coded knowledge distillation framework for image classification based on adaptive JPEG encoding

Ahmed H. Salamah [*], Shayan Mohajer Hamidi, En-Hui Yang

Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada

ARTICLE INFO

MSC:
Knowledge distillation
Deep learning
Model compression
JPEG compression

ABSTRACT

In knowledge distillation (KD), a lightweight student model yields enhanced test accuracy by mimicking the behavior of a pre-trained large model (teacher). However, the cumbersome teacher model often makes over-confident responses, resulting in poor generalization when presented with unseen data. Consequently, a student trained by such a teacher also inherits this problem. To mitigate this issue, in this paper, we present a new framework of KD dubbed *coded* knowledge distillation (CKD) in which the student is trained to mimic instead the behavior of a *coded* teacher. Compared to the teacher in KD, the *coded* teacher in CKD has an additional adaptive encoding layer in the front, which adaptively encodes an input image into a compressed version (using JPEG encoding for instance) and then feeds the compressed input image to the pre-trained teacher. Comprehensive experimental results show the effectiveness of CKD over KD. In addition, we extend the deployment of a *coded* teacher to other knowledge transfer methods, showcasing its ability to enhance test accuracy across these methods.

1. Introduction

Recent advances in many computer vision tasks have been mainly obtained by deep neural networks (DNNs) that are large in size, and therefore require high computation and memory during inference [1,2]. Consequently, there has been a surge of research endeavors focused on developing techniques to reduce the size of DNNs, enabling their practical implementation on devices with limited resources.

One promising approach to creating smaller and more efficient DNNs is knowledge distillation (KD). In KD, a small DNN (student) is trained to mimic the behavior of a pre-trained large DNN (teacher) so that the trained student can generalize in a way similar to the teacher [1]. Followed by [1], many researchers tried to understand why distillation works [3–5], and proposed to distill different forms of knowledge from the teacher model. These forms include, for example, (i) the output probability vector response [1,6–8], (ii) feature maps [9–15], and (iii) relationships between different layers [16,17].

Despite achieving significant success, we notice that due to its typically large size, the teacher is often susceptible to overfitting the training set [18–20]. In other words, for most training samples, the teacher's output probability vector closely resembles a one-hot vector, where the sole non-zero entry corresponds to the correct class label. This characteristic impedes the teacher model's ability to generalize effectively for unseen samples, thereby diminishing its capacity to support the student in achieving good generalization.

This paper aims to tackle the aforementioned issue without resorting to retraining the teacher model. Instead, our approach focuses on inputting nonlinearly processed samples into the pre-trained teacher model to obtain improved and more generalizable output probability vectors. These vectors are then passed to the student model during the distillation process. To fulfill this goal, each training sample is first adaptively compressed and then fed into the pre-trained teacher model so that the following two conditions are satisfied:

- *Condition (I)*: If the teacher correctly classifies the original sample, the adaptively compressed sample is also correctly classified.
- *Condition (II)*: The output probability vector in response to the adaptively compressed sample is not as close to the one-hot probability vector as the output probability vector in response to the original sample.

During the distillation process, instead of passing the output probability vector of the teacher in response to the original sample, we pass the output probability vector of the teacher in response to the adaptively compressed sample to the student. Since generating a compressed sample involves an encoding process, we refer to such a modified KD framework as *coded* knowledge distillation (CKD), and the corresponding modified teacher as a *coded* teacher. In other words, compared to the teacher in KD, the *coded* teacher in CKD has an additional adaptive

[*] Corresponding author.
E-mail addresses: ahamsalamah@uwaterloo.ca (A.H. Salamah), smohajer@uwaterloo.ca (S.M. Hamidi), ehyang@uwaterloo.ca (E.-H. Yang).

https://doi.org/10.1016/j.patcog.2024.110966
Received 24 July 2023; Received in revised form 21 July 2024; Accepted 28 August 2024
Available online 2 September 2024

encoding layer in the front which encodes adaptively an input image into a compressed version and then feeds the compressed image into the teacher. In CKD, the student is trained to mimic instead the behavior of a *coded* teacher.

How would each original input image I be adaptively encoded by the *coded* teacher? In this paper, we shall apply JPEG. As a popular lossy compression technique, JPEG in its primitive form uses the quality factor (QF) parameter $q \in \mathcal{Q}$ to establish a trade-off between human-perceived quality and compression ratio (CR) [21]. Different QF values q result in different compressed images $\hat{I}$. The *coded* teacher then adaptively selects a suitable q for each input image I, which is optimal in some sense, and then uses JPEG to encode I. The contributions of the paper are summarized as follows:

- We present a new KD framework, dubbed as CKD, in which an additional adaptive encoding layer is introduced before the teacher, resulting in a *coded* teacher, for the purpose of generating better, more generalizable output probability vectors to be passed to the student model during the distillation process.
- Within our proposed CKD framework, an adaptive encoding method in conjunction with JPEG is presented and justified.
- We show by experiments that in comparison with KD, CKD along with the proposed *coded* teacher can achieve better performance. This improvement is particularly evident when tested on the ImageNet and fine-grain image classification datasets. Notably, on the CUB200-2011 and Stanford Dog datasets, we observed a significant percentage gain of 3.7% and 12.08%, respectively.
- In addition, further experiments on CIFAR-100 illustrate that the proposed *coded* teacher can also be used in conjunction with many other existing knowledge transfer methods–including FitNet [9], FSP [11], FT [12], CC [17], SP [14], AB [13], and RKD [16]–to improve validation accuracy. We show that the gain of such a combination can be up to 5% over the underlying knowledge transfer method compared to TALD [22].

2. Related work

• **Knowledge distillation.** KD aims to distill the knowledge learned by a complex teacher model into a smaller student model, allowing the student model to achieve performance comparable to that of the teacher model while being computationally more efficient. To achieve this, the student loss function typically comprises a conventional cross-entropy term combined with an additional component representing the knowledge distilled from the teacher model throughout the training process. In its original form, KD involves minimizing the KL-divergence between the output probability vector of the teacher and that of the student model, where the output probability vector of the teacher is the knowledge passed to the student. Since the seminal paper by Hinton [1], many different forms of knowledge from the teacher have been investigated and passed to the student. Below, we briefly discuss some of them, with a specific focus on those that we later compare our method to in our experiments.

• **Intermediate-level distillation.** The intuition behind intermediate-level KD is to exploit the rich representations learned by the teacher model at different layers of its architecture. A pioneering work in this realm is FitNets [9], where the authors suggested using the responses from multiple intermediate hidden layers of the teacher, referred to as hints, to distill knowledge into the student model. Inspired by this work, in attention transfer (AT) [10], the attention maps generated by the teacher model guide the student to learn similar attention patterns. On the other hand, factor transfer (FT) [12] presents a different approach where transportable features are extracted using convolutional operations to paraphrase the teacher's knowledge. In another approach, Heo et al. [13] focus on transferring knowledge by matching the activation boundaries (AB) formed by hidden neurons between a teacher and a student model. The activation boundaries refer to the regions in the input space where the teacher model's activations change for the neuron's response (active or not), yielding different predictions.

Moreover, intermediate feature distributions can also play a crucial role in KD. Flow-based KD [11], known as FSP, aims to match the distributions of intermediate features of the student model with those of the teacher. The matching is typically achieved by minimizing the KL divergence or the maximum mean discrepancy, between the feature distributions of the teacher and student models. In a similar vein, similarity-preserving (SP) knowledge distillation [14] is a KD method based on the fact that inputs with similar semantics tend to generate similar activation patterns. Hence, SP focuses on preserving pairwise similarities within the teacher's feature map by transforming it into a matrix. This matrix encoding retains the similarity information of activations at the teacher layer. However, this transformation process may result in some information loss.

Another line of methodology to leverage intermediate layers in KD, involves harnessing the relationship between samples to improve knowledge transfer. Contrastive representative distillation (CRD) [15] encourages representations of similar samples to be closer in the embedding space, while simultaneously pushing apart those of different samples, from different classes. To this aim, this method exploits the correlations and higher-order output dependencies. Relational knowledge distillation (RKD) captures the pairwise relationships between the data points [16]. This can be achieved through various techniques, such as penalizing the structure difference with distance-wise and angle-wise distillation loss between the data points and encouraging the student model to reproduce similar relationships as the teacher model. In correlation congruence for knowledge distillation (CC) [17], the goal is to ensure that the student model's predictions exhibit similar correlation patterns to those of the teacher model. In addition to using correlation loss, the benefits of deploying other loss functions such as covariance loss or higher-order statistical moments are also discussed in [17].

Nevertheless, intermediate-level KD adds complexity to the distillation process compared to the logit-based KD. This, in turn, increases the computational cost and training time of intermediate-level KD methods.

• **KD using input perturbation.** In the conventional KD, both the teacher and student models are fed with the same input samples (raw images) during the distillation process. Yet, these raw images might not be able to explore the properties of the teacher model properly. Thus, Heo et al. [6] propose a KD method in which a teacher pre-trained by clean (raw) images is fed with adversarial examples during the KD process. Particularly, an attack scheme is modified to explore the properties of the teacher model by searching an adversarial sample supporting a decision boundary, referred to as boundary supporting sample (BSS). However, the current BSS attack method has limitations in fully exploring the complete range of potential perturbations that the teacher model can exhibit, as it is restricted to generating a single adversarial example. To tackle this issue, TALD [22] addresses the insufficiency of a single adversarial example by examining a broader spectrum of possible teacher model perturbations through the generation of multiple adversarial examples for the same original sample. TALD generates diverse adversarial examples using a multiple particle-based search technique known as Stein Variational Gradient Descent (SVGD). The authors of [22] execute SVGD for a specified number of iterations, denoted as L, to generate each sample sequentially, employing K particles sampled from the teacher conditional adversarial local distribution, with each iteration involving forward and backward propagation. Additionally, in [23], the authors propose a framework that incorporates an adversarial phase utilizing GANs to generate diverse examples. Afterward, a co-distillation phase involving multiple classifiers is employed to leverage the divergent examples for enhancing the distillation performance which is computationally expensive.

Although compressed images generated by a *coded* teacher in CKD can also be regarded as perturbed input images for the teacher, they serve a completely different purpose. In fact, these compressed images are generated and employed in CKD to prevent the over-confident responses of the teacher from being passed to the student during the distillation process. Later, we will show through experiments that indeed CKD holds a greater promise compared to these methods with perturbed raw images as adversarial examples.

3. Preliminaries and motivation

3.1. Preliminaries

3.1.1. Knowledge distillation formulation

As previously discussed, Hinton et al. [1] introduced response-based knowledge, represented by the output probability vector of the teacher, which is incorporated into the training process through the KD loss function. The KD loss function comprises two essential components: (i) the traditional cross-entropy loss function $\mathcal{H}(\cdot,\cdot)$, and (ii) a regularization term accounting for the teacher supervision. Specifically, the loss function in KD could be written as

$$\mathcal{L}_{KD} = (1-\alpha)\mathbf{E}_{I\sim P}\mathcal{H}(\hat{y}, p^s) + \alpha\tau^2\mathbf{E}_{I\sim P}\mathcal{L}_{soft}\big(p^t_\tau, p^s_\tau\big), \tag{1}$$

where P is the empirical distribution of the original training sample image I, $\hat{y}$ is the one-hot probability vector corresponding to the ground truth label, p^s is the output probability vector of the student in response to the input I, and $\alpha \in [0,1]$ is a factor to balance the two loss terms. In Eq. (1), $p^t_\tau = p_\tau(I;\boldsymbol{\theta}^t)$ and $p^s_\tau = p_\tau(I;\boldsymbol{\omega}^s)$ stand for the teacher's and student's output probability vectors after softening the logits with a temperature τ, respectively, and $\mathcal{L}_{soft}$ is the response-based knowledge loss defined as follows

$$\mathcal{L}_{soft} = \mathcal{KL}\big(p^t_\tau \,\big\|\, p^s_\tau\big), \tag{2}$$

where $\mathcal{KL}(\cdot \,\|\, \cdot)$ represents KL divergence, and $\boldsymbol{\theta}^t$ and $\boldsymbol{\omega}^s$ are the teacher and student models' parameters, respectively. Note that in the traditional KD, both the teacher and student accept the same original sample image I as their respective inputs during the KD process.

3.1.2. JPEG compression

JPEG [21] is widely used for lossy compression in digital image processing while maintaining visual quality. The JPEG encoding process includes three fundamental phases: discrete cosine transformation (DCT), quantization, and entropy encoding. It begins with the partitioning of the input image into 8 × 8 blocks, systematically processed in a raster scan order. Each block undergoes a transformation from the pixel domain to the DCT domain via an 8 × 8 DCT. The resulting DCT coefficients are subsequently subjected to uniform quantization using an 8 × 8 quantization table, with its entries specifying the quantization step sizes for each frequency bin, thereby representing the only lossy step, i.e. non-linear. This quantization table for JPEG encoding is designed in alignment with the assigned quality factor (QF). The DCT indices resulting from quantization are further encoded through a combination of run-length coding and Huffman coding, yielding the final bitstream. The trade-off between compression ratio (CR) and compression quality is mainly impacted by the designed quantization step controlled by the chosen QF value; that is, a higher QF yields a lower CR and better compression quality. The level of compression can be adjusted by selecting a QF value from 1 to 100.

3.1.3. Overfitting

Overfitting occurs when a DNN becomes excessively tailored to a finite number of training samples, resulting in overconfidence, which leads to a diminished ability to generalize well to unseen data [24,25]. This phenomenon arises when the network learns intricate details and noise present in the training set, rather than capturing the underlying patterns and relationships that are more relevant for making accurate predictions on unseen data. As a result, the network's performance tends to degrade when it is fed with new examples that differ from the training data, thereby compromising its overall effectiveness. In particular, this phenomenon is commonly observed in large models that possess a significant number of parameters.

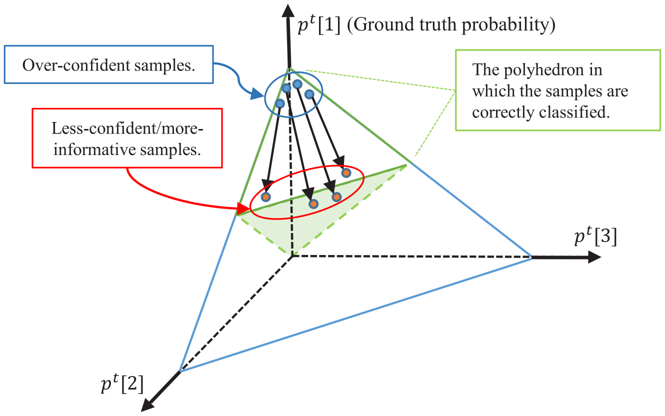

$p^t[1]$ (Ground truth probability)
Over-confident samples.
The polyhedron in which the samples are correctly classified.
Less-confident/more-informative samples.
$p^t[3]$
$p^t[2]$

Fig. 1. The simplex of output probability vector for a classification task with three classes. Our goal is to alter the original samples with peaky output probabilities (the blue dots) in order to obtain samples whose output probabilities are more uniform (the red dots). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Motivation

In the context of KD, when the teacher model suffers from overfitting, it is unable to effectively impart generalized knowledge to the student. Thus, our motivation in this work is to prevent the overconfident responses of the teacher from being passed to the student during the distillation process. To this end, during the distillation we feed the pre-trained teacher model with a compressed version of the input image such that the teacher's output probability vector for the compressed sample is more toward the uniform distribution.

To clarify our objective, consider a classification task with three classes, namely $\{c_1, c_2, c_3\}$. In this case, the teacher's output probability vector p^t has three entries denoted by $\{p^t[1], p^t[2], p^t[3]\}$, where $p^t[i]$ corresponds to the output probability for c_i, for $i \in \{1,2,3\}$. Fig. 1 shows the simplex of output probability vectors of a pre-trained teacher is depicted when its inputs are samples from class c_1. The teacher would correctly classify those samples $I \in c_1$ for which the teacher's output probability vector lies on the surface of the polyhedron defined by $p^t[1] > p^t[2]$ and $p^t[1] > p^t[3]$.

However, the teacher makes over-confident decisions for the samples whose output probabilities lie in the corner of the simplex, meaning that the probability vectors generated by the teacher for these samples are close to one-hot vectors. We refer to such probability vectors as *peaky* ones hereafter. In this case, the model shows over-confident responses to these specific samples and, in turn, cannot generalize well when fed with new unseen data of a similar type [24,25].

To prevent peaky probability vectors from being passed to the student during the distillation process, it is desirable to alter the respective input images so that (i) the altered input images are still similar to the original ones; and (ii) the output probability vectors of the teacher in response to the altered input images are closer to a uniform distribution while remaining on the surface of the correct polyhedron, similar to the red samples in Fig. 1. One way to achieve this is through compression, which will be discussed in the next section.

4. *Coded* knowledge distillation

As discussed in Section 3.2, we have to manipulate the training samples so that the teacher's output probabilities for the new samples are more uniform. We aim to realize this by compressing the input samples to the pre-trained teacher model during the distillation process. Specifically, we deploy JPEG, the widely-used lossy compression technique. The key rationales behind utilizing JPEG are (i) its popularity as a lossy compression technique, and (ii) its potential for improving accuracy as pointed out by Yang et al. [26].

It is widely recognized that naively compressing all the input images to a DNN using the same QF by JPEG will generally lead to a degradation in the classification accuracy [26–28]. However, Yang et al. [26] demonstrated that if QF can be adaptively selected on a per image basis, then JPEG compression can significantly enhance classification. Nevertheless, in their framework, the adaptive selection of QF depends on the availability of the ground truth label of the original image, which makes it unrealistic during the testing stage. Inspired by this, we here employ an adaptive selection of QF during the distillation process instead for the teacher's input only, where a ground truth label is available and can indeed be utilized. That is, we adaptively select an image-dependent QF for JPEG to compress each image before it is fed to the teacher during the distillation process. To achieve this, an adaptive encoding layer is introduced in front of the teacher model. The combination of this new adaptive encoding layer with the originally pre-trained teacher forms what we refer to as a *coded* teacher. The output $\hat{I}_{j^*}$ from the adaptive encoder layer generally depends on the input I, the ground truth label c, and the teacher model itself. In the subsequent subsection, we elaborate on the inner-working of the utilized adaptive encoder layer.

4.1. Optimal adaptive JPEG selection

To realize the proposed adaptive JPEG selection, for each sample image I, we first generate a set of its compressed versions as explained in Section 4.1.1; then, in Section 4.1.2, we propose a systematic method to select one element from this set as the input to the teacher.

4.1.1. The set of compressed versions

Consider a predetermined set of QFs: $\mathcal{Q} = \{q_j \in \mathbb{N} \mid 0 \leqslant q_j \leqslant 100\}$, where $q_j = \delta \times j$, and δ is used as a step-size between two successive QF values in $\mathcal{Q}$. Denote by $\hat{I}_j$ the compressed version of I using QF equal to q_j. Then, for each image I in the training set, we construct the following set $\mathcal{J}_I = \{\hat{I}_j \mid q_j \in \mathcal{Q}\} \cup \{I\}$. Specifically, $\mathcal{J}_I$ contains the original training sample image I and its different compressed versions obtained using different $q_j \in \mathcal{Q}$. Therefore, the size of the $\mathcal{J}_I$ will depend on the selected value of δ, and obtained as $|\mathcal{J}_I| = \lfloor \frac{100}{\delta} \rfloor + 2$.

4.1.2. Selection criterion

With the set $\mathcal{J}_I$ created, this section focuses on presenting a systematic method to select an element from $\mathcal{J}_I$. This selection mechanism should ensure that both *Conditions (I)* & *(II)*–introduced in Section 1–are satisfied. As such, we first satisfy *Conditions (I)* by finding a subset of $\mathcal{J}_I$ whose elements are correctly classified by the teacher; that is, we find the set $\mathcal{J}_I^{Correct}$[1] as follows

$$\mathcal{J}_I^{Correct} = \{\hat{I}_j \in \mathcal{J}_I \mid \operatorname*{argmax}_{z \in C} p(z|\hat{I}_j; \boldsymbol{\theta}^t) = c\}, \tag{3}$$

where C represents the set of classes. Afterward, we aim to satisfy *Condition (II)* on top of *Conditions (I)*. To achieve this objective, given that the teacher's output probability for the original sample $p(I; \boldsymbol{\theta}^t)$ is typically peaky, we select $\hat{I}_j \in \mathcal{J}_I^{Correct}$ whose output probability vector $p(\hat{I}_j; \boldsymbol{\theta}^t)$ has the maximum distance to $p(I; \boldsymbol{\theta}^t)$. For this purpose, among the possible distance metrics for the probability vectors, we utilize KL divergence. We refer to such selection mechanism as high KL (HKL) divergence selector. Henceforth, we find $\hat{I}_{j^*}$ as

$$\hat{I}_{j^*} = \operatorname*{argmax}_{\hat{I}_j \in \mathcal{J}_I^{Correct}} \mathcal{KL}\Big(p(I; \boldsymbol{\theta}^t) \ \Big\| \ p(\hat{I}_j; \boldsymbol{\theta}^t)\Big). \tag{4}$$

Using this methodology, in the *coded* teacher, for every original training sample image I, we select $\hat{I}_{j^*}$ given in Eq. (4) as the teacher's input during the distillation process. The CKD framework along with its adaptive JPEG encoding layer are depicted in Fig. 2.

4.2. Analyzing the proposed adaptive JPEG selection

In this subsection, we provide justifications for the proposed adaptive selection mechanism by presenting compelling experimental results. For this purpose, we use the following two values to evaluate whether a probability vector is peaky:

- The magnitude of ground truth probability, in the sense that a smaller magnitude tends to result in a smoother and less peaked probability distribution.
- The uniformity of the vector; that is, a more uniform vector is less peaked. Among the various existing metrics for measuring the uniformity of a vector, we specifically use entropy in our experiments. As such, a probability vector with higher entropy is more uniform.

By using these two metrics, we can assess the effectiveness of the adaptive selection method in promoting a more uniform distribution of output probabilities. Now, consider the following three experiments.

Experiment one: Using randomly picked samples from CIFAR100's training set, we want to observe the elements in the set we constructed in Section 4.1.1 to assess their level of peakiness or uniformity. To illustrate this, we utilize a pre-trained Resnet56 model, serving as the teacher, trained on the CIFAR100 dataset. We then randomly select two images, Img1 and Img2, belonging to different classes that are correctly and wrongly classified by the teacher, respectively. Then, we construct the sets $\mathcal{J}_{\text{Img1}}$ and $\mathcal{J}_{\text{Img2}}$ for these two samples (as explained Section 4.1.1). A caricature of decision boundary for the samples in $\mathcal{J}_{\text{Img1}}$ and $\mathcal{J}_{\text{Img2}}$ are depicted in Fig. 3 in which we use the following notation: (i) 'KL' represents $\mathcal{KL}\big(p(I; \boldsymbol{\theta}^t) \ \big\| \ p(\hat{I}_j; \boldsymbol{\theta}^t)\big)$, (ii) 'H' is the entropy of $p(\hat{I}_j; \boldsymbol{\theta}^t)$, and (iii) 'P' is the magnitude of the ground truth (GT) probability. The 'KL' value is written on top of the arrows connecting the original sample I to its compressed versions $\hat{I}_j$. In addition, for each compressed sample, the values for QF, 'H', and 'P' are written in a box beside that sample.

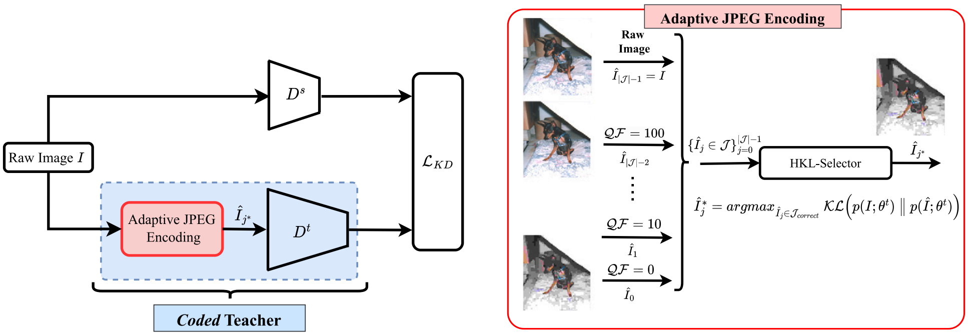
For Img1, (the left figure in Fig. 3), we observe that only seven elements in $\mathcal{J}_{\text{Img1}}$ with respective QF=$\{60, 75, 80, 90, 95, 100\}$ are correctly classified. Among these samples, the compression version with QF=60 will be chosen using the adaptive JPEG selector, as it has the highest 'KL' value. The selected compressed version by the HKL selector has the lowest 'P' and largest 'H' among all correctly classified samples. This demonstrates that the proposed selection method can successfully choose the least peaked sample from the set $\mathcal{J}_{\text{Img1}}^{correct}$.

Now, consider Img2, which is originally misclassified by the model. As seen in Fig. 3 (the right figure), there exist some elements in $\mathcal{J}_{\text{Img2}}$ which are correctly classified by the model. Again, the selected compressed version by HKL selector still gives rise to good values of 'P' and 'H'. Note that among all training samples, samples like Img2 are rare, as shown in the upper part of Table 1.

Experiment two: In this experiment, we aim to analyze the behavior of the *coded* teacher over all the training samples (unlike the Experiment one where we only considered two specific samples), and compare it with that of the original uncoded teacher. In fact, we want to demonstrate that the adoption of such an adaptive selection method assists the teacher model in avoiding the generation of *peaky* probabilities. To this end, we report the following two values for the teacher model: the average (and standard deviation) of (i) the entropy of the output probability vectors, and (ii) the GT probability, where the average and standard deviation are computed over all training samples.

Again, we use CIFAR100 dataset. The average entropy and GT probability are computed for both the *coded* teacher and the original uncoded teacher. They are shown in Table 1 for various model architectures to illustrate the generality of the proposed method. As seen in Table 1, compared to the original teacher, the *coded* teacher gives rise to higher average entropy and smaller ground truth probability

[1] If $\mathcal{J}_I^{Correct}$ is empty, we use the original sample.


Raw Image I
D^s
D^t
Adaptive JPEG Encoding
$\hat{I}_{j^*}$
$\mathcal{L}_{KD}$
Coded Teacher
Adaptive JPEG Encoding
Raw Image
$\hat{I}_{|\mathcal{J}|-1} = I$
$\mathcal{QF} = 100$
$\hat{I}_{|\mathcal{J}|-2}$
$\mathcal{QF} = 10$
$\hat{I}_1$
$\mathcal{QF} = 0$
$\hat{I}_0$
$\{\hat{I}_j \in \mathcal{J}\}_{j=0}^{|\mathcal{J}|-1}$
HKL-Selector
$\hat{I}_{j^*}$
$\hat{I}^*_j = argmax_{\hat{I}_j \in \mathcal{J}_{correct}} \mathcal{KL}\left(p(I;\theta^t) \parallel p(\hat{I};\theta^t)\right)$

(a) CKD Framework

(b) Adaptive JPEG Encoding Layer

Fig. 2. Illustration of CKD: (a) the CKD framework with an adaptive JPEG encoding based *coded* teacher; and (b) the inner-working of adaptive JPEG encoding mechanism.

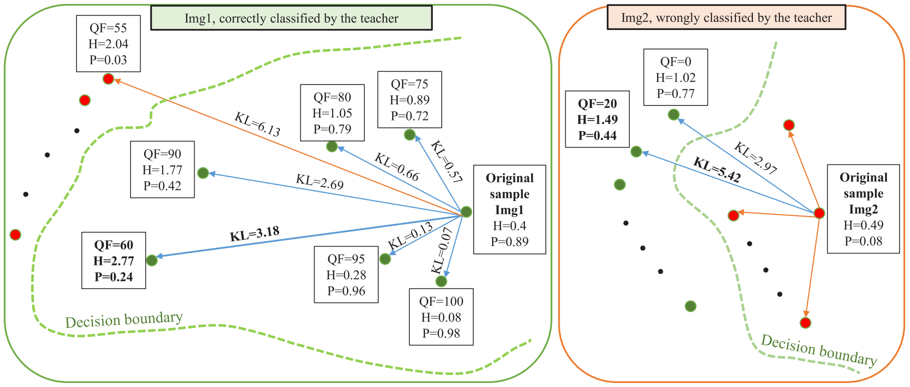


Fig. 3. The decision boundary plots for Resnet56 pre-trained on CIFAR100 dataset, when the inputs are the elements from the sets: (left) $\mathcal{J}_{\text{Img1}}$, and (right) $\mathcal{J}_{\text{Img2}}$. Green and Red dots show the correctly and wrongly classified samples, respectively. Note that the raw images Img1 and Img2 are correctly and wrongly classified, respectively.

Table 1
The average (and standard deviation) of (i) the entropy of p_I and $p_{\hat{I}}$, and (ii) the GT probability in p_I and $p_{\hat{I}}$ from both the *Coded* teacher and original teacher, respectively, over CIFAR100 dataset, where p_I and $p_{\hat{I}}$ are generated using $\tau = 1$. The pre-trained teacher models are also used in [7,15,22,29–32].

Teacher models	*Coded* teacher		Original teacher	
	Entropy	GT probability	Entropy	GT probability
Resnet56	1.4941 (±0.6634)	0.5184 (±0.1992)	0.24531 (±0.3816)	0.9275 (±0.1476)
Resnet110	1.3312 (±0.6513)	0.5607 (±0.2032)	0.10879 (±0.2195)	0.9730 (±0.0781)
VGG13	1.6524 (±0.8211)	0.5344 (±0.2205)	0.03797 (±0.0883)	0.9937 (±0.0265)
WRN-40–2	1.5653 (±0.7578)	0.5244 (±0.2121)	0.07115 (±0.1584)	0.9859 (±0.0458)
ResNet50	1.4399 (±0.7689)	0.5662 (±0.2207)	0.02423 (±0.0648)	0.9960 (±0.0200)
Resnet32 × 4	1.8490 (±0.9192)	0.5004 (±0.2289)	0.02731 (±0.0656)	0.9962 (±0.0173)

for all the tested models, confirming that the *coded* teacher produces probability vectors more towards the uniform distribution.

Experiment three: Here, we aim to show that CKD enhances the generalization of the student more significantly compared to other KD variants. For this purpose, we have conducted the following experiment: for three teacher–student pairs, we have reported the average (and standard deviation) of (i) the entropy of the student's output probability vectors, and (ii) the student's GT probability for a student trained via some KD variants (see Table 2). As depicted in Table 2, most of the KD variants result in improvement for the generalization of the student. However, the CKD method exhibits notably a significantly greater improvement and demonstrates orthogonality to temperature scaling in conventional KD. As we will discuss in the following section, it is the heightened level of uniformity passed from the *coded* teacher to the trained student that enables the latter to achieve higher accuracy compared to its baseline distiller. Nevertheless, while training a teacher model with label smoothing increases the degree of uniformity, it can negatively impact the student's accuracy by erasing information contained in intra-class relationships among individual training samples [33]. For more additional analysis, refer to the Appendix A.

5. Experiment results

To evaluate the performance of CKD, we conduct various experiments. In this section, we report our experimental results and compare them with the state-of-the-art alternatives in the literature. In particular, we have carried out experiments on four datasets, namely ImageNet [34] in Section 5.1, and CIFAR-100 [35] in Section 5.2, and two different datasets for the fine-grained classification in Section 5.3. For each of these datasets, both teacher and student models are mentioned, and the training setup is also elaborated. Additionally, we highlight the adaptability of using *coded* teachers by integrating them into other well-established knowledge transfer methods found in the literature. This integration results in an improvement in the test accuracy for each respective approach. All experiments were conducted using PyTorch [36].

5.1. ImageNet dataset

- **Dataset:** We use ImageNet ILSVRC 2012 dataset that contains 1.2M training images and 50K for testing images [34] with an average size of

Table 2
The average (and standard deviation) of (i) the entropy of p_I, and (ii) the GT probability in p_I from students trained by CKD with the *Coded* teacher and other knowledge transfer methods with the original teacher on the CIFAR-100 dataset, where p_I is generated using $\tau = 1$.

Methods	Resnet56 → Resnet20		Resnet32 × 4 → Resnet8 × 4		VGG13 → MobileNetV2	
	Entropy	GT probability	Entropy	GT probability	Entropy	GT probability
KD	0.6423 (±0.7002)	0.7623 (±0.2972)	0.5386 (±0.6714)	0.8457 (±0.2283)	0.3751 (±0.5116)	0.8763 (±0.2117)
FitNet	0.7307 (±0.7434)	0.7488 (±0.2937)	0.6253 (±0.6817)	0.8125 (±0.2410)	0.4605 (±0.5389)	0.8600 (±0.2074)
CC	0.7308 (±0.7393)	0.7497 (±0.2923)	0.5574 (±0.6540)	0.8337 (±0.2304)	0.6194 (±0.6498)	0.7958 (±0.2616)
FT	0.8552 (±0.8050)	0.7010 (±0.3155)	0.7492 (±0.7834)	0.7672 (±0.2770)	1.1749 (±0.8641)	0.5945 (±0.3372)
AB	0.7427 (±0.7466)	0.7464 (±0.2931)	0.5111 (±0.6207)	0.8526 (±0.2116)	0.4044 (±0.5054)	0.8768 (±0.1966)
SP	0.8247 (±0.7827)	0.7099 (±0.3129)	0.7595 (±0.8167)	0.7642 (±0.2858)	0.5675 (±0.7730)	0.7215 (±0.3225)
RKD	0.7547 (±0.7266)	0.7333 (±0.3014)	0.6333 (±0.6655)	0.8143 (±0.2366)	0.5104 (±0.5692)	0.8415 (±0.2224)
FSP	0.8118 (±0.7905)	0.7188 (±0.3096)	0.5587 (±0.6525)	0.8314 (±0.2327)	n/a	n/a
TALD	0.5396 (±0.6543)	0.7430 (±0.3412)	0.4013 (±0.6306)	0.8448 (±0.2761)	0.3333 (±0.5350)	0.8070 (±0.3293)
CKD	**0.9511 (±0.8326)**	**0.7001 (±0.3018)**	**0.9651 (±0.9541)**	**0.7764 (±0.2544)**	**0.6468 (±0.7211)**	**0.8367 (±0.2257)**

Table 3
The student accuracy on ImageNet dataset given by different KD methods, where the best and second-best results are **bold** and underlined, respectively. The experiment setup is the same as in [15]. *As we followed [37] for our experimental setups, the accuracy for the Resnet18 model differs from that reported in [38].

Accuracy	Teacher	Student	KD [1]	AT [10]	SP [14]	CC [17]	O-KD [39]	CKD
Top-1	73.31%	69.76%*	70.66%	70.70%	70.62%	69.96%	70.55%	**70.98%**
Top-5	91.42%	89.08%	89.88%	90.00%	89.80%	89.17%	89.59%	**90.04%**

469 × 387 to allow us to verify distillation performance for large input resolutions.

- **Teacher and student models:** We use pre-trained Resnet34 as the teacher and Resnet18 as the student model [38]. Following the experimental setup in [7,15,29–32] , the pre-trained teacher model is obtained from torchvision library [37].
- **Training setup:** The training is performed for 100 epochs with a batch size of 256 using SGD with a momentum of 0.9 and weight decay equal to 1×10^{-4} with an initial learning rate γ of 0.1 that is divided by 10 at epochs 30, 60 and 90. Standard Inception-style preprocessing [24] for augmentation settings is used in the training of teacher models as well as students. For CKD, we set $\delta = 10$ resulting in $|\mathcal{J}_I| = 12$ versions of the raw input.
- **Results and analysis:** The results are presented in Table 3, where the performance of CKD is compared with some knowledge transfer benchmarks. We note that we used O-KD to denote "Online KD" method [39]. Specifically, the results demonstrate that the *coded* teacher effectively improves the performance of the conventional KD, surpassing the performance of the AT distillation method. In addition, further experiments on the subset of ImageNet, namely ImageNet-1K subset, are presented in Appendix B.

5.2. CIFAR100 dataset

- **Dataset:** This dataset contains 50K training images with 0.5K images per class and 10K test images [35].
- **Teacher and student models:** We use different pairs of teacher and student models, considering both similar and different architectural styles. Specifically, inspired from [7,15,29–32], the teacher models are chosen among ResNet50, Resnet56, Resnet110, VGG13, WRN-40-2, and Resnet32x4.
- **Training setup:** When training CIFAR-100, we start with an initial learning rate of 0.01 or 0.05. After the first 150 epochs, we decrease the learning rate by a factor of 0.1 every 30 epochs until reaching the last 240 epochs. The initial learning rate is 0.01 for MobileNetV2, ShuffleNetV1, and ShuffleNetV2, and 0.05 for the other models. For CKD, we use $\delta = 5$, yielding $|\mathcal{J}_I| = 22$. The training setup follows the commonly used approach introduced in [7,15,29–32].
- **Results and analysis:** We conduct three sets of experiments explained below.

Set 1: We first compare the performance of CKD against conventional KD methods. In addition, we compare CKD to other perturbation-based KD methods, namely BSS [6] and TALD [22]. As discussed in Section 2, these two methods are similar to CKD in the sense that both effectively perturb the inputs to the pre-trained teacher model. We use the same student–teacher pairs as those used in [22], where both the same and different architectural styles were considered. The comparison is reported in Table 4.

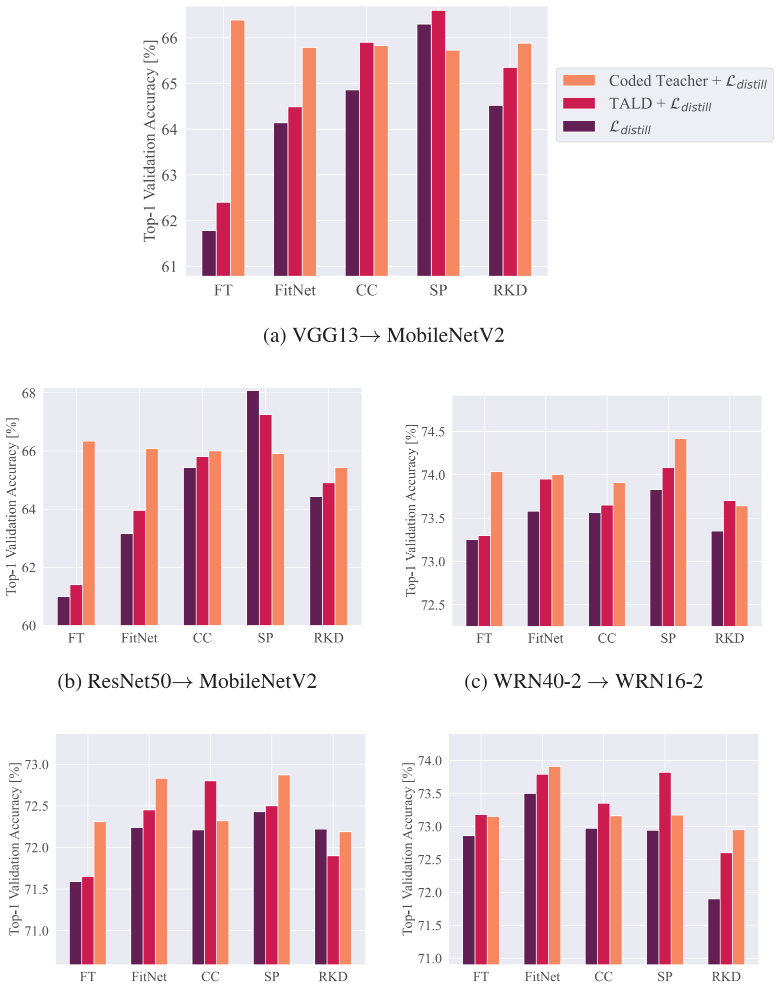
Table 4
Test accuracy (%) of students trained by different KD methods on CIFAR100, where the best and second-best results are **bold** and underlined, respectively. The student accuracy for TALD and BSS is obtained using the settings specified in [22].

Teacher Student	Resnet56 Resnet20	Resnet32 × 4 Resnet8 × 4	VGG13 MobileNetV2	ResNet50 MobileNetV2
Student	69.06%	72.50%	64.6%	64.6%
KD [1]	70.66%	73.33%	67.37%	67.35%
AT [10]	70.55%	73.44%	59.40%	58.58%
VID [40]	70.38%	73.09%	65.56%	67.57%
PKT [41]	70.34%	73.64%	67.13%	66.52%
NST [42]	69.60%	73.30%	58.16%	64.96%
BSS [6]	70.70%	73.53%	67.43%	68.10%
TALD [22]	**70.90%**	73.73%	**68.50%**	**68.70%**
CKD	70.81%	**74.42%**	67.80%	68.41%

Remarkably, CKD outperforms KD and its other four variants, namely AT, VID, PKT and NST. In particular, CKD exhibits an average gain of 0.67% compared to conventional KD. This superior performance is also evident in comparisons with BSS. However, TALD generally ranks as the best among the benchmark methods (except for one teacher–student pair). It is worth noting that both BSS and TALD have a higher level of complexity compared to CKD, as the former methods involve input perturbation using gradients (see Section 2). To ensure the accuracy and reproducibility of the results, we re-run the implementation provided by the authors in [22] to produce the results in Table 4. By utilizing their implementation,[2] we confidently validated and directly compared these methods with the performance reported in their results, ensuring a reliable evaluation.

Set 2: In this set of experiments, we compare the *coded* teacher with TALD when both are applied on top of various knowledge transfer

[2] The author-supplied TALD code for CIFAR100 in https://github.com/PotatoThanh/Adversarial-local-distribution-regularization-for-knowledge-distillation. For ImageNet, we attempted to reproduce their results, but encountered significant computational complexity, as explicitly noted by the authors. In addition, their experimental setup also deviates from the standards in the literature.



(a) VGG13→ MobileNetV2

(b) ResNet50→ MobileNetV2

(c) WRN40-2 → WRN16-2

(d) WRN40-2 → WRN40-1

(e) Resnet32×4 → Resnet8×4

Fig. 4. Comparison between the *coded* teacher and TALD when they are applied on top of different underlying knowledge transfer methods. The teacher–student pairs are reported in the caption of each figure.

methods including FT, FitNet, CC, SP, and RKD (see Fig. 4). The cost function of these methods can be represented as $\mathcal{L} = \mathcal{H} + \beta\mathcal{L}_{\text{distill}}$,[3] where $\mathcal{L}_{\text{distill}}$ denotes the loss function associated with a specific form of knowledge distilled into the student model. When applying the *coded* teacher on top of these methods, we tune β accordingly. From Figs. 4(a) and 4(b), it is clear that the coded teacher delivers significant gains. Specifically, with VGG13 as the teacher model, the *coded* teacher shows notable gains of 5% and 2.12% when compared to TALD for FT and FitNet, respectively. Similarly, when ResNet50 is used as the teacher model, the *coded* teacher achieves gains of 4% and 1.3% compared to TALD for FT and FitNet, respectively. These results demonstrate the consistent improvement achieved by the *coded* teacher.

Set 3: In the third set of experiments, we want to show whether the *coded* teacher, when applied on top of existing knowledge transfer methods, can improve their accuracy performance of the latter. Particularly, we apply the *coded* teacher on top of the following knowledge transfer methods in the literature: FitNet [9], FSP [11], FT [12], CC [17], SP [14], AB [13], RKD [16]. In our experiments, FitNet and FT were re-implemented based on the original papers; other methods were implemented, using either author-supplied or author-verified source codes. For a fair comparison, similarly to [7,15,22,29–32], we use the same pre-trained teacher models for all the benchmarks as those used in [15].[4]

Tables 5 and 6 illustrate the resulting accuracy results for student–teacher pairs with the same architectural styles and for student–teacher pairs with different architectural styles, respectively. From Tables 5 and 6, it is clear that the coded teacher can improve the accuracy performance of the underlying distillation method in most cases. In particular, when the student–teacher pair has different architectural styles, the accuracy gain can be above 5%.

5.3. Fine-grained classification task

- **Datasets:** This task involves evaluating datasets to distinguish visually similar objects in the same category. The first dataset, CUB200 (Caltech-UCSD Birds-200-2011), has 11,788 images of 200 bird subcategories divided into 5994 images for training, and each subcategory in the training set contains roughly 30 images. The second dataset, Stanford Dogs, has 20,580 images of 120 species, with 12K images for training, each class having 180 images, and the rest for validation.

[3] In the TALD framework, KD loss is incorporated for all other distillation methods, plus the TALD loss, to enhance their performance. However, CKD operates directly with other distillation methods without the addition of KD loss.

[4] We used the implementation of other methods from https://github.com/HobbitLong/RepDistiller.

Table 5
Test accuracy (%) of students on CIFAR100 dataset before and after the *coded* teacher is applied on top of the underlying distillation methods when the teacher and student pair has the same architectural style, where each underlying distillation method, when coupled with the coded teacher, is indicated by the bold prefix **C** and referred to as the corresponding coded distillation method. The accuracy for the underlying distillation methods is reported from [15,29], and the pre-trained teacher models are also used in [7,15,22,29–32].

Teacher	WRN-40-2	WRN-40-2	Resnet56	Resnet110	Resnet110	Resnet32 × 4	VGG13
Student	WRN-16-2	WRN-40-1	Resnet20	Resnet20	Resnet32	Resnet8 × 4	VGG8
Teacher	75.61%	75.61%	72.34%	74.31%	74.31%	79.42%	74.64%
Student	73.26%	71.98%	69.06%	69.06%	71.14%	72.50%	70.36%
CC	73.56%	72.21%	69.63%	69.48%	71.48%	72.97%	70.71%
CCC	**73.91**%	**72.32**%	**69.78**%	**70.21**%	**72.13**%	**73.16**%	**71.31**%
FitNet	73.58%	72.24%	69.21%	68.99%	71.06%	73.50%	71.02%
CFitNet	**74**%	**72.83**%	**69.49**%	**69.50**%	**71.66**%	**73.91**%	**71.14**%
FT	73.25%	71.59%	69.84%	70.22%	**72.37**%	72.86%	70.58%
CFT	**74.04**%	**72.31**%	**69.90**%	**70.25**%	71.87%	**73.15**%	**71.07**%
AB	**72.50**%	72.38%	69.47%	69.53%	70.98%	73.17%	70.94%
CAB	71.8%	**73.03**%	**70.11**%	**69.97**%	**72.11**%	**73.33**%	**71.35**%
SP	73.83%	72.43%	69.67%	70.04%	**72.69**%	72.94%	**72.68**%
CSP	**74.42**%	**72.87**%	**69.88**%	**70.10**%	71.98%	**73.17**%	71.85%
RKD	73.35%	**72.22**%	69.61%	69.25%	**71.82**%	71.90%	**71.48**%
CRKD	**73.64**%	72.19%	**69.85**%	**69.69**%	71.76%	**72.95**%	71.07%
FSP	72.91%	n/a	69.95%	70.11%	71.89%	72.62%	**70.23**%
CFSP	**73.13**%	n/a	**70.33**%	**70.37**%	**72.39**%	**73.08**%	69.10%

Table 6
Test accuracy (%) of students on CIFAR100 dataset before and after the coded teacher is applied to the underlying distillation methods when the teacher and student pair has different architectural styles, where each underlying distillation method, when coupled with the coded teacher, is indicated by the bold prefix **C** and referred to as the corresponding coded distillation method. The accuracy for the underlying distillation methods is obtained from [15,29], while the pre-trained teacher models are also used in [7,15,29–32].

Teacher	VGG13	ResNet50	ResNet50	Resnet32 × 4	Resnet32 × 4	WRN-40-2
Student	MobileNetV2	MobileNetV2	VGG8	ShuffleNetV1	ShuffleNetV2	ShuffleNetV1
Teacher	74.64%	79.34%	79.34%	79.42%	79.42%	75.61%
Student	64.6%	64.6%	70.36%	70.5%	71.82%	70.5%
CC	64.86%	65.43%	70.25%	71.14%	71.29%	71.38%
CCC	**65.83**%	**66**%	**71.04**%	**72.02**%	**73.40**%	**72.02**%
FitNet	64.14%	63.16%	70.69%	73.59%	73.54%	**73.73**%
CFitNet	**65.79**%	**66.08**%	**71.07**%	**73.62**%	**74.48**%	73.46%
FT	61.78%	60.99%	70.29%	71.75%	72.50%	72.03%
CFT	**66.39**%	**66.34**%	**71.09**%	**72.48**%	**73.73**%	**72.21**%
AB	66.06%	**67.20**%	70.65%	73.55%	74.31%	73.34%
CAB	**66.58**%	66.93%	**71.02**%	**74.58**%	**75.19**%	**74.50**%
RKD	64.52%	64.43%	**71.50**%	72.28%	**73.21**%	72.21%
CRKD	**65.88**%	**65.42**%	71.09%	72.28%	73.06%	**72.89**%

- **Teacher and student models:** We use wide residual networks (WRN) with different depths and fixed widths. We utilize the same student–teacher pairs for both datasets. For our teachers, we utilize pre-trained WRN-64 and WRN-28 models, which achieved validation accuracies of 38.57% and 42.32% on the CUB200 dataset, and 46.60% and 54.37% on the Stanford Dog dataset, respectively.
- **Training setup:** We use $\delta = 5$ yielding $|\mathcal{J}_I| = 22$. In our experiments, we use WRN-<d>-<w> as our baseline focusing on increasing the depth <d> and fixing the width <w> to 1, with the dropout probability of 0.3. The selected student models are WRN-10 (W10) and WRN-16 (W16), with 85K and 182K parameters, respectively. Following the settings in [43,44], we trained the networks from scratch for 200 epochs with a batch size of 32 using SGD with a momentum of 0.9 and weight decay equal to 1×10^{-4} with an initial learning rate γ of 0.1 that is divided by 10 at epochs 100 and 150. The training of both teacher and student models used the same training setup and the standard Inception-style preprocessing for augmentation settings [24]. The same values $\tau = 1$ and $\alpha = 0.5$ are used for both the KD and CKD.
- ***Coded* AT:** In addition to comparing CKD with KD, we also apply the coded teacher on the top of AT distillation method [10], yielding *coded* AT (CAT), and further compare CAT with AT. In this case, the β parameter in AT is set to 1000 as recommended in [10].

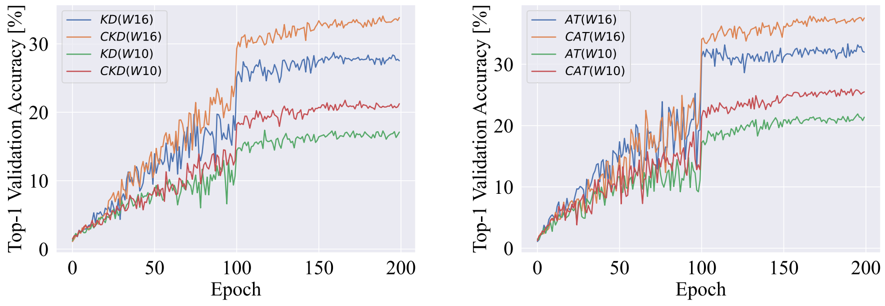
Table 7 shows the resulting accuracy results. From Table 7, it is clear that CKD and CAT improve KD and AT by a significant margin, respectively. To shed some light on the respective training processes, Fig. 5 further illustrates the respective Top-1 validation accuracy curves along the number of training epochs in the case of CUB200 dataset. As shown in Fig. 5, the CKD significantly outperforms the underlying distillation method around the 100-th epoch and stays that way afterwards in all tested cases.

6. Conclusion

In this paper, we have proposed a novel KD framework referred to as *coded* KD (CKD). In CKD, we insert an additional adaptive encoding layer in front of the teacher, yielding a *coded* teacher. The purpose of the *coded* teacher is to generate less confident output responses to be passed to the student so as to help the student generalize better. A specific JPEG adaptive encoding layer has been presented. The effectiveness and accuracy performance advantage of CKD have been demonstrated via comprehensive experimental results when it is compared with the conventional KD method and when the coded teacher is applied on top of diverse underlying knowledge transfer methods.

Table 7
The performance of CKD and CAT against conventional KD and AT.

Dataset		CUB200		Stanford Dog	
Student / Teacher		WRN-10 15.24%	WRN-16 27.63%	WRN-10 23.73%	WRN-16 34.79%
WRN-28	KD	17.68% (±1.21)	29.10% (±0.76)	25.79% (±0.88)	38.13% (±1.35)
	CKD	**21.27%(±0.52)**	**31.84%(±0.40)**	**28.32%(±0.70)**	**41.17%(±0.74)**
	AT	22.20% (±1.15)	32.94% (±0.65)	26.97% (±0.47)	39.04% (±0.57)
	CAT	**24.82%(±0.47)**	**37.10%(±0.10)**	**29.54%(±0.46)**	**41.84%(±0.33)**
WRN-64	KD	18.04% (±0.60)	29.9% (±1.04)	26.76% (±1.16)	38.30% (±1.63)
	CKD	**21.41% (±0.44)**	**33.1% (±0.89)**	**29.69% (±0.99)**	**41.66% (±0.38)**
	AT	22.55% (±0.82)	33.95% (±0.71)	27.14% (±1.36)	39.39% (±1.03)
	CAT	**24.94% (±0.92)**	**37.31% (±0.77)**	**29.73% (±0.41)**	**41.93% (±0.65)**



(a) KD for WRN-64

(b) AT for WRN-64

Fig. 5. Validation accuracy curves along the number of epochs over CUB200 dataset with (WRN-64) as the teacher and (W10) and (W16) as students: (a) CKD vs KD; and (b) CAT vs AT.

Two areas for improvement in CKD are: (1) its time complexity; and (2) the sub-optimality of the proposed JPEG adaptive encoding. Compared to the conventional KD, CKD has a slight increase in time complexity due to using adaptive image encoding. Nonetheless, this time complexity increase is significantly lower than that incurred by other knowledge transfer methods using input perturbation, such as BSS and TALD. In our future work, we will address the sub-optimality of adaptive JPEG encoding by exploring more advanced and generalized encoding methods to further improve the accuracy performance of CKD.

CRediT authorship contribution statement

Ahmed H. Salamah: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Shayan Mohajer Hamidi:** Writing – review & editing. **En-Hui Yang:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We are using a well-known dataset that has already been published online and is well-known in the literature.

Acknowledgement

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN203035-22, and in part by the Canada Research Chairs Program.

Table A.8
Covariance ($\times 10^{-5}$) for both *coded* and original teacher on the CIFAR-100.

Teacher	*Coded* teacher	Original teacher
Resnet56	13.0308	40.7161
Resnet32 × 4	23.5506	29.1309
VGG13	3.7323	58.7183
ResNet50	16.2534	61.1902

Appendix A. Analyzing student's generalization error in CKD

In this section, we aim to demonstrate the reason why CKD reduces the generalization error of the student model. To this end, we will first establish a clear definition of generalization error.

In a classification task with C classes, a DNN could be regarded as a mapping $f_\theta : I \to p_I$, where θ represents all the model parameters, $I \in \mathbb{R}^d$ is an input image, and p_I is a probability vector. One may learn such a classifier by minimizing the **true** risk $R(f_\theta)$ defined as follows:

$$R(f_\theta) \triangleq \mathbb{E}_{(I,\hat{y})}\left[\ell\left(\hat{y}, p_I\right)\right], \tag{A.1}$$

where $\ell(\cdot)$ is the loss function and $\ell(\cdot) \triangleq [\ell(1,\cdot), \ldots, \ell(C,\cdot)]$ is the vector of loss function.

On the other hand, to understand how KD works, let us write the loss function for the student in KD. To this end, denote by p_I^t and p_I^s the pre-trained teacher's and student's outputs to sample I, respectively. Then, the student uses the training dataset $\mathcal{D} \triangleq \{(I_n, \hat{y}_n)\}_{n=1}^N$ sampled from the joint distribution $P_{(I,\hat{y})}$ in order to minimize the following distillation risk function:

$$R_{\mathrm{S}}(f_\theta, \mathcal{D}) \triangleq \frac{1}{N}\sum_{n\in[N]}\left(p_{I_n}^t\right)^T \cdot \log\left(p_{I_n}^s\right), \tag{A.2}$$

Then, the generalization error for the student model is defined as follows [5]:

$$\mathbb{E}\left\{\left(R_{\mathrm{S}}(f_\theta, \mathcal{D}) - R(f_\theta)\right)^2\right\}. \tag{A.3}$$

Table B.9
CKD performance on ImageNet-1K subset with and without data augmentation.

Training Setup			Without Augmentation		With Augmentation	
Student / Teacher			Resnet18	MobileNet-V2	Resnet18	MobileNet-V2
Resnet34	KD	T1:	40.28%	49.27%	47.68% (±0.25)	52.65% (±0.44)
		T5:	67.07%	75.52%	73.81% (±0.28)	78.01% (±0.35)
	CKD	T1:	**41.27%**	**49.87%**	**49.74%(±0.09)**	**54.39%(±0.18)**
		T5:	**67.86%**	**76.21%**	**75.56%(±0.17)**	**79.57%(±0.07)**
Resnet50	KD	T1:	44.51%	51.43%	49.37% (±0.37)	53.93% **(±0.16)**
		T5:	71.45%	77.46%	75.42% (±0.24)	79.21% **(±0.14)**
	CKD	T1:	**46.78%**	**52.78%**	**51.82%(±0.21)**	**55.11%**(±0.28)
		T5:	**73.17%**	**78.46%**	**77.08%(±0.06)**	**80.31**%(±0.21)

Table B.10
Ablation study on the adaptive layer in CKD on ImageNet dataset subset.

Student / Teacher			Resnet18	MobileNet-V2
Resnet34	KD+QF50	T1:	48.11%	53.57%
		T5:	74.24%	78.91%
	KD+R(QF)	T1:	48.64%	53.48%
		T5:	74.68%	78.84%
Resnet50	KD+QF50	T1:	49.86%	54.53%
		T5:	75.61%	79.89%
	KD+R(QF)	T1:	49.89%	54.13%
		T5:	75.76%	79.42%

The smaller this value, the lower the generalization error is.

In order to provide justification about why the student model trained by CKD generalizes better than that trained by KD, we use the theoretical results derived in [45]. Particularly, the authors in [45] showed that the generalization error for the student is lower when the teacher's output vectors exhibit a lower covariance.

Hence, by conducting some experiments, we show that the outputs of the *coded* teacher exhibit a lower covariance than those of a conventional teacher. Specifically, we use exactly the same method as that used in [45] to estimate the covariance of the teacher's output probability. The results for four teacher models trained on CIFAR-100 dataset are listed in Table A.8 which suggest the output probability vectors of a *coded* teacher have lower covariance than those of an original teacher.

Appendix B. ImageNet subset dataset

- **Dataset:** We use a subset of the original ImageNet ILSVRC 2012 dataset [34], namely ImageNet-1K subset, which has also been used in the literature [46,47]. This subset shares the same validation set as the original ImageNet dataset. The training subset[5] is subsampled in a label-balanced fashion to result in a 1% configuration used in our experiments.
- **Training setup:** In the KD loss function in Eq. (1), α was set to 0.5. To determine the optimal value of τ for conventional KD, we conducted experiments for $\tau \in \{1,2,3\}$, and selected the one that yields the best performance. Specifically, we use $\tau = 2$ and $\tau = 3$ for Resnet34 and Resnet50, respectively. Then, the same values for α and τ are used in the CKD framework. This step was essential in ensuring the consistency of the source of gain between the conventional KD and CKD frameworks.
- **Results and analysis:** The results are reported in Table B.9, encompassing two distinct columns outlining CKD with and without input augmentation, which can also serve as an ablation study. It is worth noting that the authors in [48] observed that various knowledge distillation approaches, such as vanilla KD and CRD [1,15], can also achieve a modest improvement in the student model's performance solely through applying data augmentation. As such, the objective of conducting CKD with and without data augmentation was to demonstrate its efficacy in both scenarios.

As the first main column suggests (corresponding to without augmentation), CKD outperforms KD in both Top-1/Top-5 (T1/T5) validation accuracy. In the second column (with augmentation), we followed the best practice in the literature by performing three random runs with the default augmentation settings. The mean and standard deviation for T1/T5 validation accuracy is shown in Table B.9. The Top-1 validation accuracy indicates 2.18% and 2.32% improvement over the standard KD when Resnet34 and Resnet50 are the teacher models, respectively.
- **Ablation study on the adaptive layer in CKD:** To elucidate the necessity of the adaptive layer, we remove this layer and perform the following two sets of experiments: (1) using a constant QF=50 for all images, and (2) using a random QF for each image ranging from 50 to 100. The results for these experiments are listed in Table B.10. Compared to conventional KD and CKD in Table B.9, these results suggest the necessity of using an adaptive layer method in all setups.

References

[1] G. Hinton, O. Vinyals, J. Dean, et al., Distilling the knowledge in a neural network, 2, (7) 2015, arXiv preprint arXiv:1503.02531.

[2] H. Amer, A.H. Salamah, A. Sajedi, E.-H. Yang, High performance convolution using sparsity and patterns for inference in deep convolutional neural networks, 2021, arXiv preprint arXiv:2104.08314.

[3] M. Phuong, C. Lampert, Towards understanding knowledge distillation, in: International Conference on Machine Learning, PMLR, 2019, pp. 5142–5151.

[4] L. Ye, S.M. Hamidi, R. Tan, E.-H. Yang, Bayes conditional distribution estimation for knowledge distillation based on conditional mutual information, in: The Twelfth International Conference on Learning Representations, 2024, URL https://openreview.net/forum?id=yV6wwEbtkR.

[5] A.K. Menon, A.S. Rawat, S. Reddi, S. Kim, S. Kumar, A statistical perspective on distillation, in: International Conference on Machine Learning, PMLR, 2021, pp. 7632–7642.

[6] B. Heo, M. Lee, S. Yun, J.Y. Choi, Knowledge distillation with adversarial samples supporting decision boundary, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01, 2019, pp. 3771–3778.

[7] K. Zheng, E.-H. Yang, Knowledge distillation based on transformed teacher matching, in: Proc. the Twelfth International Conference on Learning Representations, ICLR, 2024.

[5] For detailed information about the sampling methodology and specific subsets used in the experiments, please refer to https://www.tensorflow.org/datasets/catalog/imagenet2012_subset.

[8] S. Mohajer Hamidi, Training neural networks on remote edge devices for unseen class classification, IEEE Signal Process. Lett. 31 (2024) 1004–1008, http://dx.doi.org/10.1109/LSP.2024.3383948.

[9] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: Hints for thin deep nets, 2014, arXiv preprint arXiv:1412.6550.

[10] S. Zagoruyko, N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, in: Fifth International Conference on Learning Representations, 2017.

[11] J. Yim, D. Joo, J. Bae, J. Kim, A gift from knowledge distillation: Fast optimization, network minimization and transfer learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4133–4141.

[12] J. Kim, S. Park, N. Kwak, Paraphrasing complex network: Network compression via factor transfer, Adv. Neural Inf. Process. Syst. 31 (2018).

[13] B. Heo, M. Lee, S. Yun, J.Y. Choi, Knowledge transfer via distillation of activation boundaries formed by hidden neurons, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01, 2019, pp. 3779–3787.

[14] F. Tung, G. Mori, Similarity-preserving knowledge distillation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1365–1374.

[15] Y. Tian, D. Krishnan, P. Isola, Contrastive representation distillation, in: International Conference on Learning Representations, 2020.

[16] W. Park, D. Kim, Y. Lu, M. Cho, Relational knowledge distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3967–3976.

[17] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, Z. Zhang, Correlation congruence for knowledge distillation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5007–5016.

[18] D. Soudry, E. Hoffer, M.S. Nacson, S. Gunasekar, N. Srebro, The implicit bias of gradient descent on separable data, J. Mach. Learn. Res. 19 (2018).

[19] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in: International Conference on Machine Learning, PMLR, 2017.

[20] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[21] G.K. Wallace, The JPEG still picture compression standard, IEEE Trans. Consumer Electron. 38 (1) (1992) xviii–xxxiv.

[22] T. Nguyen-Duc, T. Le, H. Zhao, J. Cai, D. Phung, Adversarial local distribution regularization for knowledge distillation, in: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2023, pp. 4670–4679.

[23] H. Zhang, Z. Hu, W. Qin, M. Xu, M. Wang, Adversarial co-distillation learning for image recognition, Pattern Recognit. 111 (2021) 107659.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, (ISSN: 1063-6919) 2016, pp. 2818–2826.

[25] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, G. Hinton, Regularizing neural networks by penalizing confident output distributions, 2017, arXiv preprint arXiv:1701.06548.

[26] E.-H. Yang, H. Amer, Y. Jiang, Compression helps deep learning in image classification, Entropy (ISSN: 1099-4300) 23 (7) (2021).

[27] K. Zheng, A.H. Salamah, L. Ye, E.-H. Yang, JPEG compliant compression for DNN vision, in: 2023 IEEE International Conference on Image Processing, ICIP, 2023, pp. 1875–1879, http://dx.doi.org/10.1109/ICIP49359.2023.10221982.

[28] A.H. Salamah, K. Zheng, L. Ye, E.-H. Yang, JPEG compliant compression for DNN vision, IEEE J. Sel. Areas Inf. Theory (2024).

[29] L.C. Chuanguang Yang, Y. Xu, Hierarchical self-supervised augmented knowledge distillation, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI, 2021, pp. 1217–1223.

[30] P. Chen, S. Liu, H. Zhao, J. Jia, Distilling knowledge via knowledge review, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 5008–5017.

[31] T. Huang, S. You, F. Wang, C. Qian, C. Xu, Knowledge distillation from a stronger teacher, 2022, arXiv preprint arXiv:2205.10536.

[32] R. Miles, A. Lopez-Rodriguez, K. Mikolajczyk, Information theoretic representation distillation, in: BMVC, 2022.

[33] R. Müller, S. Kornblith, G.E. Hinton, When does label smoothing help? Adv. Neural Inf. Process. Syst. 32 (2019).

[34] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[35] A. Krizhevsky, G. Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Toronto, ON, Canada, 2009.

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019).

[37] S. Marcel, Y. Rodriguez, Torchvision the machine-vision package of torch, in: Proceedings of the 18th ACM International Conference on Multimedia, 2010, pp. 1485–1488.

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[39] X. Zhu, S. Gong, et al., Knowledge distillation by on-the-fly native ensemble, Adv. Neural Inf. Process. Syst. 31 (2018).

[40] S. Ahn, S.X. Hu, A. Damianou, N.D. Lawrence, Z. Dai, Variational information distillation for knowledge transfer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

[41] N. Passalis, M. Tzelepi, A. Tefas, Probabilistic knowledge transfer for lightweight deep representation learning, IEEE Trans. Neural Netw. Learn. Syst. 32 (5) (2020) 2030–2039.

[42] Z. Huang, N. Wang, Like what you like: Knowledge distill via neuron selectivity transfer, 2017, arXiv preprint arXiv:1707.01219.

[43] S. Yun, J. Park, K. Lee, J. Shin, Regularizing class-wise predictions via self-knowledge distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13876–13885.

[44] H. Lee, Y. Park, H. Seo, M. Kang, Self-knowledge distillation via dropout, Comput. Vis. Image Underst. (2023).

[45] H. Wang, S. Lohit, M.N. Jones, Y. Fu, What makes a" good" data augmentation in knowledge distillation-a statistical perspective, Adv. Neural Inf. Process. Syst. 35 (2022) 13456–13469.

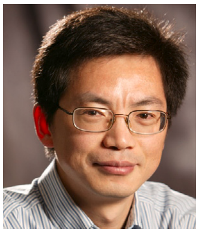
[46] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.

[47] K. Kotar, G. Ilharco, L. Schmidt, K. Ehsani, R. Mottaghi, Contrasting contrastive self-supervised representation learning pipelines, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 9949–9959.

[48] W. Li, S. Shao, W. Liu, Z. Qiu, Z. Zhu, W. Huan, What role does data augmentation play in knowledge distillation? in: Proceedings of the Asian Conference on Computer Vision, ACCV, 2022, pp. 2204–2220.

Ahmed H. Salamah obtained his B.Sc. and M.Sc. degrees in Electronics and Communications in 2014 and 2017, respectively, from AASTMT, Alexandria, Egypt. Currently, he is pursuing a Ph.D. in Electrical and Computer Engineering at the University of Waterloo. His research specialization lies in deep learning performance optimization, focusing on computation, classification, and image compression.

Shayan Mohajer Hamidi received the B.Sc. from Sharif University, Iran, in 2016, and the MASc. from the University of Waterloo, Waterloo, ON, Canada, in 2018, both in electrical engineering. He is now working toward his Ph.D. at the University of Waterloo with his interests in machine learning and optimization



En-Hui Yang received the B.Sc. in applied mathematics from Huaqiao University, China, and Ph.D. in electrical engineering from the University of Southern California, USA, in 1986, and 1996, respectively. Since June 1997, he has been a Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada.