ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240786

# Grouped Logit Distillation Enhanced with Superclass Awareness for Efficient Knowledge Transfer

**Shuoxi Zhang**[a,*], **Hanpeng Liu**[a,**], **Yuyi Wang**[b,c,***] **and Kun He**[a,****,1]

[a]School of Computer Science and Technology, Huazhong University of Science and Technology
[b]CRRC Zhuzhou Institute, Zhuzhou, China.
[c]Tengen Intelligence Institute, Zhuzhou, China.
ORCID (Kun He): https://orcid.org/0000-0001-7627-4604.

**Abstract.** Knowledge distillation (KD) facilitates student training by transferring information beyond plain labels, specifically through the categorical relationships from the teacher. However, this class relationship knowledge is, by nature, easily dominated by a few classes. This phenomenon prevents knowledge distillation from fully extracting the knowledge of the teacher model, thereby impeding the transfer of knowledge. To this end, we introduce a grouping strategy to the knowledge distillation paradigm, termed **G**rouped **L**ogit **D**istillation (GLD). This strategy involves distilling knowledge within each group and across all groups, potentially transferring relationships in a comprehensive manner. Furthermore, we delve deeper into the grouping mechanism and attempt to incorporate a **superclass** mechanism using information derived from features of the teacher model. Our enhanced version, GLD++, performs knowledge distillation more meticulously by organizing information based on superclasses. We evaluate the effectiveness of our approaches through extensive experiments across standard benchmark datasets, obtaining state-of-the-art performance.

## 1 Introduction

Deep learning has significantly advanced many fields, including computer vision [8, 34, 35], natural language processing [16, 39], and speech recognition [26, 7]. The efficacy of Deep Neural Networks (DNNs) stems from their over-parameterized architecture, which extracts rich representative features from raw data. However, the complexity of DNNs presents challenges in terms of computational requirements, particularly for mobile devices. To address this issue, there is an increasing demand to reduce the computational cost and facilitate the deployment of deep learning models in resource-constrained environments. Consequently, model compression, including knowledge distillation [9], quantization [6, 45], and pruning [5], has become an important area. The primary objective of these techniques is to maintain the high performance while enhancing their practicality and efficiency for broader applications.

Due to its easy implementation, Knowledge Distillation [9] (KD) has emerged as a primary choice for mitigating the substantial computational demands in mobile scenarios. KD provides a solution by facilitating knowledge transfer from a larger, computationally intensive teacher model to a smaller, parameter-compact student model. This involves training the student model to emulate the teacher's behavior, leading to compact and capable models. Because of its effectiveness in reducing model complexity while preserving performance, KD has become prominence as a key strategy for deploying deep learning solutions in resource-constrained environments.

Existing KD methods can be categorized into two main branches: logit-based and feature-based. As introduced in vanilla KD [9], Hinton *et al.* conduct knowledge transfer on the prediction logits using a temperature scaling strategy to transfer "dark knowledge", or the class relationship information, from the teacher to the student. The alignment of logits between the teacher and student models facilitates the transfer of categorical information. In contrast, feature distillation focuses on matching the intermediate features between teacher and student models. This approach has gained popularity due to the richness of the feature representations, often outperforming logit-based distillation methods in various tasks. However, the increased design complexity and computational overhead associated with feature alignment, particularly when the teacher and student models differ in size and architecture, present significant challenges.

Intuitively, one might expect logit-based methods to match the effectiveness of feature distillation given that logits encode higher-level semantics. To identify potential weaknesses, we analyze the logits outputted by the teacher network. We observe a pronounced *skewed distribution*, where only a few class logits are highly activated, while the vast majority of others remain near zero (in Sec. 3.2). Objectively, such skew enables the network to classify certain classes with high confidence. However, within the KD framework, this characteristic restricts knowledge transfer, preventing full utilization of the rich relational information among classes. Therefore, a more comprehensive approach to distilling dark knowledge, that involves leveraging the relationships among all categories, remains unexplored.

In this paper, we introduce a novel logit distillation technique termed **G**rouped **L**ogit **D**istillation (**GLD**). Within the GLD framework, we separate the traditional distillation loss into two components: inter-group and intra-group distillation losses. Specifically, we categorize all classes into distinct groups and adjust the probability distributions within each group to optimize learning. The intra-group distillation loss aims to distill the relationships within each group's probability distribution, while the inter-group loss seeks to transfer

* zhangshuoxi@hust.edu.cn
** hanpengliu@hust.edu.cn
*** yuyiw920@163.com
**** brooklet60@hust.edu.cn
[1] Corresponding author.

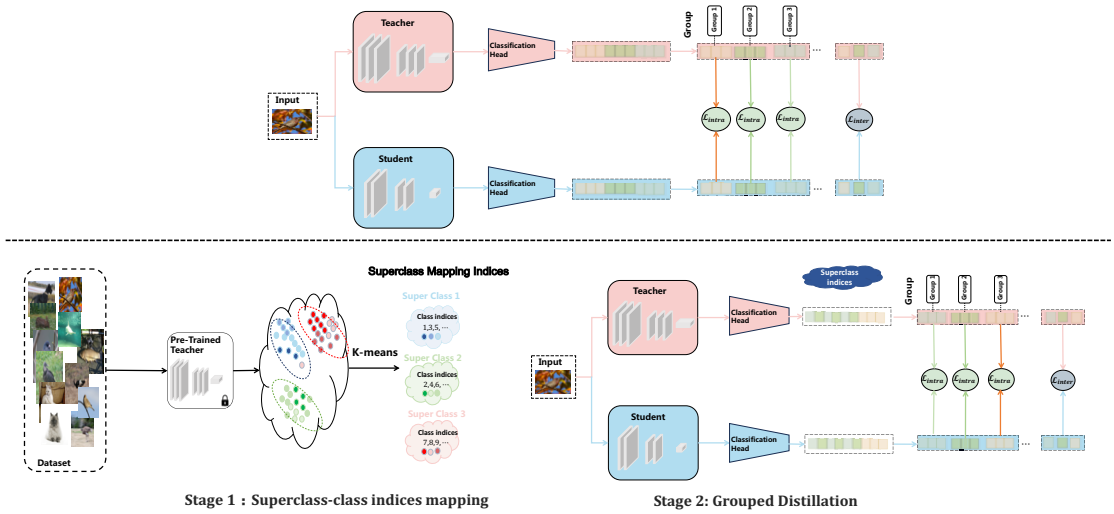


**Figure 1**: Overall framework of our GLD (above) and GLD++ (below). GLD++ conducts $K$-means clustering on the feature from pre-trained teacher to generate superclass mapping indices. Then we use the indices to group the logits in the hierarchical fashion.

the knowledge of probability distributions between groups. This approach enables efficient distillation of the teacher's *dark knowledge* regarding the relationships among various categories, thus alleviating the knowledge gap present in existing KD approaches.

Furthermore, we strive to develop more effective strategies for grouped distillation. Specifically, we adapt the principle of hierarchical categorization from biology science to knowledge distillation, classifying entities from *classes* to *superclasses*. This method is based on the biological rationale of organizing entities according to their evolutionary relationships. We argue that this approach can be analogously implemented in our GLD framework. We propose the **superclass-aware grouped logit distillation** (denoted as **GLD++**). In this enhanced version, we introduce the concept of *superclass* to present aggregations of classes with shared characteristics. Initially, we perform clustering on the feature representation of the pre-trained teacher to define superclass indices. We then align the logits of the teacher and the student models according to these superclass indices. The final phase involves implementing GLD on these aligned logits. Through GLD++, our objective is to distill information that reflects the hierarchical relationships among categories, thus improving the precision of the knowledge transferred to the student model.

To validate the effectiveness of our proposed approach, we conduct a comprehensive series of experiments on standard benchmark datasets. The results consistently show that our methods, denoted as GLD and GLD++, surpass all contemporary state-of-the-art techniques, including several prominent feature-based distillation methods, without increasing time costs. These findings underscore the efficiency and efficacy of our approach across various distillation scenarios, establishing it as a promising solution for knowledge distillation in deep learning.

Our main contributions can be summarized as follows:

- We identify a critical issue — the skewed distribution that plagues current logit-based KDs. To address this, we introduce a grouping strategy to logit distillation by decoupling the conventional KD loss into intra- and inter-group losses. We contend that this paradigm facilitates distillation in a more fine-grained manner.
- Furthermore, we propose the enhanced version of GLD— GLD++. GLD++ abstracts the superclass notion to benefit the subsequent grouping distillation. To our best knowledge, this is the first attempt to apply the hierarchy structure in biology to the distillation paradigm, which fully leverages the knowledge learned from the pre-trained teacher.
- Our approach consistently outperforms state-of-the-art baselines in extensive experiments, encompassing various network architectures and diverse tasks, including classification and detection.

## 2 Related Work

In this section, we first provide a concise overview of prior research in knowledge distillation, covering prevalent methods in both logit and feature distillation. Subsequently, we review foundational studies on clustering, a strategy we employ to generate superclasses.

### 2.1 Logit Distillation

Hinton *et al.* [9] first design the temperature scaling strategy on prediction logits to distill the teacher's *dark knowledge* of category prediction probabilities. This approach is based on the premise that the relational information within the probability distribution can provide additional supervision and crucial regularization [40]. Utilizing this characteristic, logit distillation has demonstrated the potential to improve generalization of student models. However, a significant knowledge gap still exists, prompting further investigation into how to transfer knowledge effectively. To narrow this gap, researchers propose distinct methods to improve the efficiency of knowledge transfer. For instance, Huang *et al.* [12] introduce the Pearson coefficient loss to boost the distillation process. Moreover, Zhao *et al.* [44] astutely notice that directly minimizing the KL divergence of logit distributions could impede effective knowledge transfer. To address this, they decouple the conventional KD loss into two distinct components: target and non-target losses. However, they still overlook the primary reason for the *skewed distribution* that limits the utility of the vanilla KD loss. To the best of our knowledge, we are the first to introduce the grouping strategy within the KD framework to mitigate the knowledge gap.

### 2.2 Feature Representation and Distillation

Feature representation is fundamental to the success of deep learning models. The field has evolved from relying on hand-crafted fea-

tures to automated feature learning, which leverages complex transformations and multiple levels of abstraction. Foundational works by Krizhevsky *et al.* [20] and LeCun *et al.* [22] highlight the capability of deep networks to extract semantically rich features from data, signifying a critical transition towards feature learning that can decipher complex patterns within extensive datasets. The integration of feature learning with knowledge distillation has since emerged as an effective strategy to overcome the limitations of logit distillation. This was first introduced in FitNet [29], which encouraged student models to mimic the salient features of their teachers. Subsequent methods have further refined the alignment and transfer of knowledge from teacher features, employing techniques such as attention mechanisms [41], neural selectivity [13], and paraphrasing [18]. While these feature-based distillation approaches have proven more versatile and effective than their logit-based counterparts, they pose challenges such as size mismatches between teacher and student features, leading to increased computational and memory demands. Unlike directly aligning the feature representation for KD, our enhanced distillation GLD++ utilizes features to construct the superclass clustering.

### 2.3 Clustering

Clustering is an essential technique for grouping entities with shared attributes into distinct categories. Over recent decades, the literature on clustering strategies has expanded considerably. Foundational methods such as $K$-means [1], hierarchical clustering [27], and DBSCAN [17] have established the groundwork for dividing data into meaningful clusters. The $K$-means algorithm partitions data into $K$ separate, non-overlapping clusters by minimizing intra-cluster variance. In contrast, hierarchical clustering generates a dendrogram that represents multilevel groupings of data based on a distance metric. DBSCAN, meanwhile, is particularly effective at identifying clusters of various shapes in dense regions, handling noise robustly, and offering an alternative to centroid-based clustering. The goal of our enhanced method, GLD++, is to ensure that each group, or superclass, comprises semantically related categories. Clustering serves as the optimal approach to achieving this goal. Specifically, we apply $K$-means clustering to the features extracted from a pre-trained teacher network to define the mappings for superclasses.

## 3 The Proposed Method

Consider the training set $\mathcal{D} = (\boldsymbol{x}_i, y_i)_{i=1}^N$ with $N$ training samples, where $\boldsymbol{x}_i$ denotes the $i$-th sample and $y_i$ the corresponding label over $C$ classes. The intermediate feature and the prediction logit of the $i$-th instance are denoted as $\boldsymbol{f}_i, \boldsymbol{z}_i$, respectively. Moreover, their dimensions are $C$ and $D$ respectively. In this section, we first discuss the basic concept of KD. Subsequently, we identify the primary impediment to the effectiveness of KD as the *skewed distribution*. To mitigate this issue, we introduce a novel methodology named Grouped Logit Distillation (**GLD**). Finally, we illustrate an enhanced version of the GLD — **GLD++**, which incorporates superclass relativity to facilitate the grouping paradigm.

### 3.1 Vanilla Knowledge Distillation

The concept of knowledge distillation, initially introduced by Hinton *et al.* [9], is designed for transferring knowledge from a well-trained teacher model to a compact student. In this process, knowledge is conveyed via the softened logits. Consequently, the optimization of the loss function within the vanilla KD can be written as follows:

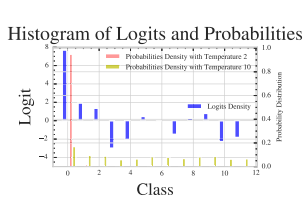


**Figure 2**: Histogram of logit and Softmax scores with different $\tau$.

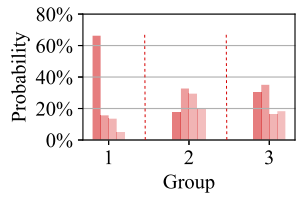


**Figure 3**: Probability distribution upon grouped logits.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \alpha \mathcal{L}_{\text{KD}}. \tag{1}$$

The loss terms $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{KD}}$ represent the cross-entropy (CE) and KD losses, respectively. The hyper-parameter $\alpha$ balances the two losses. Specifically, $\mathcal{L}_{\text{CE}}$ quantifies the discrepancy between the model predictions and the ground truth labels $y$, as follows:

$$\mathcal{L}_{\text{CE}}(\boldsymbol{x}_i, y_i) = -\sum_{j=1}^{C} y_{ij} \log \boldsymbol{p}_{ij}, \tag{2}$$

where the probability distribution $\boldsymbol{p}_i$ is computed on the prediction logits $\boldsymbol{z}_{\cdot,j}$ (the $j$-th class) by the Softmax equation:

$$\text{Softmax}(\boldsymbol{z}_{\cdot,j}, \tau = 1) = \frac{\exp(\boldsymbol{z}_{\cdot,j}(x)/\tau = 1)}{\sum_{j'} \exp(\boldsymbol{z}_{\cdot,j'}(x)/\tau = 1)}. \tag{3}$$

The hyper-parameter $\tau$ is the temperature parameter that aims to soften the probability distribution, and in the CE loss we use the primary probability distribution with $\tau = 1$. A higher $\tau$ tends to smooth the probability distribution across the classes, potentially highlighting the relationships among the classes. Utilizing softened logits, the KD framework facilitates the transfer of *dark knowledge* from the teacher to the student by minimizing the Kullback-Leibler (KL) divergence [15] between their respective prediction logits. The corresponding KD loss can be expressed as follows:

$$\mathcal{L}_{\text{KD}} = KL(p^{\text{tea}} || p^{\text{stu}}) = \sum_{j=1}^{C} p_j^{\text{tea}} \log \frac{p_j^{\text{tea}}}{p_j^{\text{stu}}}, \tag{4}$$

where $p_i^{\text{tea}}$ and $p_i^{\text{stu}}$ indicate the probability distribution on the $i$-th instance from the teacher and student output, respectively. All the prediction logits of the teacher and the student are scaled by temperature $\tau > 1$ as in Eq. (3).

### 3.2 Grouped Knowledge Distillation

We start with a toy example by randomly selecting an image from CIFAR-100 and passing it through a pre-trained ResNet-18. The probabilistic histogram for the image's ground-truth class and 11 randomly selected other classes is displayed in Fig. 2 (The choice of 12 classes is solely for better illustration.). This demonstrates that only a few categories closely related to the target class exhibit significantly high values, while the logits for other categories are nearly negligible. Consequently, applying a uniform temperature to soften the logits introduces challenges: a high temperature ($\tau = 10$) flattens all probability distributions, diminishing their ability to convey meaningful semantic information, while a low temperature ($\tau = 2$) restricts the KD framework to the relative relationships among a narrow set of categories. Although softened logits provide broader insights into category relationships compared to ground truth labels, fully leveraging the teacher's comprehensive understanding of relationships across all classes remains challenging.

To address this challenge, we propose Grouped Logit Distillation (GLD). In the toy example mentioned earlier, we divide the 12 classes into 3 groups, with each group containing four classes, and the ground-truth class placed in the first group. As shown in Fig. 3, by grouping the prediction logits, we can uncover relationships among categories (Groups 2 and 3) that previously exhibited low probabilities (in Classes 4-11 of Figure 2), thus enabling a more comprehensive distillation of the teacher model's "dark knowledge." Assuming we group $C$ categories into $G$ groups, with each group representing a probability distribution for $\frac{C}{G}$ categories, we formulate the distillation loss function for intra-group probability distributions as follows:

$$\mathcal{L}_{intra} = \sum_{g=1}^{G} KL(\boldsymbol{p}_g^{\text{tea}} || \boldsymbol{p}_g^{\text{stu}}). \tag{5}$$

Here $\boldsymbol{p}_g \in \mathbb{R}^{C/G}$ denotes the probability distribution of the $g$-th group, which can be computed by Eq. 3 upon on the grouped logits of the teacher and student, respectively. In our study, we use $\tau = 4$ to soften the grouped logits.

One may question whether intra-group distillation alone is sufficient to distill the teacher's prediction confidence. We contend that this knowledge can be transferred via inter-group distillation. Therefore, in addition to the intra-group loss, we incorporate distribution loss across groups. This process aggregates probabilities within each group, yielding a collective likelihood for each category within that group. For example, considering a 6-category probability distribution `{0.47, 0.03, 0.16, 0.10, 0.12, 0.12}`. Here, we divide the inter-group probability distribution with 3 groups, resulting in the inter-group probabilities as `{0.47+0.03, 0.16+0.10, 0.12+0.12}`.The teacher's prediction information related to the ground-truth label is captured within the probability of the group that contains the label. This inter-group technique simplifies the model's predictions by consolidating category probabilities at the group level, thereby facilitating a more holistic knowledge transfer. We define our inter-group distillation loss as follows:

$$\mathcal{L}_{inter} = KL(\boldsymbol{P}^{\text{tea}} || \boldsymbol{P}^{\text{stu}}), \tag{6}$$

where $\boldsymbol{P} \in \mathbb{R}^{G}$ denotes the inter-group probability distribution. Thus, the total KD loss in our paradigm can be written as:

$$\mathcal{L}_{\text{KD}} = \mathcal{L}_{intra} + \mathcal{L}_{inter}. \tag{7}$$

The overall framework of GLD is presented in Fig. 1 (above).

### 3.3 Superclass-Aware Grouped Distillation

Intuitively, it is logical for GLD, as discussed in the previous section, to ensure that each group shares a similar semantics. This concept is inspired by the hierarchical classification commonly used in biology, where multiple `subspecies` constitute a `species`, analogous to how classes form the superclass in our framework. In GLD, if the intra-group distribution can effectively capture the relationships among similar classes, and the inter-group distribution describe the probabilistic associations between different superclasses, then this grouped KD approach would be more coherent. To achieve this, we introduce the concept of *superclass* in our enhanced method: superclass-aware grouped logit distillation (GLD++). In this enhanced version, instead of arbitrarily grouping logits, we first generate class-superclass mapping indices, and use these indices to guide the grouping and subsequent GLD.

One might consider clustering as the most suitable method for generating subclasses. Given that the KD paradigm utilizes a pre-trained teacher capable of extracting features to convey the semantic information of original images, we perform pre-clustering on the training dataset before group distillation in our enhanced method. We employ the basic clustering technique, $K$-means, and assign each original class to the superclass with the **highest data frequency** as the corresponding superclass. As shown in Fig. 1, a sample from Class 8 (denoted with pink) is misclustered into Superclass 1; however, the majority of Class 8 falls into Superclass 3, thus Class 8 belongs to Superclass 3. With this paradigm, we may generate a mapping table between superclasses and original classes, which guides the grouping of logits for the subsequent grouped distillation. We contend that we may distill coherent and precise teacher knowledge. Intuitively, the more powerful the teacher's feature extraction capability is, the more accurate the superclass identification will be. We will verify this hypothesis in the experimental section.

## 4 Experiments

To demonstrate the efficacy of our methodology, we conduct a comparative analysis of GLD and GLD++ with various leading-edge KD techniques. The evaluation is conducted upon diverse tasks, including image classification and object detection.

### 4.1 Experimental Setup

**Datasets.** For the classification evaluation, we conduct the comparison on the CIFAR-100 dataset [20] and ImageNet [30]. The CIFAR-100 dataset comprises $32 \times 32$ pixel color images representing objects from 100 distinct categories. Furthermore, we evaluate the effectiveness of our method in a large-scale classification context using the ImageNet dataset, where all images are uniformly resized to $224 \times 224$ pixels. Both datasets undergo standard data augmentation and normalization, in accordance with methodologies described in [8, 42, 11]. For the detection evaluation, we utilize the MS-COCO dataset [23], which comprises 118k training images and 5k validation images across 80 categories.

**Networks.** We evaluate the performance of mainstream and lightweight network architectures on the CIFAR-100 and ImageNet datasets, following the methodologies described in [2]. Our experiments involve a range of popular teacher-student pairs including ResNet [8], VGG [31], WideResNet [42], MobileNet [10], and ShuffleNet [43]. For CIFAR-100, we modify ResNet to better suit small-scale datasets by integrating PreAct layers [8]. In the ImageNet evaluations, we expand our investigation to include the impact of distillation from large pre-trained models such as BiT [19] and Swin [25], beyond the basic network configurations. For detection evaluation, we take Faster-RCNN [28]-FPN [38] as the backbone, AP50, and AP75 as the evaluation metric.

**Training Details.** For CIFAR-100 training, we utilize the SGD optimizer with 0.9 Nesterov momentum across a total of 240 epochs, reducing the learning rate by a factor of 10 at the 150-th, 180-th, and 210-th epochs. We employ standard data augmentation techniques, such as flipping and random cropping. The initial learning rate is set at 0.01 for lightweight architectures and 0.05 for others, with an additional weight decay of $5e^{-4}$ for $L_2$ regularization. For ImageNet training, we follow the practice by PyTorch official, and set the batch size at 512 and the weight decay rate at $1e^{-4}$. The initial learning rate of 0.1 is modified according to a cosine learning rate schedule, aiming to optimize top-1 accuracy on the validation set. All ImageNet experiments are performed on 4 RTX 3090 GPUs, with the total epochs set at 120, focusing on maximizing top-1

**Table 1**: Top-1 test accuracy (%) of various approaches on CIFAR-100. The teacher-student pairs share similar architectures. Each experiment is repeated three times, we report the mean and standard deviation of the top-1 accuracy. The best results appear in **bold** and the second best results appear with underline.

| Type | Student | ResNet-8×4<br>72.51 ± 0.29 | VGG-8<br>70.46 ± 0.29 | ResNet20<br>69.06 ± 0.22 | WRN-40-1<br>71.98 ± 0.17 | WRN-16-2<br>73.43 ± 0.22 | ResNet32<br>71.14 ± 0.25 |
|---|---|---|---|---|---|---|---|
| — | Teacher | ResNet-32×4<br>79.42 | VGG-13<br>74.64 | ResNet56<br>73.44 | WRN-40-2<br>76.31 | WRN-40-2<br>76.31 | ResNet110<br>74.31 |
| Logits | KD [9] | 74.12 ± 0.15 | 72.66 ± 0.13 | 70.66 ± 0.22 | 73.42 ± 0.22 | 74.92 ± 0.20 | 73.02 ± 0.16 |
| | DTD-KA [37] | 73.78 ± 0.22 | 72.98 ± 0.14 | 70.99 ± 0.24 | 73.49 ± 0.16 | 74.73 ± 0.20 | 72.88 ± 0.13 |
| | DKD [44] | 76.32 ± 0.26 | 74.68 ± 0.23 | 71.79 ± 0.17 | 76.11 ± 0.17 | 76.55 ± 0.14 | 74.11 ± 0.17 |
| | DIST [12] | 76.55 ± 0.21 | 74.71 ± 0.21 | 71.85 ± 0.15 | 76.14 ± 0.11 | 76.11 ± 0.17 | 74.10 ± 0.15 |
| | MLD [14] | 77.08 ± 0.19 | 75.18 ± 0.16 | 72.19 ± 0.13 | 75.35 ± 0.17 | 76.63 ± 0.14 | 74.11 ± 0.17 |
| Features | FitNet [29] | 73.89 ± 0.22 | 73.54 ± 0.12 | 71.52 ± 0.16 | 74.12 ± 0.20 | 75.75 ± 0.12 | 72.52 ± 0.07 |
| | AT [41] | 74.57 ± 0.17 | 73.63 ± 0.12 | 71.76 ± 0.14 | 74.43 ± 0.11 | 75.28 ± 0.13 | 73.32 ± 0.11 |
| | SP [33] | 73.90 ± 0.17 | 73.44 ± 0.21 | 71.48 ± 0.11 | 73.17 ± 0.21 | 75.34 ± 0.21 | 73.63 ± 0.21 |
| | CRD [32] | 75.59 ± 0.23 | 73.88 ± 0.18 | 71.68 ± 0.11 | 75.51 ± 0.22 | 76.01 ± 0.11 | 73.48 ± 0.16 |
| | SemCKD [2] | 75.58 ± 0.22 | 74.42 ± 0.21 | 71.98 ± 0.17 | 74.78 ± 0.21 | 75.42 ± 0.15 | 74.12 ± 0.22 |
| | ReviewKD [3] | 76.42 ± 0.20 | 74.61 ± 0.17 | 71.98 ± 0.11 | 75.78 ± 0.19 | 75.88 ± 0.05 | 74.17 ± 0.20 |
| | NORM [24] | 76.76 ± 0.22 | 73.95 ± 0.14 | 71.35 ± 0.13 | 75.42 ± 0.18 | 76.26 ± 0.05 | 73.95 ± 0.31 |
| Logits | GLD (Ours) | 77.22 ± 0.21 | 75.88 ± 0.17 | 72.34 ± 0.11 | 76.38 ± 0.11 | 76.48 ± 0.21 | 74.84 ± 0.11 |
| | GLD++ (Ours) | **77.81 ± 0.19** | **76.23 ± 0.21** | **72.77 ± 0.23** | **76.76 ± 0.14** | **76.89 ± 0.21** | **75.22 ± 0.14** |

**Table 2**: Top-1 test accuracy (%) of various distillation approaches with different architectures on CIFAR-100.

| Type | Student | ShuffleV1<br>70.50 ± 0.22 | WRN-16-2<br>73.43 ± 0.22 | VGG-8<br>70.46 ± 0.29 | MobileV2<br>64.60 ± 0.32 | MobileV2<br>64.60 ± 0.32 | ShuffleV1<br>70.50 ± 0.22 |
|---|---|---|---|---|---|---|---|
| — | Teacher | ResNet-32x4<br>79.42 | ResNet-32x4<br>79.42 | ResNet50<br>79.10 | WRN-40-2<br>76.31 | VGG-13<br>74.64 | WRN-40-2<br>76.31 |
| Logits | KD [9] | 74.00 ± 0.16 | 74.90 ± 0.29 | 73.81 ± 0.24 | 69.07 ± 0.26 | 67.37 ± 0.22 | 74.83 ± 0.13 |
| | DTD-KA [37] | 73.99 ± 0.12 | 74.11 ± 0.21 | 73.91 ± 0.21 | 68.99 ± 0.41 | 67.41 ± 0.12 | 74.90 ± 0.14 |
| | DKD [44] | 77.42 ± 0.11 | 76.68 ± 0.22 | 75.98 ± 0.22 | 69.47 ± 0.21 | 69.71 ± 0.26 | 76.41 ± 0.13 |
| | DIST [12] | 77.21 ± 0.14 | 76.70 ± 0.17 | 75.88 ± 0.20 | 69.78 ± 0.24 | 69.81 ± 0.22 | 76.05 ± 0.11 |
| | MLD [14] | 77.18 ± 0.12 | 76.41 ± 0.15 | 75.71 ± 0.20 | 69.41 ± 0.24 | 70.23 ± 0.22 | 76.44 ± 0.11 |
| Features | FitNet [29] | 74.82 ± 0.13 | 74.70 ± 0.35 | 73.72 ± 0.18 | 68.71 ± 0.21 | 63.16 ± 0.23 | 74.11 ± 0.23 |
| | AT [41] | 74.76 ± 0.19 | 75.38 ± 0.18 | 73.45 ± 0.17 | 68.64 ± 0.12 | 63.42 ± 0.21 | 73.73 ± 0.19 |
| | SP [33] | 73.80 ± 0.21 | 75.16 ± 0.32 | 73.86 ± 0.15 | 68.48 ± 0.22 | 65.42 ± 0.21 | 74.01 ± 0.11 |
| | CRD [32] | 75.46 ± 0.23 | 75.70 ± 0.29 | 74.42 ± 0.21 | 69.87 ± 0.17 | 69.73 ± 0.21 | 76.05 ± 0.23 |
| | SemCKD [2] | 75.41 ± 0.11 | 75.65 ± 0.23 | 74.68 ± 0.22 | 69.88 ± 0.30 | 68.78 ± 0.22 | 74.81 ± 0.21 |
| | ReviewKD [3] | 76.21 ± 0.10 | 76.22 ± 0.21 | 75.68 ± 0.27 | 69.28 ± 0.35 | 69.73 ± 0.12 | 75.78 ± 0.21 |
| | NORM [24] | 77.42 ± 0.17 | 76.41 ± 0.19 | 75.67 ± 0.18 | 69.78 ± 0.24 | 68.93 ± 0.17 | 77.06 ± 0.11 |
| Logits | GLD (Ours) | 77.63 ± 0.18 | 76.99 ± 0.22 | 76.14 ± 0.14 | 69.99 ± 0.11 | 70.18 ± 0.21 | 77.27 ± 0.13 |
| | GLD++ (Ours) | **77.91 ± 0.20** | **77.48 ± 0.20** | **76.54 ± 0.21** | **70.83 ± 0.25** | **70.68 ± 0.22** | **77.89 ± 0.13** |

accuracy in the validation set.

**Baselines.** We compare our approach with two main kinds of KD baselines (*i.e.*, logit-based and feature-based distillation):

- **Logit-based:** includes the vanilla KD [9], DTD-KA [37], DKD [44], DIST [12] and MLD [14].
- **Feature-based:** includes FitNet [29], AT [41], SP [33], CRD [32], SemCKD [2], ReviewKD [3] and NORM [24].

## 4.2 Image Classification

### 4.2.1 Results on CIFAR-100

**Results with Similar Architectures.** When the teacher and student have similar architectures, Table 1 shows the general superiority of our GLD and GLD++ over all existing mainstream methods. It is noteworthy that our two approaches surpass the state-of-the-art approaches, by 0.16% and 0.87% with ResNet-32×4/ResNet-8×4 network pairs and by 0.24% and 0.62% with WRN-40-2/WRN-40-1 network pairs. In addition, our method even beat the current feature-based method NORM, making our methods as good alternate for efficient distillation.

**Results with Different Architectures.** Similar results are found when we switch the student-teacher pairs to heterogeneous architectures. From Table 2, we observe that our method surpasses the state-of-the-art approaches. Especially when comparing our GLD++ to all cutting-edge methods, we may notice that our GLD++ outperforms all approaches by 0.49%, 0.78%, 0.56%, 0.95%, 0.45%, and 0.83% with the heterogeneous architecture pairs. These results from Tab. 1 and Tab. 2 clearly demonstrate the superiority of our method.

### 4.2.2 Results on ImageNet

**Results with Normal Teachers.** To show the effectiveness of our method on large-scale vision tasks, we evaluate our method on the ImageNet-1k dataset. Our tests include homogeneous (ResNet34/ResNet18) and heterogeneous (ResNet50/MobileNet) network pairs. The results are summarized in Table 3. Similar to CIFAR-100, our GLD outperforms the baselines on ImageNet-1k. We observe that our method achieves the best classification performance compared with the other distillation methods. Notably, our GLD++ surpasses KD by 2.76% and 3.03% in Top-1 accuracy with these two pairs, respectively. All these results illustrate the effectiveness of our method on large-scale dataset learning.

**Table 3**: Evaluation results of baseline settings on ImageNet. We use ResNet34/ResNet18 and ResNet50/MobileNet as teacher/student pairs and follow the standard PyTorch training practice for ImageNet.

| Student (Teacher) | Metric | Teacher | Student | AT | KD | SP | CRD | DKD | ReviewKD | DIST | GLD | GLD++ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet18 (ResNet34) | Top-1 | 73.31 | 69.75 | 70.70 | 70.66 | 70.62 | 71.17 | 71.51 | 71.61 | 71.88 | **72.81** | **73.42** |
| | Top-5 | 91.42 | 89.07 | 90.00 | 89.88 | 89.80 | 90.13 | 90.46 | 90.51 | 90.42 | **91.20** | **91.64** |
| MobileNet (ResNet50) | Top-1 | 76.16 | 70.13 | 70.78 | 70.68 | 70.99 | 71.37 | 72.66 | 72.56 | 72.94 | **73.42** | **73.81** |
| | Top-5 | 92.86 | 89.49 | 90.50 | 90.30 | 90.61 | 90.41 | 90.87 | 91.00 | 91.12 | **92.01** | **92.46** |

**Table 4**: Top-1 accuracies (%) of student networks ResNet18/50 distilled from large pre-trained models BiT-M-R50/Swin-L on the ImageNet-1k. "-" denotes directly training without any distillation. We represent our results in gray, and the ↑ present the improvements over vanilla KD.

| Method | Student | Teacher | |
|---|---|---|---|
| | | BiT-M-R50 [19] | Swin-L [25] |
| - | ResNet18 | 69.8 | 69.8 |
| KD [9] | ResNet18 | 70.7 | 70.9 |
| DIST [12] | ResNet18 | 71.8 | 72.2 |
| DKD [44] | ResNet18 | 72.1 | 72.4 |
| GLD | ResNet18 | **72.8 (↑ 2.1)** | **72.8 (↑ 1.9)** |
| GLD++ | ResNet18 | **73.8 (↑ 3.1)** | **74.1 (↑ 3.2)** |
| - | ResNet50 | 76.15 | 76.15 |
| KD [9] | ResNet50 | 77.6 | 78.5 |
| DIST [12] | ResNet50 | 78.2 | 78.8 |
| DKD [44] | ResNet50 | 78.1 | 78.7 |
| GLD | ResNet50 | **78.4 (↑ 0.8)** | **79.2 (↑ 0.7)** |
| GLD++ | ResNet50 | **79.4 (↑ 1.8)** | **80.2 (↑ 1.7)** |

**Results with Large Teachers.** Our evaluation on ImageNet-1K is extended by leveraging knowledge from large, pre-trained teacher models. We employ BiT-M-R50 and Swin-L as teachers. The results, summarized in Table 4, reveal notable findings. First, our approaches confer substantial benefits to network performance, as evidenced by our results exceeding the accuracy of prevailing distillation methods. Moreover, we observe that our GLD++ version outperforms the basic GLD variant with a considerable margin. This improvement may be attributed to large teachers providing a more accurate panorama of superclass clustering, which clearly benefits the subsequent grouped distillation. These compelling results robustly validate the efficacy of our approach in enhancing performance for large-scale training and underscore its effectiveness in knowledge distillation.

### *4.3 Object Detection*

We extend our experiments to objection detection, a more complicated task which needs the dense prediction of the network. We select several prevalent distillation techniques [44] as our benchmarks. All detection procedures remain unchanged, with the exception of the added distillation loss in our framework. As shown in Table 5, our method GLD and its upgraded version GLD++, achieves competitive results on MS-COCO validation set. The results show that our two methods significantly outperforms vanilla KD. Furthermore, our methods outperform the state-of-the-art feature-based approaches, demonstrating their effectiveness in knowledge distillation for dense prediction tasks.

## 5 Discussion

To better understand our grouped distillation, we conduct further experiments from three perspectives. Initially, we execute feature transfer experiments to demonstrate the transferability of the features learned through our distillation process. Secondly, we conduct the ablation study to show the indispensability and compatibility of the proposed losses and modules. Subsequently, we perform visualization upon our methods with several existing distillation methods.

### *5.1 Feature Transferability*

We continue to conduct several experiments to examine the feature transferability of our approaches. As shown in Tab. 6, we train linear fully-connected (FC) layers as the classifier with the feature extractor frozen for STL-10 [4] and Tiny-ImageNet [21] datasets. We use an SGD optimizer with 0.9 momentum and no weight decay strategy in classifier training. We set the batch size to 128, and the number of total epochs to 40. Our initial learning rate is set to 0.1, then divided by 10 for every 10 epochs. From Tab. 6, we observe that our method beats all existing techniques, manifesting its feature transferability.

### *5.2 Ablation Study*

**Indispensability of the Losses.** Table 7 presents the ablation study on the proposed losses. Notably, we observe that decoupling conventional KD loss into two parts, *i.e.*, intra-group loss $\mathcal{L}_{intra}$ and inter-group loss $\mathcal{L}_{inter}$ can enhance the performance of KD for both architectures. Moreover, it is interesting that intra-group loss or inter-group distillation loss alone can also enhance the prediction performance. However, the use of $\mathcal{L}_{inter}$ alone does not achieve the same efficacy as combined with $\mathcal{L}_{intra}$, this may owing to that using $\mathcal{L}_{inter}$ alone construct a coarse-grained probability distribution. Besides, all the results in GLD++ paradigm show their priority to GLD, suggesting that generating superclass indices may benefits the subsequent distillation process.

**The Compatibility of Grouping Paradigm.** We investigate into the compatibility of our grouping strategy with current mainstream distillation methods, and we present the result in Table 8. We incorporate our grouping and superclass grouping paradigm into DIST and DKD. Note that DIST distill the knowledge by using Pearson coefficient, thus we maintain this loss function except adding the grouping mechanism. Likely, as DKD decouples KD into target and non-target distillation, we act the grouping strategy upon DKD in its non-target part. We may observe that when we add the grouping and superclass grouping to both DIST and DKD, the performance improvement are consistent over all student-teacher pairs, confirming the compatibility of our module with prevailing logit-based KD approaches.

### *5.3 Visualization*

**t-SNE Visualization.** We present t-SNE visualizations of KD, our GLD and GLD++ in Fig. 4. From this visualization, We may observe that our GLD representations show better separability compared with KD. The result with GLD++ is even better. This result verifies that our approaches enhance the discernibility of the feature representation, which may benefit the classification performance.

**Time Efficiency.** One might question whether our method attain performance at the cost of training efficiency. To address this concern, we show the training time/accuracy scatter plot in Fig. 5. As GLD++ conducts an extra cluster before distillation, we compute the mean training time per batch for a fair comparison. The result presents the balance of our GLD++ to achieve time efficiency and performance.

**Table 5**: Results on MS-COCO. We take Faster-RCNN [28] with FPN [38] as the backbone, and AP, $AP_{50}$, and $AP_{75}$ as the evaluation metric. The original accuracy results of the teacher and student model are also reported.

| | | AP | $AP_{50}$ | $AP_{75}$ | AP | $AP_{50}$ | $AP_{75}$ | AP | $AP_{50}$ | $AP_{75}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | Teacher | ResNet101 | | | ResNet101 | | | ResNet50 | | |
| | | 42.04 | 62.48 | 45.88 | 42.04 | 62.48 | 45.88 | 40.22 | 61.02 | 43.81 |
| | Student | ResNet18 | | | ResNet50 | | | MobileNetV2 | | |
| | | 33.26 | 53.61 | 35.26 | 37.93 | 58.84 | 41.05 | 29.47 | 48.87 | 30.90 |
| Feature | FitNet [29] | 34.13 | 54.16 | 36.71 | 38.76 | 59.62 | 41.80 | 30.20 | 49.80 | 31.69 |
| | FGFI [36] | 35.44 | 55.51 | 38.17 | 39.44 | 60.27 | 43.04 | 31.16 | 50.68 | 32.92 |
| | ReviewKD [3] | 36.75 | 56.72 | 34.00 | 40.36 | 60.97 | 44.08 | 33.71 | 53.15 | 36.13 |
| Logits | KD [9] | 33.97 | 54.66 | 36.62 | 38.35 | 59.41 | 41.71 | 30.13 | 50.28 | 31.35 |
| | DIST [12] | 34.89 | 56.32 | 37.68 | 39.24 | 60.82 | 42.77 | 31.98 | 52.33 | 34.02 |
| | DKD [44] | 35.05 | 56.60 | 37.54 | 39.25 | 60.90 | 42.73 | 32.34 | 53.77 | 34.01 |
| | **GLD** (Ours) | **37.03** | **57.86** | **37.89** | **40.65** | **61.97** | **44.68** | **33.87** | **54.07** | **35.31** |
| | **GLD++** (Ours) | **37.66** | **57.99** | **38.22** | **41.18** | **62.45** | **44.88** | **34.01** | **54.62** | **35.58** |

**Figure 4**: The penultimate layer visualization of ResNet-8×4 with KD (left), GLD (middle) and GLD++ (right) on CIFAR-100.

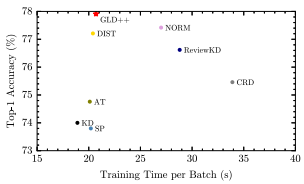


**Figure 5**: Training time (per batch) vs. accuracy on CIFAR-100 for distillation methods (ResNet-32×4/ShuffleV1 pair).

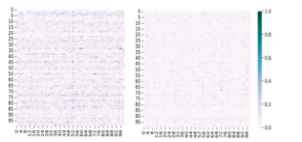

**Figure 6**: Visualization of correlation difference between the student and the teacher logits. GLD++ (right) compare with KD (left).

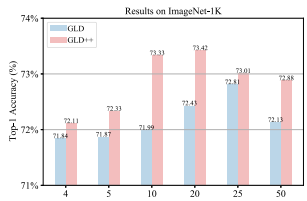


**Figure 7**: Top-1 prediction accuracy with different number of groups (ResNet18/ResNet34) on ImageNet-1k.

**Table 6**: Experiment of feature transfer by using the representation learned from CIFAR-100 to STL-10 and TinyImageNet datasets. We freeze the network and train a linear classifier on top of the last feature layer to perform a 10-way (STL-10) or 200-way (TinyImageNet) classification. The combination of teacher network ResNet-32×4 and student network ResNet-8×4 is employed.

| | Student | KD | AT | FitNet | DIST | GLD | GLD++ | Teacher |
|---|---|---|---|---|---|---|---|---|
| CIFAR100→STL-10 | 71.33 | 73.01 | 73.67 | 73.12 | 75.12 | 76.46 | **76.98** | 70.60 |
| CIFAR100→TinyImageNet | 35.10 | 35.39 | 35.42 | 35.55 | 37.13 | 38.57 | **39.92** | 34.20 |

**Table 7**: Ablation study on the proposed losses on CIFAR-100. The baseline denotes the plain student training without any KD. In other cases, the knowledge from pre-trained ResNet-32×4 is used for distillation.

| Module | Distillation $KD$ | Distillation $\mathcal{L}_{intra}$ | Distillation $\mathcal{L}_{inter}$ | ResNet-8×4 | ShuffleV1 |
|---|---|---|---|---|---|
| Baseline | - | - | - | 72.51 | 70.50 |
| KD | ✓ | - | - | 74.78 | 75.22 |
| GLD | ✓ | ✓ | - | 75.74 | 75.98 |
| | ✓ | - | ✓ | 75.32 | 75.61 |
| | ✓ | ✓ | ✓ | **77.22** | **77.63** |
| GLD++ | ✓ | ✓ | - | 76.22 | 76.41 |
| | ✓ | - | ✓ | 75.81 | 75.76 |
| | ✓ | ✓ | ✓ | **77.81** | **77.91** |

**Table 8**: The compatibility of our paradigm with current mainstream distillation methods. We perform the evaluation on CIFAR-100. 'Method', 'Method+', 'Method++' present the original method, incorporating grouped distillation, and extra superclass clustering to the method, respectively.

| Method | ResNet-32×4 | | VGG13 | |
|---|---|---|---|---|
| | ResNet-8×4 | ShuffleV1 | VGG-8 | MobileV2 |
| DIST | 76.55 | 77.21 | 74.71 | 69.81 |
| DIST+ | 76.91 | 77.53 | 75.42 | 70.08 |
| DIST++ | 77.32 | 77.78 | 75.78 | 70.34 |
| DKD | 76.32 | 77.42 | 74.68 | 69.71 |
| DKD+ | 76.42 | 77.51 | 74.71 | 69.98 |
| DKD++ | 76.88 | 77.87 | 75.56 | 70.65 |

**Correlation Difference.** We also visualize the heatmaps of correlation differences for baseline KD and our GLD++. We choose ResNet-8×4/ResNet-32×4 as the distillation pair. From Fig. 6, one can notice that our GLD method helps the student to predict more similar logits with the teacher compared with KD, achieving better distillation performance.

**The Number of Groups.** We systematically analyze the impact of group quantity adjustments within our framework, providing insights into the search of the best group number hyper-parammeter for the optimal prediction. Fig. 7 shows that the best performance occurs at a moderate level, we believe that this principle may benefit the hyper-parameter search of our paradigm in real-world scenarios.

## 6 Conclusion

In this paper, we proposed new distillation methods, named Grouped Logit Distillation (GLD) and its enhanced variant, Superclass-Aware Grouped Logit Distillation (GLD++). The two methods utilize grouping mechanism to mitigate the effect of skewed logit distributions in traditional methods, a primary problem that lags the efficacy of KD. In our enhanced variant GLD++, we also incorporate superclass information to our grouped distillation, thus facilitating a more nuanced and effective knowledge transferring. Our extensive experimental evaluation confirmed that GLD and GLD++ not only achieve superior performance compared to existing methods but also maintain computational efficiency, making them highly suitable for deployment in resource-constrained environments. We contend that the insights into leveraging hierarchical relationships within class groupings in this study would pave the way for more sophisticated distillation techniques in the future, enhancing the utility and applicability of deep learning models in real-world applications. Furthermore, our experiments also witnessed that GLD++ effectively leverages the robust feature representation capabilities of larger models. This efficiency makes our method a viable distillation option in the foundation model era.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. U22B2017). Yuyi Wang is supported by the Natural Science Foundation of Hunan Province, China (Grant No. 2024JJ5128).

## References

[1] J. Burkardt. K-means clustering. *Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics*, 2009.

[2] D. Chen, J.-P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen. Cross-layer distillation with semantic calibration, 2021.

[3] P. Chen, S. Liu, H. Zhao, and J. Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2021.

[4] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[5] S. Gao, F. Huang, W. Cai, and H. Huang. Network pruning via performance maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9270–9280, 2021.

[6] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[7] D. Guiming, W. Xia, W. Guangyan, Z. Yan, and L. Dan. Speech recognition based on convolutional neural networks. In *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, pages 708–711. IEEE, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[12] T. Huang, S. You, F. Wang, C. Qian, and C. Xu. Knowledge distillation from a stronger teacher. *Advances in Neural Information Processing Systems*, 35:33716–33727, 2022.

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

[14] Y. Jin, J. Wang, and D. Lin. Multi-level logit distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24276–24285, 2023.

[15] J. M. Joyce. Kullback-leibler divergence. In *International Encyclopedia of Statistical Science*, pages 720–722. Springer, 2011.

[16] U. Kamath, J. Liu, and J. Whitaker. *Deep learning for NLP and speech recognition*, volume 84. Springer, 2019.

[17] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014.

[18] J. Kim, S. Park, and N. Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems*, 2018.

[19] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 491–507. Springer, 2020.

[20] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[21] Y. Le and X. S. Yang. Tiny imagenet visual recognition challenge. 2015.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324, 1998.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[24] X. Liu, L. Li, C. Li, and A. Yao. Norm: Knowledge distillation via n-to-one representation matching. *arXiv preprint arXiv:2305.13803*, 2023.

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[26] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE access*, 7:19143–19165, 2019.

[27] F. Nielsen and F. Nielsen. Hierarchical clustering. *Introduction to HPC with MPI for Data Science*, pages 195–211, 2016.

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

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

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 211–252, 2015.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[32] Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. *arXiv:1910.10699*, 2019.

[33] F. Tung and G. Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[34] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018.

[35] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 1451–1460. Ieee, 2018.

[36] T. Wang, L. Yuan, X. Zhang, and J. Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4933–4942, 2019.

[37] T. Wen, S. Lai, and X. Qian. Preparing lessons: Improve knowledge distillation with better supervision. *Neurocomputing*, pages 25–33, 2021.

[38] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[39] Y. Yu, X. Si, C. Hu, and J. Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31 (7):1235–1270, 2019.

[40] S. Yun, J. Park, K. Lee, and J. Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 13876–13885, 2020.

[41] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv:1612.03928*, 2016.

[42] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv:1605.07146*, 2016.

[43] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018.

[44] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11953–11962, 2022.

[45] Y. Zhou, S.-M. Moosavi-Dezfooli, N.-M. Cheung, and P. Frossard. Adaptive quantization for deep neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.