In this paper, we introduce InDistill, a method that serves as a warmup stage for enhancing Knowledge Distillation (KD) effectiveness. InDistill focuses on transferring critical information flow paths from a heavyweight teacher to a lightweight student. This is achieved via a training scheme based on curriculum learning that considers the distillation difficulty of each layer and the critical learning periods when the information flow paths are established. This procedure can lead to a student model that is better prepared to learn from the teacher. To ensure the applicability of InDistill across a wide range of teacher-student pairs, we also incorporate a pruning operation when there is a discrepancy in the width of the teacher and student layers. This pruning operation reduces the width of the teacher’s intermediate layers to match those of the student, allowing direct distillation without the need for an encoding stage. The proposed method is extensively evaluated using various pairs of teacher-student architectures on CIFAR-10, CIFAR-100, and ImageNet datasets demonstrating that preserving the information flow paths consistently increases the performance of the baseline KD approaches on both classification and retrieval settings. The code is available at https://github.com/gsarridis/InDistill.

1. Introduction

The constant effort to increase the performance of Deep Neural Networks (DNN) has resulted in deeper architectures that require massive compute power and memory to train and deploy. Also, several application settings need to deploy DNNs on hardware with limited resources, which has increased the popularity of the research field of model compression. During the past years, model compression has attracted a lot of research interest and many pertinent approaches have been proposed such as parameter pruning, quantization, low-rank factorization, and Knowledge Distillation (KD) [8, 9].

The objective of KD, one of the most effective ways for model compression [8, 14], is to transfer knowledge from a powerful network to a smaller and faster one, to extend its performance capabilities. Early KD approaches aim at transferring the teacher’s logits [18]. This enables smaller models to understand how bigger models perceive the inputs, which enhances the learning of the former based on knowledge captured by the latter. However, the lack of intermediate layer supervision impedes the learning of how the information flows through the teacher’s layers reducing the student’s generalization potential [36]. To remedy this, recent KD methods [38, 48] distill knowledge from intermediate layers, as it is empirically shown to improve the KD effectiveness.
Pruning methods are applied to reduce storage requirements or inference time [9, 26, 28]. Unstructured pruning [46] removes unimportant weights reducing the model’s storage size, while structured pruning [7, 12, 25, 27, 43] removes the less important CNN filters reducing both the model’s storage size and processing time. Furthermore, prior work have explored combining KD with pruning for model compression. Some approaches apply pruning before KD [34], while others prune and subsequently apply KD only to the layers unaffected by pruning [2] or use KD to fine-tune pruned models [6]. However, none of the aforementioned methods follow the standard teacher-student paradigm; thus, they cannot be compared with typical KD approaches.

Furthermore, it is worth noting that during the training process, a neural network undergoes several phases [1]. More precisely, the first training epochs are considered responsible for the creation of the model’s information flow paths. These paths represent the routes through which information travels and is processed within the network, effectively shaping the network’s ability to learn. It is important to highlight that after their formation, these paths can only be fine-tuned through the rest of the training procedure. This critical fact has been taken into consideration in prior work [36] to effectively distill intermediate layers, without, however, taking into consideration the increasing distillation difficulty of successive intermediate layers, from shallower to deeper network parts.

This paper introduces InDistill, a method that addresses the aforementioned limitations and can be employed as a warmup stage prior to any KD approach (see Fig. 1). Inspired by curriculum learning principles [3] and the critical learning periods of a neural network [36], InDistill suggests a simple, yet effective way to distill the teacher’s information flow paths to the student, by directly transferring knowledge from each intermediate layer separately and in ascending transferring difficulty order (i.e., from shallow to deep layers). Additionally, to expand the applicability of InDistill, channel pruning is specifically applied to the teacher’s intermediate layers when a width disparity exists between the teacher and student architectures. This selective pruning aligns the architectures, facilitating direct feature map matching and thus preserving the crucial information flow paths. Similarly, in scenarios with an extremely high capacity gap between the teacher and student models, an auxiliary teacher is also considered to facilitate the application of InDistill.

The proposed method has been evaluated on a wide range of classification and retrieval settings with the use of widely adopted benchmarks (i.e., CIFAR-10 [24], CIFAR-100 [24], and ImageNet [10]) combined with various state-of-the-art KD approaches and is found to boost their performance in the vast majority of our conducted experiments. The main contributions of this work are:

- A method that introduces a curriculum learning-based training scheme considering the distillation difficulty of each layer and the critical learning periods crucial for effectively transferring the teacher’s information flow paths to the student, as a warmup step before employing any KD approach.
- A pruning mechanism applied in scenarios where there is a width disparity between teacher and student architectures. This mechanism aligns the architectures, facilitating direct knowledge transfer from feature maps and preserving the crucial information flow paths.
- A wide analysis involving 5 baseline KD methods and 3 widely-adopted benchmarks, demonstrating the effectiveness of InDistill in improving the performance of the baseline approaches.

2. Related work

Logits-based knowledge distillation. KD, initially proposed in [18], is an effective way for model compression by reproducing the teacher’s class probability distribution using a much faster and smaller student model. Several KD methods exploiting only the model’s output probability distributions have been proposed [11, 22, 30, 32, 33, 52]. A recent work [51] reformulates the classical KD loss into target class knowledge distillation and non-target class knowledge distillation, demonstrating the efficacy of logit distillation with this Decoupled Knowledge Distillation (DKD) methodology. The Multi-Level KD (MLKD) [19] approach enhances logit distillation through multilevel prediction alignment, by incorporating batch-level and class-level prediction alignments. Furthermore, logit standardization [40] prior to the distillation can address the mismatch in logit magnitudes between teacher and student models and thus improve the distillation effectiveness.

Features-based knowledge distillation. The first method that leverages feature maps of the teacher in order to provide the student with extra supervision is presented in Romero et al. [38]. This method transfers knowledge from only one intermediate layer with the aid of feature map encoding. Similarly, Probabilistic Knowledge Transfer (PKT) [35] distills knowledge from the penultimate layer’s feature maps by trying to match the teacher’s and student’s distributions of the samples’ pairwise similarities. However, transferring knowledge of one intermediate layer can not capture the critical connections between the layers. To alleviate this shortcoming, the Attention Transfer (AT) [48] proposes an attention mechanism that applies distillation to all intermediate layer representations. Similarly, the Hierarchical Self-supervised Augmented Knowledge Distillation (HSAKD) [45] employs classifiers on top of all intermediate layers to supervise the KD process. In a different perspective, ReviewKD [5] introduces cross-stage connection
3. Methodology

3.1. Problem formulation

The problem of transferring knowledge from a teacher to a student model is formulated as follows. Let \( X \in \mathbb{R}^{3 \times h \times w} \) denote an input image with \( h \) and \( w \) height and width, respectively. \( d(\cdot) \) is the teacher model, and \( l = 1, \ldots, L_d \) its layer indices. Then, \( \mathbf{P}^{(l)} = d(X, l) \in \mathbb{R}^{n_{d,l} \times h_{d,l} \times w_{d,l}} \) denotes the teacher’s \( l \)-th layer feature map, where \( n_{d,l}, h_{d,l}, \) and \( w_{d,l} \) is its number of channels and spatial dimensions. Accordingly, consider a student model \( g(\cdot) \) with \( \theta \) learnable parameters, \( L_g \) layers and \( \mathbf{S}^{(l)} = g(X, l) \in \mathbb{R}^{n_{g,l} \times h_{g,l} \times w_{g,l}} \) the \( l \)-th layer’s feature map. In addition, let \( q_l, q_s \in \mathbb{R}^C \) be the class probability distributions of the teacher and student model, respectively, with \( C \) the number of output classes. Then, the goal of a typical logit-based KD is to match the teacher’s and student’s class probability distributions (\( q_t \approx q_s \)). In contrast, Indistill aims to prepare the student model \( g(\cdot) \) for this process by learning \( \theta \) parameters that can enhance KD effectiveness.

3.2. Information flow preservation

First, let us consider the simplest scenario, where teacher and student architectures share the same width, and thus it is feasible to transfer knowledge from the corresponding feature maps directly. Specifically, the loss between \( l \)-th layer feature maps, \( \mathbf{P}^{(l)} \) and \( \mathbf{S}^{(l)} \) is defined as:

\[
\mathcal{L}_{MSE}^{(l)} = \|\mathbf{P}^{(l)} - \mathbf{S}^{(l)}\|_2^2, \tag{1}
\]

where \( \| \cdot \|_2 \) denotes the \( l_2 \)-norm. Note that Indistill can only be applied to intermediate layers.

Curriculum learning suggests dividing a hard task into sub-tasks w.r.t. their difficulty and performing training sequentially in increasing difficulty. When trying to transfer useful information from all the teacher’s layers, one could hypothesize that transferring knowledge from shallow layers is easier than deep layers, for two reasons. First, shallow layers hold in general low-level information (e.g., edges and corners), while deep layers hold task-specific high-level information. Second, deep layer distillation also requires transferring the knowledge obtained by all previous layers as well as the information flow paths until then. Figure 2 exemplifies the higher loss when transferring knowledge from deeper layers, validating this intuition empirically. Given the above, we propose a curriculum learning scheme that considers the distillation of each layer as a separate sub-task. Particularly, \( L_g \) sequential sub-tasks are spawned equal to the number of student layers. These sub-tasks determine the training schedule. If the total number of training epochs is \( E \), then the number of epochs corresponding to each sub-task (i.e., layer) can be calculated as:

\[
e_i = \begin{cases} 
\frac{a + ib}{E - \sum_{i=1}^{L_g} e_i} & i \neq L_g \smallskip 
E - \sum_{i=1}^{L_g} e_i & i = L_g, \quad i \in \{1, 2, \cdots, L_g\}.
\end{cases} \tag{2}
\]
where \( a \) is a threshold parameter indicating the minimum number of epochs for each layer’s training and \( b \) is a parameter that increments the number of epochs w.r.t. the sub-task’s difficulty. Also, the set of epochs that corresponds to each sub-task \( i \) is \( S_i = \{ r_i + 1, r_i + 2, \ldots, r_i + e_i \} \) where:

\[
r_i = \begin{cases} 
0 & i = 1 \\
\sum_{k=1}^{n_i} e_k & i > 1, i \in \{1, 2, \ldots, L_g\}.
\end{cases}
\]

Note that in each sub-task \( i \), the first \( i \) layers are trained. By adopting the proposed curriculum learning scheme, the first \( \sum_{i=1}^{L_g-1} e_i \) epochs are dedicated to the preservation of the teacher’s information flow paths. After this warmup stage, any existing KD loss can be employed, combined with a task-specific loss (e.g., the cross-entropy loss for the classification tasks). The final loss is calculated as:

\[
L = \begin{cases} 
L_{MSE}^{(i)} & i \neq L_g, i \in \{1, 2, \ldots, L_g\} \\
L_{KD} + L_{task} & i = L_g
\end{cases}
\]

This way, the student model can effectively form the critical connections that facilitate the following KD process.

### 3.3. Channel pruning

In more complex KD scenarios, where there is a width discrepancy between the teacher and student networks, channel pruning is employed to force architectural alignment and thus allow for direct KD on feature maps (i.e., \( L_{MSE} \)). The typical criterion for evaluating the importance of a filter is the \( l_1 \)-norm or \( l_2 \)-norm [16, 25, 44]. Here, we opt for the approach that applies structured channel pruning based on \( l_1 \)-norm [25]. Specifically, let \( K \in \mathbb{R}^{n_1 \times n_o \times k \times k} \) denote the kernel of an arbitrary layer, where \( k \) is the kernel size, and \( n_i \) and \( n_o \) the input and output channels, respectively. Then, we follow the pruning procedure described in Algorithm 1. Channel pruning is applied to all teacher’s intermediate layers to force architectural alignment. After applying Algorithm 1 with \( p = q \cdot n_d \cdot l \), the teacher’s pruned feature maps \( P^{(i)} \in \mathbb{R}^{n_g \times h_g \times w_g \times l} \) are of equal size to the student’s feature maps, which allows for direct knowledge transfer involving no encoding. Fig. 3 provides an overview of InDistill, including its pruning component.

### 3.4. Auxiliary teacher

In cases of teacher-student pairs with an extreme capacity gap, different number of layers \( L_d \neq L_g \), or structurally different architectures (e.g., a teacher with and a student without residual connections), we follow prior works [31, 36] that make use of an auxiliary model built from the teacher model using standard knowledge distil-
lation schemes. $f(\cdot)$ denotes the auxiliary model, $L_f$ the
number of layers (here $L_f=L_q$), and $A^{(l)} = f(X, l) \in \mathbb{R}^{n_{f,l} \times h_{f,l} \times w_{f,l}}$ its output feature maps. Note that $h_{f,l}=h_{g,l}$
and $w_{f,l}=w_{g,l}$ as the networks share kernel sizes. The auxiliary
model is designed to have the same number of layers and the
same number of output channels as the student, after applying pruning with pruning rate $q$ to all intermediate
layers. As a result, the auxiliary model’s feature map sizes $n_{f,l}$ depend on the student model’s feature maps sizes $n_{g,l}$ and the pruning rate $q \in [0, 1)$, namely $n_{f,l} = \frac{n_{g,l}}{1-q}$. Having
designed the auxiliary model, standard logit-based KD [18]
is applied to transfer knowledge from the teacher to the aux-
iliary model.

4. Experimental setup
4.1. Datasets and model architectures
The proposed method has been evaluated on three well-
known datasets: CIFAR-10 [24], CIFAR-100 [24], and Im-
ageNet [10]. Regarding the model architectures, there are
three types of setups based on the capacity gap and archi-
tectural differences between teachers and students:

Small capacity gap (same width). Following prior work
[19, 42, 51], we evaluate the proposed method using seven
ResNet [17] teacher-student pairs widely used for
evaluating KD methodologies (e.g., ResNet34-ResNet18,
ResNet56-ResNet20, ResNet110-ResNet32, etc.). These
pairs have the same width, meaning the feature maps de-
derived from their blocks have matching dimensions.

Large capacity gap (different width). We em-
ploy the following teacher-student pairs: ResNet50-
ResNet18, ResNet32x4-ResNet32, ResNet32x2-ResNet20,
and ResNet32x4-ResNet20. The teachers of these pairs
have either 2 or 4 times the width of the corresponding stu-
dents. In this scenario, channel pruning is applied to the
teacher’s intermediate layers to match the student’s width,
facilitating direct KD via feature-based distillation while
preserving the crucial information flow paths.

Extreme capacity gap (different architecture). Following prior work [36], we consider ResNet18 as the
teacher model (consisting of around 11 million trainable
parameters) and a tiny CNN consisting of 3 convolutional
layers as the student model (CNN-S). Due to the structural
differences as well as the large capacity gap between teacher
and student, we additionally use an auxiliary teacher model
(CNN-A). Table 1 presents the model architectures in detail.
For CNN-A and CNN-S, batch normalization is applied af-
after each convolutional layer. The feature maps are flattened
before propagation to the fully-connected layers.

4.2. Implementation details and evaluation protocol
For the CIFAR-10 dataset, the models are trained using
the Adam [23] optimizer for 60 epochs with a learning rate
of 0.001 and for 10 epochs with a learning rate of 0.0001.
The batch size is set to 128 and the parameters $a$ and $b$ are
set to 2 and 1, respectively. For CIFAR-100, the models
are trained for 240 epochs using the SGD [41] optimizer
with 0.9 momentum and 0.05 learning rate for the first 150
epochs, then the learning rate decays every 30 epochs by a
factor of 10. The batch size is equal to 64 and the weight
decay is 0.0005. For ImageNet, the models are trained for
100 epochs using the SGD optimizer with 0.9 momentum
and 0.1 learning rate, while the learning rate decays every
30 epochs by a factor of 10. The batch size is equal to 256,
and the weight decay is 0.0001. The parameters $a$ and $b$, are
set to 5 and 1, respectively.

The official training/test splits are utilized for evaluation.
For this evaluation, we report accuracy (top-1 and top-5) for
classification tasks, and mean Average Precision (mAP)
and Precision@k (P@k) for retrieval tasks. In the context of
retrieval evaluation, the training sets are employed to cre-
ate the databases. Subsequently, the test sets are used to
query these databases, allowing us to measure the retrieval
performance of representations. Finally, we also evaluate
the methods w.r.t. information flow preservation. The
information flow of a network can be modeled through Mu-
tual Information (MI) [36]. MI divergence loss proposed
in PKT [35] aims at minimizing the MI divergence between
teacher and student, thus forcing the student model to mimic
the teacher’s information flow paths. Hence, we consider it
as a proper measure of information flow preservation and
denote it as $L_{MI}$. The experiments were conducted using
NVIDIA RTX-3060 and RTX-4090 GPUs.

4.3. Baseline methods
To evaluate InDistill’s effectiveness in enhancing the per-
formance of baseline KD approaches, we consider 5 state-
of-the-art methodologies: the original KD method (KD)
[18], the Probabilistic Knowledge Transfer (PKT) [35], the
Contrastive Representation Distillation (CRD) [42], the
Multilevel Knowledge Distillation (MLKD) [19], and the
Decoupled Knowledge Distillation (DKD) [51]. Addition-
ally, for comparison with the proposed scheme, we include
in our evaluation two methods that claim to preserve the
teacher’s information flow: the Flow of Solution Procedure
(FSP) [47] and the PKT-H-CR [36]. More details regarding
the aforementioned methods can be found in Sec. 2.

| model       | conv (#filters, #kernel) | fc (#neurons) |
|-------------|--------------------------|---------------|
| CNN-A       | (16, 3×3)                | 128           |
| CNN-S       | (8, 3×3)                 | 64            |

Table 1. Auxiliary teacher and student model architectures. These network architectures are used for extreme capacity gap experi-
ments on CIFAR-10. conv and fc stands for convolutional and
fully-connected layer, respectively.
### 5. Results

#### 5.1. Evaluation on standard KD benchmarks

First, we evaluate InDistill on the CIFAR-100 dataset, employing the standard teacher-student pairs used for benchmarking of KD methodologies. The results are summarized in Tab. 2. For each teacher-student pair, we report the performance of the baselines compared with the performance of the baselines combined with the proposed InDistill warmup. InDistill consistently improves the performance across most methods, demonstrating significant gains. Notably, KD+InDistill shows a remarkable improvement in all teacher-student settings, with the highest gain of +2.40% when WRN-32x4 is the teacher and WRN-8x4 is the student. Similarly, for CRD, the gain achieved by adding InDistill is notable, especially with a +0.63% improvement for ResNet56 to ResNet20.

**Table 2. Evaluation on small teacher-student capacity gap.** Top-1 accuracy on CIFAR-100 of seven teacher-student homogeneous pairs trained with five KD methods with and without InDistill warmup. Performance of teachers and students without distillation is provided for reference. Numbers in parenthesis indicate performance gain or loss from the use of InDistill.

| method          | top-1 acc | top-5 acc |
|-----------------|-----------|-----------|
| Teacher         | 75.61     | 72.34     |
| Student         | 71.98     | 69.06     |
| KD [18]         | 73.54     | 70.66     |
| KD+InDistill    | 75.09 (+1.55) | 72.16 (+1.50) |
| PKT [35]        | 74.35     | 70.34     |
| PKT+InDistill   | 73.28 (-0.17) | 70.36 (+0.02) |
| CRD [42]        | 74.74     | 71.79     |
| CRD+InDistill   | 74.22 (+0.08) | 71.95 (-0.02) |
| DKD [51]        | 74.81     | 71.97     |
| DKD+InDistill   | 73.39 (+0.58) | 71.95 (+0.63) |
| MLKD [19]       | 75.35     | 72.19     |
| MLKD+InDistill  | 75.71 (+0.36) | 72.64 (+0.45) |

#### 5.2. Evaluation on large teacher-student capacity gap

**Table 3. Large-scale evaluation on small capacity gap.** Top-1 and top-5 accuracy on ImageNet of five KD methods both with and without InDistill warmup. ResNet34 and ResNet18 are used as teacher and student networks.

| method          | top-1 acc | top-5 acc |
|-----------------|-----------|-----------|
| Teacher         | 48.21     | 30.91     |
| Student         | 24.79     | 24.67     |
| KD [18]         | 56.59     | 53.20     |
| KD+InDistill    | 59.79 (+3.20) | 67.45 (+2.37) |
| PKT [35]        | 24.67     | 25.75 (+1.08) |
| PKT+InDistill   | 25.75 (+1.08) | 27.46 (+1.05) |
| CRD [42]        | 25.99     | 31.10 (+5.11) |
| CRD+InDistill   | 31.10 (+5.11) | 32.19 (+4.73) |

**Table 4. Evaluation on large teacher-student capacity gap.** Classification and retrieval performance evaluation on ImageNet and CIFAR-100. ResNet50 and ResNet18 are used as teacher and student for the ImageNet dataset. ResNet32\times4 and ResNet32 are used as teacher and student for the CIFAR-100 dataset.

| method          | ImageNet | CIFAR-100 |
|-----------------|----------|-----------|
| Teacher         | 77.94    | 56.59     |
| Student         | 43.42    | 59.79     |
| KD [18]         | 30.91    | 53.20     |
| KD+InDistill    | 31.52 (+0.61) | 67.45 (+2.37) |
| PKT [35]        | 24.67    | 53.20     |
| PKT+InDistill   | 25.75 (+1.08) | 67.45 (+2.37) |
| CRD [42]        | 25.99    | 31.10 (+5.11) |
| CRD+InDistill   | 31.10 (+5.11) | 32.19 (+4.73) |

Table 3 presents the results on the ImageNet dataset. The performance metrics include top-1 and top-5 accuracy for different methods, with and without the InDistill approach. The addition of InDistill yields improvements across all methods. For instance, KD+InDistill shows an increase of +0.31% in top-1 accuracy and +0.37% in top-5 accuracy. Similarly, for the top-performing method MLKD, MLKD+InDistill achieves gains of +0.13% in top-1 accuracy and +0.22% in top-5 accuracy, demonstrating the robustness and efficacy of the InDistill approach.
Table 5. Evaluation with increasing capacity gap. Top-1 accuracy on CIFAR-100 for three network pairs. Std stands for the standard deviation of the top-1 accuracy over the three runs.

| Teacher arch. | Student arch. | ResNet32 | ResNet32x2 | ResNet32x4 |
|---------------|---------------|----------|------------|------------|
| Teacher       | 71.75         | 75.97    | 79.42      | -          |
| Student       | 69.06         | 69.06    | 69.06      | -          |
| KD [18]       | 70.86         | 69.84    | 68.08      | 1.41       |
| KD+InDistill  | 71.99 (+1.13) | 71.65 (+1.81) | 70.96 (+2.88) | 0.52 (-0.89) |
| PKT [35]      | 71.12         | 69.94    | 68.66      | 1.23       |
| PKT+InDistill | 71.80 (+0.68) | 71.92 (+1.98) | 71.26 (+2.60) | 0.35 (-0.88) |
| CRD [42]      | 71.41         | 70.62    | 69.72      | 0.85       |
| CRD+InDistill | 71.91 (+0.50) | 72.24 (+1.62) | 71.14 (+1.42) | 0.56 (-0.29) |
| DKD [51]      | 71.16         | 70.22    | 68.47      | 1.37       |
| DKD+InDistill | 71.94 (+0.78) | 71.95 (+1.73) | 71.11 (+2.64) | 0.48 (-0.89) |
| MLKD [19]     | 71.60         | 71.99    | 70.54      | 0.75       |
| MLKD+InDistill| 71.97 (+0.37) | 72.48 (+0.49) | 71.22 (+0.68) | 0.63 (+0.12) |

Table 6. Evaluation on extreme teacher-student capacity gap. Classification and retrieval performance evaluation on CIFAR-10. ResNet-18, CNN-A, and CNN-S are used as teacher, auxiliary, and student networks, respectively.

| Method        | mAP  | P@100 | top-1 acc |
|---------------|------|-------|----------|
| Teacher       | 90.47| 92.26 | 95.23    |
| Auxiliary     | 66.78| 75.91 | 82.60    |
| Student       | 39.00| 58.77 | 69.06    |
| FSP [47]      | 37.66| 57.41 | 71.14    |
| PKT-H-CR [36] | 53.00| 64.06 | 73.33    |
| KD [18]       | 39.38| 58.77 | 73.08    |
| KD+InDistill  | 42.82 (+3.44) | 61.09 (+2.32) | 75.17 (+2.09) |
| PKT [35]      | 51.41| 62.56 | 73.27    |
| PKT+InDistill | 54.61 (+3.20) | 65.40 (+2.84) | 74.13 (+0.86) |
| CRD [42]      | 47.11| 61.64 | 71.99    |
| CRD+InDistill | 49.57 (+2.46) | 62.23 (+0.59) | 73.32 (+1.33) |
| DKD [51]      | 42.82| 61.11 | 74.45    |
| DKD+InDistill | 44.08 (+1.26) | 62.18 (+1.07) | 74.84 (+0.39) |
| MLKD [19]     | 43.24| 61.54 | 73.76    |
| MLKD+InDistill| 45.40 (+2.16) | 63.15 (+1.61) | 74.87 (+1.11) |

5.2. Evaluation on large capacity gap settings

Apart from the standard KD evaluation setups, we consider an additional teacher-student pair for CIFAR-100 and ImageNet datasets – specifically, ResNet50-ResNet18 and ResNet32x4-ResNet32, respectively. In that way, we assess the contribution of pruning in our method while the capacity (and the width) of the teacher increases. The corresponding results are presented in Tab. 4, concerning both classification and retrieval tasks. In terms of accuracy, the improvements are similar to the ones in small capacity gap experiments with gains ranging from +0.09% to +0.87%. Notably, for the retrieval task, we observe significantly higher improvements in terms of mAP and P@100\(^1\), such as +7.06% mAP and +4.79% P@100 for CRD+InDistill on CIFAR100. These gains can be attributed to the preservation of information flow paths, enabling better feature learning.

While the capacity gap between teacher and student increases, such as in the model pairs ResNet32-ResNet20, ResNet32x2-ResNet20, and ResNet32x4-ResNet20, traditional KD methods often suffer from diminished performance, as evidenced by Tab. 5. In particular, the baseline approaches generally result in decreased top-1 accuracy across most model pairs when width discrepancy increases. Integrating InDistill before applying KD significantly mitigates this issue. Specifically, it enhances the student’s performance notably in the ResNet32x2-ResNet20 pair compared to the wider ResNet32-ResNet20 pair in all experiments, except for KD+InDistill, and demonstrates less accuracy drop from ResNet32x2-ResNet20 to ResNet32x4-ResNet20 pairs compared to baseline approaches. This trend is reflected in the standard deviation (std) values presented in Tab. 5, where InDistill consistently demonstrates reduced std compared to the baseline approaches (ranging from -0.12 to -0.89). Furthermore, we investigate the effect of pruning all layers of the teacher model rather than only the specific layer being transferred. For the ResNet32x4-ResNet20 pair using KD as the primary knowledge distillation approach, we observe a 0.55% accuracy drop (resulting in 70.41% accuracy) compared to InDistill. This reduction occurs because pruning all layers drops the teacher model’s overall capacity. Additionally, apart from width discrepancies, it is important to note that when spatial dimensions differ between teacher and student, pooling can be used to adjust them accordingly.

5.3. Evaluation on extreme capacity gap settings

Finally, we evaluate the performance of the proposed method in extreme scenarios where there is a need to employ a very lightweight model (e.g., simple CNNs with only a few thousand parameters, as in Tab. 1). In such cases, an auxiliary teacher is employed to allow for applying InDistill, due to the significant architectural differences between teacher and student models. Table 6 presents the performance of InDistill combined with 5 baseline methods. Teacher and student baselines refer to from-scratch training without KD, while auxiliary baseline refers to performance after KD from the teacher. The proposed method consistently boosts the performance of all the core KD approaches in both retrieval and classification settings. Specifically, for the mAP metric, InDistill boosts performance significantly across methodologies, such as +3.44% for KD, +3.20% for PKT, +2.46% for CRD, +1.26% for DKD, and +2.16% for MLKD. Similarly, for the classification setting, InDistill increases the accuracy of all methods, with improvements ranging from +0.86% to +2.09%. Finally, in terms of latency the students’ mean inference time is 14.89× faster than the teachers’ (i.e., 0.49ms on CPU) and in terms of storage requirements the students’ storage requirement is 28KB while teachers’ is 42.7MB.

\(^1\)The presented values are computed based on cosine similarity.
5.4. Preserving information flow

In Fig. 4, we present the $\mathcal{L}_{MI}$ loss curves to assess the proposed method’s capability to distill the teacher’s information flow. In particular, the loss curves of InDistill without a learning scheme (i.e., transferring knowledge from all layers simultaneously), with the WD scheme [36], and with the curriculum learning scheme, are depicted for comparison. Although InDistill without learning scheme reduces $\mathcal{L}_{MI}$ during the first training epochs (when the critical connections are created) compared to the baseline PKT method, it impedes the learning process during the last training epochs and achieves 51.54% mAP. On the contrary, applying a learning scheme aware of the critical learning periods, such as WD and the proposed learning scheme, can successfully address this limitation. However, the proposed scheme is superior to WD, as not only is aware of the critical learning periods but it also takes into account the complexity of transferring the knowledge of multiple layers. This can also be confirmed by the mAP values being 54.61% and 53.91% for the proposed scheme and WD, respectively. Thus, we confirm that forcing the architectural alignment and being aware of the layers’ successive transfer difficulty enables the student model to mimic the teacher’s information flow more effectively.

5.5. Ablation study

Table 7 presents how InDistill’s components (i.e., teacher pruning and curriculum learning) cumulatively affect the student’s performance. Furthermore, as already demonstrated in Fig. 4, pruning contributes to the critical connections’ creation during the first training epochs, but eventually hinders the training procedure. On the contrary, pruning combined with curriculum learning enables the critical connections formation as well as the effective learning of the final task, and results in considerably enhanced student’s performance. Finally, Fig. 5 presents how the hyperparameters $a$ and $b$ affect the student’s performance when applying InDistill via our curriculum scheme.

6. Conclusions

In this work, we introduce a curriculum learning-based training scheme that acts as a warmup step to preserve the teacher model’s information flow paths, thereby improving the effectiveness of any subsequent KD approach. Additionally, when there is a difference in the width of the teacher and student layers, we apply a pruning operation that reduces the width of the teacher’s intermediate layers to align with the student’s, enabling direct distillation. The above strategy effectively enhances the performance of many baseline methods on a wide range of retrieval and classification experiments.

Acknowledgments

This research was supported by the European Commission under the Horizon Europe project ELIAS (Grant Agreement 101120237), the H2020 projects AI4Media (Grant Agreement 951911) and MediaVerse (Grant Agreement 957252), and the Junior Star GACR project (Grant Agreement 21-28830M).
References

[1] Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep networks. In Proc. ICLR, 2019. 2, 3
[2] Nima Aghli and Eraldo Ribeiro. Combining weight pruning and knowledge distillation for cnn compression. In Proc. CVPR, 2021. 2, 3
[3] Yoshua Bengio. Learning deep architectures for AI. Now Publishers Inc, 2009. 2
[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Proc. ICML, 2009. 3
[5] 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. 2
[6] Shi Chen and Qi Zhao. Shallowing deep networks: Layerwise pruning based on feature representations. IEEE TPAMI, 41(12), 2018. 2, 3
[7] Zhiqiang Chen, Ting-Bing Xu, Changde Du, Cheng-Lin Liu, and Huiguang He. Dynamical channel pruning by conditional accuracy change for deep neural networks. IEEE TNNLS, 32(2), 2020. 2
[8] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine, 35(1), 2018. 1
[9] Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannath Sarangapani. A comprehensive survey on model compression and acceleration. Artificial Intelligence Review, 53(7), 2020. 1, 2
[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proc. CVPR. IEEE, 2009. 2, 5
[11] Xiang Deng and Zhongfei Zhang. Comprehensive knowledge distillation with causal intervention. In Proc. NeurIPS, 2021. 2
[12] Xiaohan Ding, Tianxiao Hao, Jianchao Tan, Ji Liu, Jungong Han, Yuchen Guo, and Guiguang Ding. Resrep: Lossless cnn pruning via decoupling remembering and forgetting. In Proc. ICCV. 2021. 2
[13] Chen Gong, Dacheng Tao, Stephen J Maybank, Wei Liu, Guoliang Kang, and Jie Yang. Multi-modal curriculum learning for semi-supervised image classification. IEEE TIP, 25(7), 2016. 3
[14] Jianping Hou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. IJCV, 129(6), 2021. 1
[15] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In Proc. ICML, 2017. 3
[16] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In Proc. NeurIPS, 2015. 4
[17] 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, 2016. 5
[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. stat, 1050, 2015. 1, 2, 5, 6, 7
[19] 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. 2, 5, 6, 7
[20] Faisal Khan, Bilge Mutlu, and Jerry Zhu. How do humans teach: On curriculum learning and teaching dimension. In Proc. NeurIPS, 2011. 3
[21] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In Proc. NeurIPS, 2018. 3
[22] Seung Wook Kim and Hyo-Eun Kim. Transferring knowledge to smaller network with class-distance loss. In Proc. ICLR Workshop, 2017. 2
[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. International Conference for Learning Representations, 2015. 5
[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report, 2009. 2, 5
[25] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In Proc. ICLR, 2017. 2, 3, 4
[26] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. Neurocomputing, 461, 2021. 2
[27] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In Proc. ICCV, 2017. 2
[28] Xiaolong Ma, Geng Yuan, Sheng Lin, Zhengang Li, Hao Sun, and Yanzhi Wang. Resnet can be pruned 60×: Introducing network purification and unused path removal (p-rm) after weight pruning. In Proc. NANOARCH. IEEE, 2019. 2
[29] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. TNNLS, 31(9), 2019. 3
[30] Zhong Meng, Jinyu Li, Yong Zhao, and Yifan Gong. Conditional teacher-student learning. In Proc. NeurIPS, 2019. 3
[31] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In Proc. AAAI, 2020. 3, 4
[32] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In Proc. NeurIPS, 2019. 2
[33] Dae Young Park, Moon-Hyun Cha, Daesin Kim, Bohyung Han, et al. Learning student-friendly teacher networks for knowledge distillation. In Proc. NeurIPS, 2021. 2
[34] Jihyuk Park and Albert No. Prune your model before distill it. In European Conference on Computer Vision, pages 120–136. Springer, 2022. 2
[35] Nikolaoas Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In Proc. ECCV, 2018. 2, 4, 5, 6, 7, 8
[36] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In Proc. CVPR, 2020. 1, 2, 3, 4, 5, 7, 8
[37] Anastasia Pentina, Viktoria Sharmanska, and Christoph Lampert. Curriculum learning of multiple tasks. In Proc. CVPR, 2015. 3
[38] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Proc. ICLR, 2015. 1, 2
[39] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. IJCV, 2022. 3
[40] Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15731–15740, 2024. 2
[41] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In International conference on machine learning, pages 1139–1147. PMLR, 2013. 5
[42] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In Proc. ICLR, 2019. 3, 5, 6, 7
[43] Zi Wang, Chengcheng Li, and Xiangyang Wang. Convolutional neural network pruning with structural redundancy reduction. In Proc. CVPR, 2021. 2
[44] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In Proc. NeurIPS, 2016. 4
[45] Chuanguang Yang, Zhulin An, Linhang Cai, and Yongjun Xu. Hierarchical self-supervised augmented knowledge distillation. In Proc. IJCAI, 2021. 2
[46] Zhengwu Yang and Han Zhang. Comparative analysis of structured pruning and unstructured pruning. In International Conference on Frontier Computing, pages 882–889. Springer, 2021. 2
[47] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In Proc. CVPR, 2017. 3, 5, 7
[48] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In Proc. ICLR, 2017. 1, 2
[49] Wojciech Zaremba and Ilya Sutskever. Learning to execute. arXiv preprint arXiv:1410.4615, 2014. 3
[50] Jiwen Zhang, Jianqing Fan, Jiajie Peng, et al. Curriculum learning for vision-and-language navigation. In Proc. NeurIPS, 2021. 3
[51] 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. 2, 5, 6, 7
[52] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In Proc. NeurIPS, 2018. 2