Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks

Lin Wang, *Student Member, IEEE,* and Kuk-Jin Yoon, *Member, IEEE*

Abstract—Deep neural models, in recent years, have been successful in almost every field, even solving the most complex problem statements. However, these models are huge in size with millions (and even billions) of parameters, demanding heavy computation power and failing to be deployed on edge devices. Besides, the performance boost is highly dependent on redundant labeled data. To achieve faster speeds and to handle the problems caused by the lack of labeled data, knowledge distillation (KD) has been proposed to transfer information learned from one model to another. KD is often characterized by the so-called 'Student-Teacher' (S-T) learning framework and has been broadly applied in model compression and knowledge transfer. This paper is about KD and S-T learning, which are being actively studied in recent years. First, we aim to provide explanations of what KD is and how/why it works. Then, we provide a comprehensive survey on the recent progress of KD methods together with S-T frameworks typically used for vision tasks. In general, we investigate some fundamental questions that have been driving this research area and thoroughly generalize the research progress and technical details. Additionally, we systematically analyze the research status of KD in vision applications. Finally, we discuss the potentials and open challenges of existing methods and prospect the future directions of KD and S-T learning.

Index Terms—Knowledge distillation (KD), Student-Teacher learning (S-T), Deep neural networks (DNN), Visual intelligence.

✦

1 INTRODUCTION

THE success of deep neural networks (DNNs) generally depends on the elaborate design of DNN architectures. In large-scale machine learning, especially for tasks such as image and speech recognition, most DNN-based models are over-parameterized to extract the most salient features and to ensure generalization. Such cumbersome models are usually very deep and wide, which require a considerable amount of computation for training and are difficult to be operated in real-time. Thus, to achieve faster speeds, many researchers have been trying to utilize the cumbersome models that are trained to obtain lightweight DNN models, which can be deployed in edge devices. That is, when the cumbersome model has been trained, it can be used to learn a small model that is more suitable for real-time applications or deployment [1] as depicted in Fig. 1(a).

On the other hand, the performance of DNNs is also heavily dependent on very large and high-quality labels to training datasets. For such a reason, many endeavours have been taken to reduce the amount of labeled training data without hurting too much the performance of DNNs. A popular approach for handling such a lack of data is to *transfer knowledge* from one source task to facilitate the learning on the target task. One typical example is semi-supervised learning in which a model is trained with only a small set of labeled data and a large set of unlabeled data. Since the supervised cost is undefined for the unlabeled examples,

it is crucial to apply consistency costs or regularization methods to match the predictions from both labeled and unlabeled data. In this case, knowledge is transferred within the model that assumes a dual role as *teacher* and *student* [2]. For the unlabeled data, the student learns as before; however, the teacher generates targets, which are then used by the student for learning. The common goal of such a learning metric is to form a better teacher model from the student without additional training, as shown in Fig. 1(b). Another typical example is self-supervised learning, where the model is trained with artificial labels constructed by the input transformations (*e.g.*, rotation, flipping, color change, cropping). In such a situation, the knowledge from the input transformations is transferred to supervise the model itself to improve its performance as illustrated in Fig. 1(c).

This paper is about *knowledge distillation (KD) and student-teacher (S-T) learning*, a topic that has been actively studied in recent years. Generally speaking, KD is widely regarded as a primary mechanism that enables humans to quickly learn new complex concepts when given only small training sets with the same or different categories [3]. In deep learning, KD is an effective technique that has been widely used to transfer information from one network to another network whilst training constructively. KD was first defined by [4] and generalized by Hinton *et al.* [1]. KD has been broadly applied to two distinct fields: model compression (refer to Fig. 1(a)) and knowledge transfer (refer to Fig. 1 (b) and (c)). For model compression, a smaller student model is trained to mimic a pretrained larger model or an ensemble of models. Although various forms of knowledge are defined based on the purpose, one common characteristic of KD is symbolized by its *S-T framework*, where the model providing knowledge is called the teacher and the model learning the

---

• *L. Wang and K.-J. Yoon are with the Visual Intelligence Lab., Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Guseong-dong, Yuseong-gu, Daejeon 34141, Republic of Korea. E-mail: {wanglin, kjyoon }@kaist.ac.kr.*
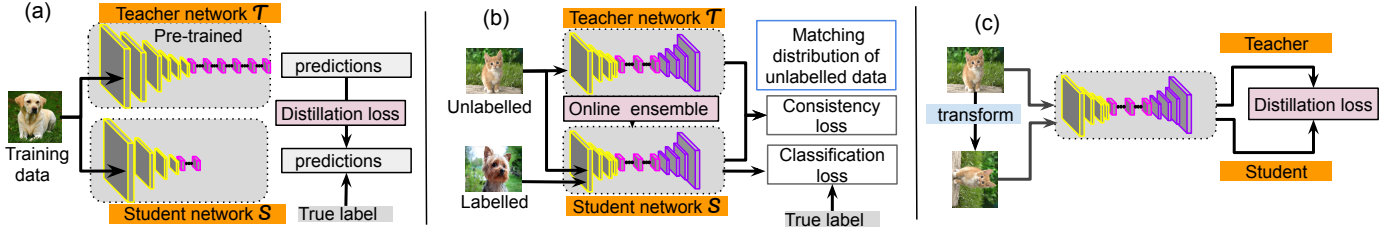
Fig. 1. Illustrations of KD methods with S-T frameworks. (a) for model compression and for knowledge transfer, *e.g.*, (b) semi-supervised learning and (c) self-supervised learning.

knowledge is called the student.

In this work, we focus on analyzing and categorizing existing KD methods accompanied by various types of S-T structures for model compression and knowledge transfer. We review and survey this rapidly developing area with particular emphasis on the recent progress. Although KD has been applied to various fields, such as visual intelligence, speech recognition, natural language processing (NLP), etc., this paper mainly focuses on the KD methods in the vision field, as most demonstrations have been done on computer vision tasks. KD methods used in NLP and speech recognition can be conveniently explained using the KD prototypes in vision. As the most studied KD methods are for model compression, we systematically discuss the technical details, challenges, and potentials. Meanwhile, we also concentrate on the KD methods for knowledge transfer in semi-supervised learning, self-supervised learning, etc., and we highlight the techniques that take S-T learning as a way of learning metric.

We explore some fundamental questions that have been driving this research area. Specifically, what is the theoretical principle for KD and S-T learning? What makes one distillation method better than others? Is using multiple teachers better than using one teacher? Do larger models always make better teachers and teach more robust students? Can a student learn knowledge only if a teacher model exists? Is the student able to learn by itself? Is off-line KD always better than online learning?

With these questions being discussed, we incorporate the potentials of existing KD methods and prospect the future directions of the KD methods together with S-T frameworks. We especially stress the importance of recently developed technologies, such as neural architecture search (NAS), graph neural networks (GNNs), and gating mechanisms for empowering KD. Furthermore, we also emphasize the potential of KD methods for tackling challenging problems in particular vision fields such as $360°$ vision and event-based vision.

The main contributions of this paper are three-fold:

- We give a comprehensive overview of KD and S-T learning methods, including problem definition, theoretical analysis, a family of KD methods with deep learning, and vision applications.
- We provide a systematic overview and analysis of recent advances of KD methods and S-T frameworks hierarchically and structurally and offer insights and summaries for the potentials and challenges of each category.

- We discuss the problems and open issues and identify new trends and future direction to provide insightful guidance in this research area.

The organization of this paper is as follows. First, we explain why we need to care about KD and S-T learning in Sec.2. Then, we provide a theoretical analysis of KD in Sec.3. Section 3 is followed by Sec.4 to Sec.8, where we categorize the existing methods and analyze their challenges and potential. Fig. 2 shows the taxonomy of KD with S-T learning to be covered in this survey in a hierarchically-structured way. In Sec.9, based on the taxonomy, we will discuss the answers to the questions raised in Sec.1. Section 10 will present the future potentials of KD and S-T learning, followed by a conclusion in Sec.11.

2 WHAT IS KD AND WHY CONCERN IT?

What's KD? Knowledge Distillation (KD) was first proposed by [4] and expanded by [1]. KD refers to the method that helps the training process of a smaller student network under the supervision of a larger teacher network. Unlike other compression methods, KD can downsize a network regardless of the structural difference between the teacher and the student network. In [1], the knowledge is transferred from the teacher model to the student by minimizing the difference between the logits (the inputs to the final softmax) produced by the teacher model and those produced by the student model.

However, in many situations, the output of softmax function on the teacher's logits has the correct class at a very high probability, with all other class probabilities very close to zero. In such a circumstance, it does not provide much information beyond the ground truth labels already provided in the dataset. To tackle such a problem, [1, 5] introduced the concept of 'softmax temperature', which can make the target to be 'soft.' Given the logits z from a network, the class probability p_i of an image is calculated as [5]:

$$p_i = \frac{\exp(\frac{z_i}{\rho})}{\sum_j \exp(\frac{z_i}{\rho})} \quad (1)$$

where ρ is the temperature parameter. When $\rho = 1$, we get the standard softmax function. As ρ increases, the probability distribution produced by the softmax function becomes softer, providing more information as to which classes the teacher found more similar to the predicted class. The information provided in the teacher model is called *dark knowledge* [1]. It is this dark knowledge that affects the overall flow of information to be distilled. When computing
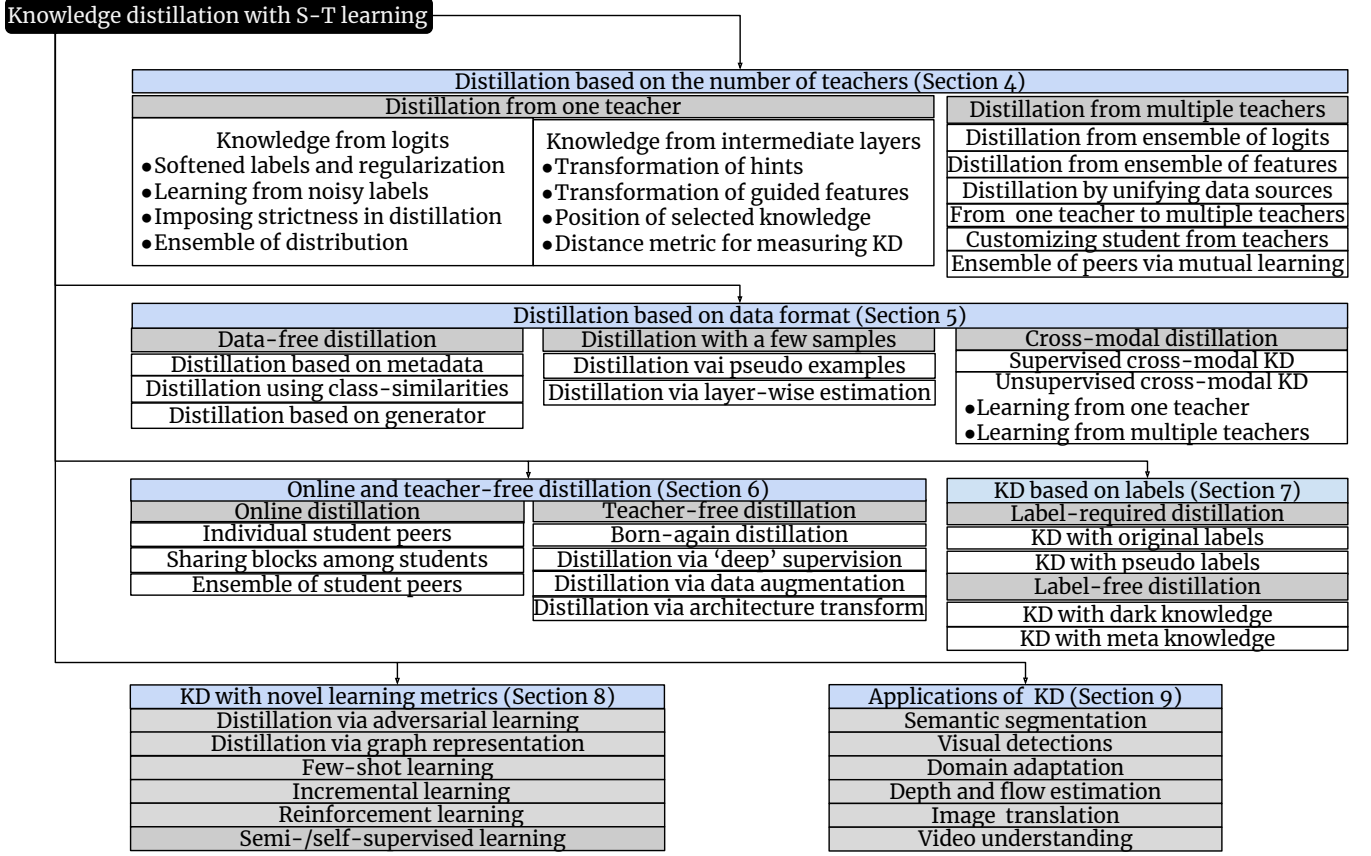
Knowledge distillation with S-T learning	

Distillation based on the number of teachers (Section 4)

Distillation from one teacher		Distillation from multiple teachers
Knowledge from logits •Softened labels and regularization •Learning from noisy labels •Imposing strictness in distillation •Ensemble of distribution	Knowledge from intermediate layers •Transformation of hints •Transformation of guided features •Position of selected knowledge •Distance metric for measuring KD	Distillation from ensemble of logits Distillation from ensemble of features Distillation by unifying data sources From one teacher to multiple teachers Customizing student from teachers Ensemble of peers via mutual learning

Distillation based on data format (Section 5)

Data-free distillation	Distillation with a few samples	Cross-modal distillation
Distillation based on metadata Distillation using class-similarities Distillation based on generator	Distillation vai pseudo examples Distillation via layer-wise estimation	Supervised cross-modal KD Unsupervised cross-modal KD •Learning from one teacher •Learning from multiple teachers

Online and teacher-free distillation (Section 6)

Online distillation	Teacher-free distillation
Individual student peers Sharing blocks among students Ensemble of student peers	Born-again distillation Distillation via 'deep' supervision Distillation via data augmentation Distillation via architecture transform

KD based on labels (Section 7)

Label-required distillation
KD with original labels
KD with pseudo labels
Label-free distillation
KD with dark knowledge
KD with meta knowledge

KD with novel learning metrics (Section 8)

Distillation via adversarial learning
Distillation via graph representation
Few-shot learning
Incremental learning
Reinforcement learning
Semi-/self-supervised learning

Applications of KD (Section 9)

Semantic segmentation
Visual detections
Domain adaptation
Depth and flow estimation
Image translation
Video understanding

Fig. 2. Hierarchically-structured taxonomy of knowledge distillation with S-T learning.

the distillation loss, the same ρ used in the teacher is used to compute the logits of the student.

For the images with ground truth, [1] stated that it is beneficial to train the student model together with the ground truth labels in addition to the teacher's soft labels. Therefore, we also calculate the 'student loss' between the student's predicted class probabilities and the ground truth labels. The overall loss function, composed of the student loss and the distillation loss, is calculated as:

$$\mathcal{L}_{KD} = \alpha * H(y, \sigma(z_s)) + \beta * H(\sigma(z_t; \rho), \sigma(z_s; \rho))$$
$$= \alpha * H(y, \sigma(z_s) + \beta * [KL(\sigma(z_t; \rho), \sigma(z_s, \rho)) + H(\sigma(z_t))]$$
(2)

where H is the loss function, y is the ground truth label, σ is the softmax function parameterized by the temperature ρ ($\rho \neq 1$ for distillation loss), and α and β are coefficients. z_s and z_t are the logits of the student and teacher respectively. **Why concern KD?** KD has become a field in itself in the machine learning community, with broad applications to computer vision, speech recognition, NLP, etc. From 2014 to now, many research papers [6, 7] have been presented in the major conferences, such as CVPR, ICCV, ECCV, NIPS, ICML, ICLR, etc., and the power of KD has been extended to many learning processes (*e.g.*, few-shot learning) except to model compression. The trend in recent years is that KD with S-T frameworks has become a crucial tool for knowledge transfer, along with model compression. The rapid increase in scientific activity on KD has been accompanied and nourished by a remarkable string of empirical successes

both in academia and industry. The particular highlights on some representative applications are given in Sec.9, and in the following Sec.3, we provide a systematic theoretical analysis.

3 A THEORETICAL ANALYSIS OF KD

Many KD methods have been proposed with various intuitions. However, there is no commonly agreed theory as to how knowledge is transferred, thus making it difficult to effectively evaluate the empirical results and less actionable to design new methods in a more disciplined way. Recently, Ahn *et al.* [8], Hegde *et al.* [9] and Tian *et al.* [10] formulate KD as a maximization of mutual information between the representations of the teacher and the student networks. Note that the representations here can refer to either the logits information or the intermediate features. From the perspective of representation learning and information theory, the mutual information reflects the joint distribution or mutual dependence between the teacher and the student and quantifies how much information is transferred. We do agree that maximizing the mutual information between the teacher and the student is crucial for learning constructive knowledge from the teacher. We now give a more detailed explanation regarding this.

Based on Bayes's rule, the mutual information between two paired representations can be defined as:

$$I(T; S) = H(R(T)) - H(R(T)|R(S))$$
$$= -\mathbb{E}_T[\log p(R(T))] + \mathbb{E}_{T,S}[\log p((R(T)|R(S))]$$
(3)

where $R(T)$ and $R(S)$ are the representations from both the teacher and the student, and $H(\cdot)$ is the entropy function. Intuitively, the mutual information is to increase degree of certainty in the information provided in $R(T)$ when $R(S)$ is known. Therefore, maximizing $\mathbb{E}_{T,S}[\log p((R(T)|R(S)))]$ w.r.t. the parameters of the student network S increases the lower bound on mutual information. However, the true distribution of $p((R(T)|R(S)))$ is unknown, instead it is desirable to estimate $p((R(T)|R(S)))$ by fitting a variations distribution $q((R(T)|R(S)))$ to approximate the true distribution $p((R(T)|R(S)))$. Then Eqn. 3 can be rewritten as:

$$
\begin{aligned}
I(T;S) &= H(R(T)) + \mathbb{E}_{T,S}[\log p(R(T)|R(S))] \\
&= H(R(T)) + \mathbb{E}_{T,S}[\log q((R(T)|R(S)))] + \\
&\quad \mathbb{E}_S[KL(p(R(T)|R(S))||q(R(T)|R(S)))]
\end{aligned}
\tag{4}
$$

Assuming there is sufficiently expressive way of modeling q, Eqn. 4 can be updated as:

$$
I(T;S) \geq H(R(T)) + \mathbb{E}_{T,S}[\log q((R(T)|R(S)))]
\tag{5}
$$

Note that the last term in Eqn. 4 is non-negative since $KL(\cdot)$ function is non-negative and $H(R(T))$ is constant w.r.t the parameters to be optimized. By modeling q, it is easy to quantify the amount of knowledge being learned by the student. In general, q can be modeled by a Gaussian distribution, Monte Carlo approximation, or noise contrastive estimation (NCE). We do believe that theoretically explaining how KD works is connected to representation learning, where the correlations and higher-order output dependencies between the teacher and the student are captured. The critical challenge is increasing the lower bounds of information, which is also pointed out in [10].

In summary, we have theoretically analyzed how KD works and mentioned that the *representation* of knowledge is crucial for the transfer of knowledge and learning of the student network. Explicitly dealing with the representation of knowledge from the teacher is significant and challenging, because the knowledge from the teacher expresses a more general learned information (*e.g.*feature information, logits, data usage, etc.) that is helpful for building up a well-performing student. In the following sections, we will provide a hierarchically-structured taxonomy for the KD methods regarding how the information is transferred for both teacher and student, how knowledge is measured, and how the teacher is defined.

4 KD BASED ON THE NUMBER OF TEACHERS

4.1 Distillation from one teacher

Overall insight: *Transferring knowledge from a large teacher network to a smaller student network can be achieved using either the logits or feature information from the teacher.*

4.1.1 Knowledge from logits

Softened labels and regularization. Hinton *et al.* [1] and Ba and Caruana [11] propose to shift the knowledge from teacher network to student network by learning the class distribution via softened softmax (also called 'soft labels') given in Eqn. (1). The softened labels are in fact achieved by introducing temperature scaling to increase of small probabilities. These KD methods achieved some surprising results on vision and speech recognition tasks. Recently, Mangalam *et al.* [12] introduce a special method based on class re-weighting to compress U-net into a smaller version. Re-weighting of the classes, in fact, softens the label distribution by obstructing inherent class imbalance. Compared to [1], some works such as Ding *et al.* [13], Hegde *et al.* [9], Tian *et al.* [10], Cho *et al.* [14] and Wen *et al.* [15], point out that the trade-off (see Eqn. 2) between the soft label and the hard label is rarely optimal, and since α, β and T are fixed during training time, it lacks enough flexibility to cope with the situation without the given softened labels. Ding *et al.* [13] instead propose *residual label* and *residual loss* to enable the student to use the erroneous experience during the training phase, preventing over-fitting and improving the performance. Similarly, Tian *et al.* [10] formulate the teacher's knowledge as structured knowledge and train a student to capture significantly more *mutual information* during contrastive learning. Hegde *et al.* [9] propose to train a *variational* student by adding sparsity regularizer based on variational inference, similar to the method in [8]. The sparsification of the student training reduces over-fitting and improves the accuracy of classification. Wen *et al.* [15] notice that the knowledge from the teacher is useful, but uncertain supervision also influences the result. Therefore, they propose to fix the incorrect predictions (knowledge) of the teacher via smooth regularization and avoid overly uncertain supervision using dynamic temperature.

On the other hand, Cho *et al.* [16], Yang *et al.* [17] and Liu *et al.* [18] focus on different perspectives of regularization to avoid under-/over-fitting. Cho *et al.* [16] discover that early-stopped teacher makes a better student especially when the capacity of the teacher is larger than the student's. Stopping the training of the teacher early is akin to regularizing the teacher, and stopping knowledge distillation close to convergence allows the student to fit the training better. Liu *et al.* [18] focus on modeling the distribution of the parameters as prior knowledge, which is modeled by aggregating the distribution (logits) space from the teacher network. Then the prior knowledge is penalized by a sparse recording penalty for constraining the student to avoid over-regularization. Mishra *et al.* [19] combine network quantization with model compression by training an apprentice using KD techniques and showed that the performance of low-precision networks could be significantly improved by distilling the logits of the teacher network. Yang *et al.* [17] propose a snapshot distillation method to perform S-T (similar network architecture) optimization in *one generation*. Their method is based on a cycle learning rate policy (refer to Eqn. 2 and Eqn. 6) in which the last snapshot of each cycle (*e.g.*,W_T^{l-1} in iteration $l-1$) serves as a teacher in the next cycle (*e.g.*, W_T^l in iteration l). Thus, the idea of snapshot distillation is to extract supervision signals in earlier epochs in the same generation to make sure the difference between teacher and student is sufficiently large to avoid underfitting. The snapshot distillation loss can be described as:

$$
\begin{aligned}
\mathcal{L}(x, W_{l-1}) &= \alpha * H(y, \sigma(z_s^{l-1}; \rho = 1)) + \\
&\quad \beta * H(\sigma(z_t^l; \rho = \tau), \sigma(z_t^{l-1}, \rho = \tau))
\end{aligned}
\tag{6}
$$

where the W_{l-1} is the weights of student at iteration $l-1$. z_s^{l-1} and z_t^{l-1} represent the logits of student and teacher

at iteration $l-1$. *More detailed analysis for the methods with mutual information and one generation will be discussed in Sec. 6.1.*

Learning from noisy labels [20, 21, 22, 23] propose methods that utilize the similar knowledge (softened labels) as in [1] but focus on data issue. Specifically, [20] assume that there is a small clean dataset D_c and a large noisy dataset D_n, while [21] and [22] use both labeled and unlabeled data to improve the performance of student. In [20], the aim of distillation is to use the large amount of noisy data D_n to augment the small clean dataset D_c to learn a better visual representation and classifier. That is, the knowledge is distilled from the small clean dataset D_c to facilitate a better model from the entire noisy dataset D_n. The method is essentially different from [1] focusing on inferior model instead of inferior dataset. The same loss function in Eqn. 2 is used, except $z_t = \sigma[f_{D_c}(x)]$, where f_{D_c} is an auxiliary model trained from the clean dataset D_c. Furthermore, a risk function on the unreliable label $\bar{y}$ is defined as $R_{\bar{y}} = \mathbb{E}_{D_t}[||\bar{y} - y^*||]^2$, where y^* is the unknown ground truth label and D_t is the unseen test dataset. $R_{\bar{y}}$ is an indicator that measures the level of noise in the distillation process.

Xu *et al.* [22] probes a positive-unlabeled classifier for addressing the problem of requesting the entire original training data, which can not be easily uploaded to the cloud. [21] trains a *noisy* student by next three steps: 1) train a teacher model on labeled data, 2) use the teacher to generate pseudo labels on unlabeled images, and 3) train a student model on the combination of labeled images and pseudo labeled images while injecting noise (adversarial perturbation) to the student for better generalization and robustness. This way, the student generalizes better than the teacher. Similarly, [23] study adversarial perturbation and consider it as a crucial element in improving both the generalization and the robustness of the student. Based on how humans learn, two learning theories for the S-T model are proposed: fickle teacher and soft randomization. The fickle teacher model is to transfer the teacher's uncertainty to the student using Dropout [24] in the teacher model. The soft randomization method is to improve the adversarial robustness of student model by adding Gaussian noise in the knowledge distillation. In this setting, the original distillation objective for the student in Eqn. 2 can be updated as:

$$\mathcal{L}(x+\delta, W) = \alpha * H(y, \sigma(z_s; \rho = 1)) + \beta * H(\sigma(z_t; \rho = \tau), \sigma(z_s, \rho = \tau)) \quad (7)$$

where δ is the variation of adversarial perturbation. It is shown that using the teacher model trained on clean images to train the student model with adversarial perturbation can retain the adversarial robustness and mitigate the loss in generalization.

Imposing strictness in distillation. In contrast, Yang *et al.* [25], Yu *et al.* [26], Arora *et al.* [27], RKD [28] and Peng *et al.* [29] shift to a new perspective focusing more on putting *strictness* to the distillation process via optimization (*e.g.,* distribution embedding, mutual relations, etc). In particular, [25] initiates to put strictness on the teacher while [26] proposes two teaching metrics to impose strictness on the student. Yang *et al.* observe that, except learning *primary class* (namely, the ground truth), learning *secondary class* (

high confidence scores in the *dark knowledge* in [1]) may help to alleviate the risk of the student over-fitting. They thus introduce a framework of optimizing neural networks in *generations* (namely, iterations), which requires training a patriarchal model M^0 only supervised by the dataset. After m generations, the student M^m is trained by m-th generation with the supervision of a teacher M^{m-1}. Since the secondary information is crucial for training a robust teacher, a fixed integer K standing for the semantically similar class is chosen for each image, and the gap between the confidence scores of the primary class and other $K-1$ classes with highest scores is computed, This can be described as:

$$\mathcal{L}(x, W^T) = \alpha * H(y, \sigma(z_t; \rho = 1)) + \beta * [f_{a_1}^T - \frac{1}{K-1} \sum_{k=2}^{K} f_{a_k}^T] \quad (8)$$

where f_{a_k} indicates the k-th largest elements of the output (logits) z_t. Note that this S-T optimization is similar to BAN [30]; however, the goal here is to help the student learn inter-class similarity and prevent over-fitting. Different from the teacher in [30], the teacher here is deeper and larger than the student. [26] extends [1] for metric learning by using embedding networks to project the information (logits) learned from images to the embedding space. The embeddings are typically used to perform distance computation between the data pairs of a teacher and a student. From this point of view, the knowledge computed based on the embedding network is the actual knowledge as it represents the data distribution. They design two different teachers: absolute teacher and relative teacher. For the absolute teacher, the aim is to minimize the distance between the teacher and student embeddings while the aim for the relative teacher is to enforce the student to learn any embedding as long as it results in a similar distance between the data points. They also explore hints [1] and attention [36] to strengthen the distillation of embedding networks. *We will give more explicit explanations of these two techniques in Sec. 4.1.2.*

[27] proposes an embedding module that captures interactions between query and document information for question answering. The embedding of the output representation (logits) includes a simple attention model with a query encoder, a prober history encoder, a responder history encoder, and a document encoder. The attention model minimizes the summation of cross-entropy loss and KL-divergence loss, inspired by [1]. On the other hand, [31] and RKD [28] consider another type of strictness, namely the *mutual relation or relation knowledge* of the two examples in the learned representations for both the teacher and the student. This approach is very similar to the relative teacher in [26] since both aim to measure the distance between the teacher's and the student's embeddings. However, RKD [28] also considers the angle-wise relational measure, similar to persevering secondary information in [25].

Ensemble of distribution. Although various methods have been proposed to extract knowledge from logits, some works [16, 32, 33, 34] show that KD is not always practical due to knowledge uncertainty. The performance of the student degrades when the gap between the student and the teacher is large. [33] points out that estimating

TABLE 1
A taxonomy of KD methods using logits. The given equations here are the generalized objective functions, and they may vary in individual work.

Method	Sub-category	Description	KD objective function	References
KD from logits	Softened labels and regularization	Distillation using soft labels and add regularizatio to avoid under-/over-fitting	Eqn. 6	[1, 9, 10, 11, 12, 14, 15] [8, 9, 15, 16, 17, 18, 19, 19]
	Learning from noisy labels	Adding noise or using noisy data	Eqn. 6 or Eqn. 7	[20, 21, 22, 23, 24]
	Imposing strictness	Adding optimization methods to teacher or student	Eqn. 8 or Eqn. 6	[25, 26, 27, 28, 29, 30, 31]
	Ensemble of distribution	Estimating model or data uncertainty	Eqn. 9	[16, 32, 33, 34, 35]
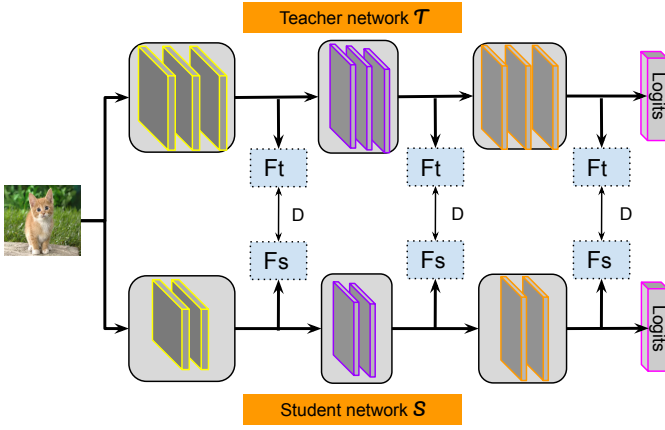
Fig. 3. An illustration of general feature-based distillation.

the model's uncertainty is crucial since it ensures a more reliable knowledge to be transferred. They stress on the ensemble approaches to estimate the data uncertainty and the distributional uncertainty. To estimate the distributional uncertainty, an ensemble distribution distillation approach anneals the temperature of the softmax to not only capture the mean of ensemble soft labels but also the diversity of the distribution. Meanwhile, [35] proposes a similar approach of matching the distribution of distillation-based multi-exit architectures, in which a sequence of feature layers is augmented with early exits at different depths. By doing so, the loss defined in Eqn. 2 becomes:

$$\mathcal{L}(x, W) = \frac{1}{K} \sum_{k=1}^{K} [\alpha * H(y, \sigma(p_s^k; \rho = 1) + \\ \beta * H(\sigma(p_t^k; \rho = \tau), \sigma(p_s^k, \rho = \tau))] \quad (9)$$

where K indicates the total number of exits, and p_s^k and p_t^k represent the k-th probabilistic output at exit k.

Conversely, [2, 27, 30, 32, 34, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50] propose to add more teachers or other auxiliaries, such as teaching assistant and small students, to improve the robustness of ensemble distribution. We will explicitly analyze these approaches in the following Sec. 4.2.

Summary. Table. 1 summarizes the KD methods that use logits or 'soft labels'. We divide these methods into four

categories. In overall, distillation using logits needs to transfer the dark knowledge to avoid over-/under-fitting. Meanwhile, the gap of model capacity between the teacher and the student is also very crucial for effective distillation. Moreover, the drawbacks of learning from logits are obvious. First, the effectiveness of distillation is limited to softmax loss and relies on the number of classes. Second, it is impossible to apply these methods to the KD problems in which there are no labels (*e.g.*, low-level vision).

Open challenges: The original idea in [1] is in its apparent generality: any student can learn from any teacher; however, it is shown that this promise of generality is hard to be achieved on some datasets [16, 36] (*e.g.*, ImageNet [51]) even when regularization or strictness techniques are applied. When the capacity of the student is too low, it is hard for the student to incorporate the logits information of the teacher successfully. Therefore, it is expected to improve the generality and provide a better representation of logits information, which can be easily absorbed by the student.

4.1.2 Knowledge from the intermediate layers

Overall insight: *Feature-based distillation enables learning richer information from the teacher and provides more flexibility for performance improvement.*

Apart from distilling knowledge from the softened labels, Romero *et al.* [52] initially introduce *hint* learning rooted from [1]. A hint is defined as the outputs of a teacher's hidden layer, which helps guide the student's learning process. The goal of student learning is to learn a feature representation that is the optimal prediction of the teacher's intermediate representations. Essentially, the function of hints is a form of regularization; therefore, a pair of hint and guided (a hidden layer of the student) layer has to be carefully chosen such that the student is not over-regularized. Inspired by [52], many endeavours have been taken to study the methods to choose, transport and match the hint layer (or layers) and the guided layer (or layers) via various layer transform (*e.g.*, transformer [53, 54]) and distance (*e.g.*, MMD [55]) metrics. Generally, the hint learning objective can be written as:

$$\mathcal{L}(F_T, F_S) = D(TF_t(F_T), TF_s(F_S)) \quad (10)$$

Where F_T and F_S are the selected hint and guided layers of teacher and student. TF_t and TF_s are the transformer

and regressor functions for the hint layer of teacher and guided layer of student. $D(\cdot)$ is the distance function(*e.g.*, l_2) measuring the similarity of the hint and the guided layers.

Fig. 3 depicts the general paradigm of feature-based distillation. It is shown that various intermediate feature representations can be extracted from different positions and are transformed with a certain type of regressor or transformer. The similarity of the transformed representations is finally optimized via an arbitrary distance metrics D (*e.g.*, L_1 or L_2 distance). In this paper, we carefully scrutinize various design considerations of feature-based KD methods and summarize four key factors that are often considered: *transformation of the hint, transform of the guided layer, position of the selected distillation feature, and distance metric* [53]. In the following parts, we will analyze and categorize all existing feature-based KD methods concerning these four aspects.

Transformation of hints As pointed in [8], the knowledge of teacher should be easy to learn as the student. To do this, teacher's hidden features are usually converted by a transformation function T_t. Note that the transformation of teacher's knowledge is a very crucial step for feature-based KD since there is a risk of losing information in the process of transformation. The transformation methods of teacher's knowledge in AT [36], MINILM [58], FSP [57], ASL[70], Jacobian [59], KP [56], SVD [71], SP [61], MEAL [62], KSANC [66], and NST [55] cause the knowledge to be missing due to the reduction of feature dimension. Specifically, AT [54] and MINILM [58] focus on attention mechanisms (*e.g.*, self-attention [72]) via an attention transformer T_t to transform the activation tensor $F \in \mathbb{R}^{C \times H \times W}$ to C feature maps $F \in \mathbb{R}^{H \times W}$. FSP [57] and ASL [70] calculate the information flow of the distillation based on the Gramian matrices, through which the tensor $F \in \mathbb{R}^{C \times H \times W}$ is transformed to $G \in \mathbb{R}^{C \times N}$, where N represents the number of matrices. Jacobian [59] and SVD [71] map the tensor $F \in \mathbb{R}^{C \times H \times W}$ to $G \in \mathbb{R}^{C \times N}$ based on Jacobians using first-order Taylor series and truncated SVD, respectively, inducing information loss. KP [56] projects $F \in \mathbb{R}^{C \times H \times W}$ to M feature maps $F \in \mathbb{R}^{M \times H \times W}$, causing loss of knowledge. Similarly, SP [61] proposes a similarity-preserving knowledge distillation method based on the observation that semantically similar inputs tend to elicit similar activation patterns. To achieve this goal, the teacher's feature $F \in \mathbb{R}^{B \times C \times H \times W}$ is transformed to $G \in \mathbb{R}^{B \times B}$, where B is the batch size. The G encodes the similarity of the activations at the teacher layer, but leads to an information loss during the transformation. MEAL [62] and KSANC [66] both use *pooling* to align the intermediate map of the teacher and student, leading to an information loss when transforming the teacher's knowledge. NST [55] and PKT [73] match the distributions of neuron selectivity patterns and the affinity of data samples between the teacher and the student networks. The loss functions are based on minimizing the maximum mean discrepancy (MMD) and Kullback-Leibler (KL) divergence between these distributions respectively, thus causing information loss when selecting neurons.

On the other hand, FT [54] proposes to extract good *factors* through which transportable features are made. The transformer TF_t is called the *paraphraser* and the transformer TF_s is called the *translator*. To extract the teacher factors, an adequately trained paraphraser is needed. Mean-while, to enable the student to assimilate and digest the knowledge according to its own capacity, a user-defined paraphrase ratio is used in the paraphraser to control the factor of the transfer. Heo *et al.* [63] use the original teacher's feature in the form of binarized values, namely via a separating hyperplane (activation boundary (AB)) that determines whether neurons are activated or deactivated. Since AB only considers the activation of neurons and not the magnitude of neuron response, there is information loss in the feature binarization process. Similar information loss happens in IRG [18], where the teacher's feature space is transformed to a graph representation with vertices and edges where the relationship matrices are calculated. IR [68] distills the internal representations of the teacher model to the student model. However, since multiple layers in the teacher model are compressed into one layer of the student model, there is information loss when matching the features. Heo *et al.* [53] design TF_t with a margin ReLU function to exclude the negative (adverse) information and to allow positive (beneficial) information. The margin m is determined based on batch normalization [74] after 1×1 convolution in the student's transformer TF_s.

Conversely, FitNet [52], RCO [60], Chung *et al.* [64], Wang *et al.* [65], Gao *et al.* [69] and Kulkarni *et al.* [67] do not add additional transformation to the teacher's knowledge; this leads to no information loss from teacher's side. However, not all knowledge from the teacher is beneficial for the student. As pointed by [53], features include both adverse and beneficial information. For effective distillation, it is important to impede the use of adverse information and to avoid missing the beneficial information.

Transformation of the guided features The transformation TF_s of the guided features (namely, student transform) of the student is also an important step for effective KD. Interestingly, the SOTA works such as AT [36], MINILM [58], FSP [57], Jacobian [59], FT [54], SVD [71], SP [61], KP [56], IRG [18], RCO [60],MEAL [62], KSANC [66], NST [55], Kulkarni *et al.* [67], Gao *et al.* [69] and Aguilar *et al.* [68] use the same TF_s as the TF_t, which means the same amount of information might be lost in both transformations of the teacher and the student.

Different from the transformation of teacher, FitNet [52], AB [63], Heo *et al.* [53], and VID [8] change the dimension of teacher's feature representations and design TF_s with a 'bottleneck' layer (1×1 convolution) to make the student's features match the dimension of the teacher's features. Note that Heo *et al.* [53] add a batch normalization layer after a 1×1 convolution to calculate the margin of the proposed margin ReLU transformer of the teacher. There are some advantages of using 1×1 convolution in KD. First, it offers a channel-wise pooling without a reduction of the spatial dimensionality. Second, it can be used to create a one-to-one linear projection of the stack of feature maps. Lastly, the projection created by 1×1 convolution can also be used to directly increase the number of feature maps in the distillation model. In such a case, the feature representation of student does not decrease but rather increase to match the teacher's representation; this does not cause information loss in the transformation of the student.

Exceptionally, some works focus on a different aspect of the transformation of student's feature representations.

TABLE 2
A taxonomy of knowledge distillation from the intermediate layers (feature maps). KP incidates knowledge projection.

Method	Teacher's TF_t	Student's TF_s	Distillation position	Distance metric	Lost knowledge
FitNet [52]	None	1×1 Conv	Middle layer	L_1	None
AT [36]	Attention map	Attention map	End of layer group	L_2	Channel dims
KP [56]	Projection matrix	Projection matrix	Middle layers	L_1 + KP loss	Spatial dims
FSP [57]	FSP matrix	FSP matrix	End of layer group	L_2	Spatial dims
FT [54]	Encoder-decoder	Encoder-decoder	End of layer group	L_1	Channel + Spatial dims
AT [36]	Attention map	Attention map	End of layer group	L_2	Channel dimensions
MINILM [58]	Self-ttention	Self-attention	End of layer group	KL	Channel dimensions
Jacobian [59]	Gradient penalty	Gradient penalty	End of layer group	L_2	Channel dims
SVD [57]	Truncated SVD	Truncated SVD	End of layer group	L_2	Spatial dims
VID [8]	None	1×1 Conv	Middle layers	KL	None
IRG [18]	Instance graph	Instance graph	Middle layers	L_2	Spatial dims
RCO [60]	None	None	Teacher's train route	L_2	None
SP [61]	Similarity matrix	Similarity matrix	Middle layer	Frobenius norm	None
MEAL [62]	Adaptive pooling	Adaptive pooling	End of layer group	$L_{1/2}$/KL/L_{GAN}	None
Heo [62]	Margin ReLU	1×1 Conv	Pre-ReLU	Partial L_2	Negative features
AB [63]	Binarization	1×1 Conv	Pre-ReLU	Margin L_2	feature values
Chung [64]	None	None	End of layer	L_{GAN}	None
Wang [65]	None	Adaptation layer	Middle layer	Margin L_1	Channel + Spatial dims
KSANC [66]	Average pooling	Average pooling	Middle layers	$L_2 + L_{GAN}$	Spatial dims
Kulkarni [67]	None	None	End of layer group	L_2	None
IR [68]	Attention matrix	Attention matrix	Middle layers	KL+ Cosine	None
Liu [18]	Transform matrix	Transform matrix	Middle layers	KL	Spatial dims
NST [55]	None	None	Intermediate layers	MMD	None
Gao [69]	None	None	Intermediate layers	L_2	None

Wang *et al.* [65] make the student imitate the fine-grained local feature regions close to object instances of the teacher's representations. This is achieved by designing a particular adaptation function TF_s to fulfill the imitation task. IR [68] aims to let the student acquire the abstraction in a hidden layer of the teacher by matching the internal representations. That is, the student is taught to know how to compress the knowledge from multiple layers of the teacher into a single layer. In such a setting, the transformation of the student's guided layer is done by a self-attention transformer. Chung *et al.* [64], on the other hand, propose to impose no transformation to both student and teacher, but rather add a discriminator to distinguish the feature map distributions of different networks (teacher or student).

Distillation positions of features In addition to the transformation of teacher's and student's features, distillation position of the selected features is also very crucial in many cases. Earlier, FitNet [52], AB [63], and Wang *et al.* [65] use the end of an arbitrary middle layer as the distillation point. However, this method is shown to have poor distillation performance. Based on the definition of layer group [75], in which a group of layers have same spatial size, AT [36], FSP [57], Jacobian [59], MEAL [62], KSANC [66], Gao *et al.* [69] and Kulkarni *et al.* [67] define the distillation position at the end of each layer group, in contrast to FT [54] and NST [55] where the position lies only at the end of last layer group. Compared to FitNet, FT achieves better results since it focuses more on informational knowledge. IRG [18] considers all the above-mentioned critical positions; namely, the distillation position lies not only in the end of earlier layer group but also in the end of the last layer group. Interestingly, VID [8], RCO [60], Chung *et al.* [64], SP [61], IR [68], and Liu *et al.* [18] generalize the selection of distillation

positions by employing variational information maximization [76], curriculum learning [77], adversarial learning [78], similarity-presentation in representation learning [79], muti-task learning [80], and reinforcement learning [81]. We will discuss more for these methods in later sections.

Distance metric for measuring distillation The quality of KD from teacher to student is usually measured by various distance metrics. The most commonly used distance function is based on L_1 or L_2 distance. FitNet [52], AT [36], NST [55], FSP [57], SVD [71], RCO [60], FT [54], KSANC [66], Gao *et al.* [69] and Kulkarni *et al.* [67] are mainly based on L_2 distance, whereas MEAL [62] and Wang *et al.* [65] mainly use L_1 distance. On the other hand, Liu *et al.* [18] and IR [68] utilize KL-divergence loss to measure feature similarities. Furthermore, a cosine-similarity loss is adopted by IR [68] and RKD [28] to regularize the context representation on the feature distributions of teacher and student.

Some works also resort to the adversarial loss for measuring the quality of KD. MEAL [62] shows that the student learning the distilled knowledge with discriminators is better optimized than the original model, and the student can learn distilled knowledge from a teacher model with arbitrary structures. Among the works focusing on feature-based distillation, KSANC [60] adds an adversarial loss at the last layer of both teacher and student networks, while MEAL adds multi-stage discriminators in the position of every extracted feature representation. It is worth mentioning that using adversarial loss has shown considerable potential in improving the performance of KD. We will explicitly discuss the existing KD techniques based on adversarial learning in the following Sec. 8.1.

Potentials and open challenges Table. 9 summarizes the existing feature-based KD methods. It is shown that most

works employ feature transformations for both the teacher and the student. $L1$ or $L2$ loss is the most commonly used loss for measuring KD quality. A natural question one may ask is what's wrong with directly matching the features of teacher and student? If we consider the activation of each spatial position as a feature, the flattened activation map of each filter is a sample of the space of selected neurons with dimension HW, which reflects how DNN learns an image [55]. Thus, when matching distribution, it is less desirable to directly match the samples since the sample density can be lost in the space, as pointed in [52]. Although [69] proposes to distill knowledge by directly matching feature maps, a teaching assistant is introduced to learn the residual errors of between the feature maps of the student and teacher. This approach better mitigates the performance gap between the teacher and student, thus improving generalization capability.

Potentials: Feature-based methods show more generalization capabilities and quite promising results. In the next research, more flexible ways of determining the representative knowledge of features are expected. The approaches used in representation learning (*e.g.*, parameter estimation, graph models) might be reasonable solutions for these problems. Additionally, neural architecture search (NAS) techniques may better handle the selection of features. Furthermore, feature-based KD methods are possible for use in cross-domain transfer and low-level vision problems.

Open challenges: Although we have discussed most existing feature-based methods, it is still hard to say which one is best. First, it is difficult to measure the different aspects in which information is lost. Additionally, most works randomly choose intermediate features as knowledge, and yet do not provide a reason as to why they can be the representative knowledge among all layers. Third, the distillation position of features is manually selected based on the network or the task. Lastly, multiple features may not represent better knowledge than the feature of a single layer. Therefore, better ways to choose knowledge from layers and to represent knowledge could be explored.

4.2 Distillation from multiple teachers

Overall insight: *The student can learn better knowledge from multiple teachers, which are more informative and instructive than a single teacher.*

While impressive progress has been achieved under the common S-T KD paradigm, where knowledge is transferred from one high-capacity teacher network to a student network. The knowledge capacity in this setting is quite limited [48], and knowledge diversity is scarce for some special cases, such as cross-model KD [82]. To this end, some works probe learning a portable student from *multiple* teachers or an ensemble of teachers. The intuition behind this can be explained in analogy with the cognitive process of human learning. In practice, a student does not solely learn from a single teacher but learn a concept of knowledge better provided with instructive guidance from multiple teachers on the same task or heterogeneous teachers on different tasks. In such a way, the student can amalgamate and assimilate various illustrations of knowledge representations from multiple teacher networks, and build a comprehensive knowledge system [37, 39, 83]. As a result, many new KD methods [2, 18, 30, 32, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 48, 49, 69, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103] have been proposed. Although these works vary in various distillation scenarios and assumptions, they share some standard characteristics that can be categorized into five types: ensemble of logits, ensemble of feature-level information, unifying data sources, and obtaining sub-teacher networks from a single teacher network, customizing student network from heterogeneous teachers and learning a student network with diverse peers, via the ensemble of logits. We now explicitly analyze each category and provide insights on how and why they are valuable for the problems.

4.2.1 Distillation from the ensemble of logits

Model ensemble of logits is one of the popular methods in KD from multiple teachers as shown in Fig. 4(a). In such a setting, the student is encouraged to learn the softened output of the assembled teachers' logits (dark knowledge) via the cross-entropy loss as done in [2, 30, 32, 37, 38, 40, 43, 46, 87, 88, 90, 92, 93, 95, 96, 101, 102], which can be generalized into:

$$\mathcal{L}_{Ens}^{logits} = H(\frac{1}{m}\sum_i^m N_{T_i}^\tau(x), N_S^\tau(x)) \qquad (11)$$

where m is the total number of teachers, H is the cross-entropy loss, $N_{T_i}^\tau$ and N_S^τ are the i-th teacher's and i-th student's logits (or softmax ouputs), and τ is the temperature. The averaged softened output serves as the incorporation of multiple teacher networks in the output layer. Minimizing Eqn. 11 achieves the goal of KD at this layer. Note that the averaged softened output is more objective than that of any of the individuals, because it can mitigate the unexpected bias of the softened output existing in some of the input data.

Unlike the methods as mentioned above, [40, 82, 84, 91, 103] argue that taking the average of individual prediction may ignore the diversity and importance variety of the member teachers of an ensemble. Thus, they propose to learn the student model by imitating the summation of the teacher's predictions with a gating component. Then, Eqn. 11 becomes:

$$\mathcal{L}_{Ens}^{logits} = H(\sum_i^m g_i N_{T_i}^\tau(x), N_S^\tau(x)) \qquad (12)$$

where g_i is the gating parameter. In [84], the g_i is the normalized similarity $sim(D_{S_i}, D_T)$ of the source domain D_S and target domain D_T.

Summary: Distilling knowledge from the ensemble of logits mainly depends on taking the average or the summation of individual teacher's logits. Taking the average alleviates the unexpected bias, but it may ignore the diversity of individual teachers of an ensemble. The summation of the logits of each teacher can be balanced by the gating parameter g_i, but ways to determine better values of g_i is an issue worth studying in further works.

(a) ensemble of logits

(b) ensemble of features

(c) unifying data sources

(e) obtaining sub-teacher networks from singe teacher

(f) customizing student from heterogeneous teachers
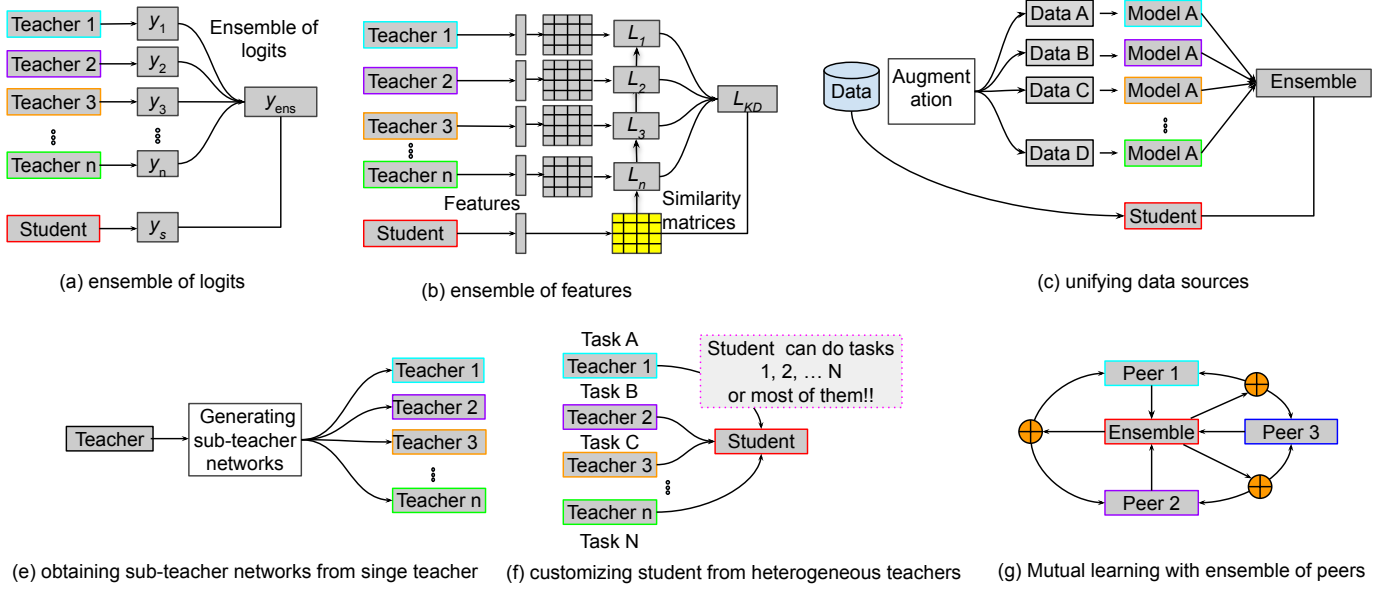
(g) Mutual learning with ensemble of peers

Fig. 4. Graphical illustration for KD with multiple teachers. The KD methods can be categorized into six types: (a) KD from the ensemble of logits, (b) KD from the ensemble of feature representations via some similarity matrices, (c) unifying various data sources from the same network (teacher) model A to generate various teacher models, (d) obtaining hierarchical or stochastic sub-teacher networks given one teacher network; (e) training a versatile student network from multiple heterogeneous teachers, (f) online KD from diverse peers via ensemble of logits.

4.2.2 Distillation from the ensemble of features

Distillation from the ensemble of feature representations is more flexible and advantageous than from the ensemble of logits, since it can provide more affluent and diverse cross-information to the student. However, distillation from the ensemble of features [18, 45, 48, 82, 92, 94, 102] is more challenging, since each teacher's feature representation at specific layers is different from the others. Hence, transforming the features, and forming an ensemble of the teachers' feature-map-level representations becomes the key problem, as illustrated in Fig. 4(b).

To address this issue, Park *et al.* [48] proposed feeding the student's feature map into some nonlinear layers (called transformers). The output is then trained to mimic the final feature map of the teacher network. In this way, the advantages of the general model ensemble and feature-based KD methods, as mentioned in Sec. 4.1.2, can both be incorporated. The loss function is given by:

$$\mathcal{L}_{Ens}^{fea} = \sum_{i}^{m} ||\frac{x_{T_i}}{||x_{T_i}||_2} - \frac{TF_i(x_S)}{||TF_i(x_S)||_2}||_1 \quad (13)$$

where x_{T_i} and x_S are i-th teacher's and i-th student's feature maps respectively, and TF is the transformer (*e.g.*, 3×3 convolution layer) used for adapting the student's feature with that of the teacher.

In contrast, Wu *et al.* [45] and Liu *et al.* [18] proposed letting the student model to imitate the learnable transformation matrices of the teacher models. This approach is an updated version of a single teacher model [61]. For the i-th teacher and student network in [45], the similarity between the feature maps is computed based on the Euclidean metric as:

$$\mathcal{L}_{Ens}^{fea} = \sum_{i}^{m} \alpha_i ||\log(A_S) - log(A_{T_i})||_F^2 \quad (14)$$

where α_i is the teacher weight for controlling the contribution of the i-th teacher, and α_i should satisfy $\sum_{i}^{m} \alpha_i = 1$. A_S and A_{T_i} are the similarity matrices of the student and the i-th teacher, respectively. These can be computed by $A_S = x_S^\mathsf{T} x_S$ and $A_T = x_{T_i}^\mathsf{T} x_{T_i}$, respectively.

Open challenges: Based on our review, it is evident that only a few studies propose distilling knowledge from the ensemble of feature representations. Although [48, 90] proposed to let the student directly mimic the ensemble of feature maps of the teachers via either non-linear transformation or similarity matrices with weighting mechanisms, there still exist some challenges. First, how can we know which teacher's feature representation is more reliable or more influential among the ensemble? Second, how can we determine the weighting parameter α_i for each student in an adaptive way? Third, instead of summing all feature information together, is there any mechanism of selecting the best feature map of one teacher from the ensemble as the representative knowledge?

4.2.3 Distillation by unifying data sources

Although the above mentioned KD methods using multiple teachers are good in some aspects, they assume that the target classes of all teacher and student models are the same. In addition, the dataset used for training is often scarce, and teacher models with high capacity are limited. To tackle these problems, some recent works [39, 42, 45, 86, 103, 104] propose data distillation by unifying data sources from multiple teachers, as illustrated in Fig. 4(c). The goal of these methods is to generate labels for the unlabeled data via various data processing approaches (*e.g.*, data augmentation) to train a student model.

Vongkulbhisal *et al.* [86] proposed to unify an ensemble of *heterogeneous classifiers* (teachers) which may be trained to classify different sets of target classes and can share the same network architecture. To generalize distillation,

a probabilistic relationship connecting the outputs of the heterogeneous classifiers with that of the unified (ensemble) classifier is proposed. Similarly, Wu *et al.* [45] and Gong *et al.* [104] also explored transferring knowledge from teacher models trained in existing data to a student model by using unlabeled data to form a decision function. Besides, some works utilize the potential of data augmentation approaches to build multiple teacher models from a trained teacher model. Radosavovic *et al.* [42] proposed a distillation method via *multiple transformations* on the unlabeled data to build diverse teacher models sharing the same network structure. The technique consists of four steps. First, a single teacher model is trained on manually labeled data. Second, the trained teacher model is applied to multiple transformations of the unlabeled data. Third, the predictions on the unlabeled data are converted into an ensemble of numerous predictions. Fourth, the student model is trained on the union of the manually labeled data and the automatically labeled data. Sau *et al.* [39] proposed an approach to simulate the effect of multiple teachers by injecting noise to the training data, and perturbing the logit outputs of a teacher. In such a way, the perturbed outputs not only simulate the setting of multiple teachers, but also generate noise in the softmax layer, thus regularizing the distillation loss.

Summary: Unifying data sources using data augmentation techniques and unlabeled data from a single teacher model to build up multiple sub-teacher models is also valid for training a student model. However, it requires a high-capacity teacher with more generalized target classes, which could confine the application of these techniques. In addition, the effectiveness of these techniques for some low-level vision problems should be studied further based on feature representations.

4.2.4 From a single teacher to multiple sub-teachers

It has been shown that students could be further improved with multiple teachers used as ensembles or used separately. However, using multiple teacher networks is resource heavy, and delays the training process. Following this, some methods [37, 41, 49, 84, 88, 90, 97] have been proposed to generate multiple sub-teachers from a single teacher network, as shown in Fig. 4(d). Lee *et al.* [49] proposed stochastic blocks and skip connections to teacher networks, so that the effect of multiple teachers can be obtained in the same resource from a single teacher network. The sub-teacher networks have reliable performance, because there exists a valid path for each batch. By doing so, the student can be trained with multiple teachers throughout the training phase. Similarly, Ruiz *et al.* [89] introduced hierarchical neural ensemble by employing a *binary-tree* structure to share a subset of intermediate layers between different models. This scheme allows controlling the inference cost on the fly, and deciding how many branches need to be evaluated. Tran *et al.* [88], Song *et al.* [41] and He *et al.* [97] introduced *multi-headed architectures* to build multiple teacher networks, while amortizing the computation through a shared heavy-body network. Each head is assigned to an ensemble member, and tries to mimic the *individual predictions* of the ensemble member.
Open challenges: Although network ensembles using stochastic or deterministic methods can achieve the effect

of multiple teachers and online KD, many uncertainties remain. Firstly, it is unclear how many teachers are sufficient for online distillation. Secondly, which structure is optimal among the ensemble of sub-teachers is unclear? Thirdly, balancing the training efficiency and accuracy of the student network is an open issue. These challenges are worth exploring in further studies.

4.2.5 Customizing student form heterogeneous teachers

In many cases, well-trained deep networks (teachers) are focused on different tasks, and are optimized for different datasets. However, most studies focus on training a student by distilling knowledge from teacher networks on the same task or on the same dataset. To tackle these problems, *knowledge amalgamation* has been initialized by recent works [18, 46, 83, 98, 99, 100, 102, 105, 106, 107] to learn a versatile student model by distilling knowledge from the expertise of all teachers, as illustrated in Fig. 4(e). Shen *et al.* [83], Ye *et al.* [99], Luo *et al.* [100] and Ye *et al.* [105] proposed training a student network by customizing the tasks without accessing human-labeled annotations. These methods rely on schemes such as branch-out [108] or selective learning [109]. The merits of these methods lie in their ability to reuse deep networks pre-trained on various datasets of diverse tasks to build a tailored student model based on the user demand. The student inherits most of the capabilities of heterogeneous teachers, and thus can perform multiple tasks simultaneously. Shen *et al.* [98] and Gao *et al.* [106] utilized a similar methodology, but focused on same task classification, with two teachers specialized in different classification problems. In this method, the student is capable of handling comprehensive or fine-grained classification. Dvornik *et al.* [46] attempted to learn a student that can predict unseen classes by distilling knowledge from teachers via few-shot learning. Rusu *et al.* [107] proposed a multi-teacher single-student policy distillation method that can distill multiple policies of reinforcement learning agents to a single student network for sequential prediction tasks.
Open challenges: Studies such as the ones mentioned above have shown considerable potential in customizing versatile student networks for various tasks. However, there are some limitations in such methods. Firstly, the student may not be compact due to the presence of branch-out structures. Secondly, current techniques mostly require teachers to share similar network structures (*e.g.*, encoder–decoder), which confines the generalization of such methods. Thirdly, training might be complicated because some works adopt a dual-stage strategy, followed by multiple steps with fine-tuning. These challenges open scopes for future investigation on knowledge amalgamation.

4.2.6 Mutual learning with ensemble of peers

One problem with conventional KD methods using multiple teachers is their computation cost and complexity, because they require pre-trained high-capacity teachers with two-stage (also called offline) learning. To simplify the distillation process, one-stage (online) KD methods [34, 40, 50, 64, 82, 85, 101, 110, 111] have been developed, as shown in Fig. 4(f). Instead of pre-training a static teacher model, these methods train a set of student models simultaneously by

TABLE 3
A taxonomy of KD with multiple teachers. L_{CE} is for cross-entropy loss, L_{Ens} is for the KD loss between the ensemble teacher and the student, L_{KD} indicates the KD loss between individual teacher and the student, $L_{KD_{fea+logits}}$ means KD loss using feature and logits, KL indicates the KL divergence loss for mutual learning, L_{GAN} is for adversarial loss, MMD means mean maximum discrepancy loss, L_{reg} is the regression loss, N/A means not available. *Note that the losses summarized are generalized terms, which may vary in individual works.*

Method	Ensemble Logits	Ensemble of Features	Unifying data sources	Customize student	Extending teacher	Online KD	Mutual learning	Major Loss functions
Anil [101]	✓	✗	✗	✗	✗	✓	✓	$L_{CE}+L_{Ens}$
Chen [50]	✓	✓	✗	✗	✗	✓	✓	$L_{CE}+L_{Ens}+L_{KD}$
Dvornik [46]	✓	✗	✗	✓	✗	✓	✓	$L_{CE}+L_{Ens}+L_{KD}$
Fukuda [91]	✓	✗	✗	✗	✗	✗	✗	$L_{CE}+L_{Ens}$
Furlanello [30]	✓	✗	✗	✗	✗	✗	✗	$L_{CE}+L_{Ens}$
He [97]	✗	✓	✗	✗	✓	✗	✗	L_1+L_{KD}
Jung [93]	✓	✗	✗	✗	✗	✗	✗	$L_{CE}+L_{KD}$
Lan [40]	✓	✗	✗	✗	✗	✓	✗	$L_{CE}+L_{Ens}$
Lee [49]	✓	✓	✗	✗	✓	✓	✓	N/A
Liu [18]	✗	✓	✗	✓	✗	✓	✗	$L_{CE}+L_{KD}$
Luo [100]	✓	✓	✗	✓	✗	✗	✗	$L_{CE}+L_{KD_{fea+logits}}$
Zhou [102]	✓	✓	✗	✓	✗	✗	✗	MMD+L_{KD}
Mirzadeh [32]	✓	✗	✗	✗	✗	✗	✗	$L_{CE}+L_{KD}$
Papernot [38]	✓	✗	✓	✗	✗	✓	✗	L_{KD}
Park [48]	✗	✓	✗	✗	✗	✗	✗	$L_{CE}+L_{KD}$
Radosavovic [42]	✓	✗	✓	✗	✗	✗	✗	N/A
Ruder [84]	✓	✗	✗	✗	✗	✗	✗	$L_{CE}+L_{KD}$
Ruiz [89]	✓	✗	✗	✗	✓	✗	✗	$L_{CE}+L_{Ens}$
Sau [39]	✓	✗	✓	✗	✗	✗	✗	L_2 (KD)
Shen [98]	✓	✓	✗	✓	✗	✗	✗	$L_{KD}+L_{PL}$
Shen [83]	✓	✓	✗	✓	✗	✗	✗	$L_{KD_{fea+logits}}+L_{reg}$
Song [41]	✓	✗	✗	✗	✓	✓	✗	$L_{CE}+L_{KD}$
Tarvaninen [2]	✓	✗	✗	✗	✗	✓	✗	L_{KD}
Tran [88]	✓	✗	✗	✗	✓	✗	✓	$L_{CE}+L_{Ens}+$KL
Vongkulbhisal [86]	✓	✗	✓	✗	✗	✗	✗	$L_{CE}+L_{Ens}$
Wu [45]	✗	✓	✓	✗	✗	✗	✗	L_{KD}
Wu [90]	✓	✗	✗	✗	✗	✗	✗	$L_{CE}+L_{Ens}$
Yang [87]	✓	✗	✗	✗	✗	✗	✗	$L_{CE}+L_{KD}$
Ye [99]	✓	✓	✗	✓	✗	✗	✗	$L_{KD}+L_{KD}$
You [37]	✓	✓	✗	✗	✗	✗	✗	$L_{CE}+L_{KD_{fea+logits}}$
Zhang [82]	✓	✓	✗	✗	✗	✗	✗	$L_{CE}+L_{KD_{fea+logits}}$
Zhang [34]	✓	✗	✗	✗	✗	✓	✓	$L_{CE}+$KL
Zhu [85]	✓	✗	✗	✗	✗	✓	✓	$L_{CE}+$KL$+L_{Ens}$
Chung [64]	✓	✓	✗	✗	✗	✓	✓	$L_{CE}+L_{GAN}+$KL
Kim [110]	✓	✓	✗	✗	✗	✓	✓	$L_{CE}+L_{Ens}+$KL
Hou [111]	✗	✓	✗	✗	✗	✓	✗	$L_{CE}+L_{Ens}$
Xiang [103]	✓	✗	✓	✗	✗	✗	✗	$L_{CE}+L_{KD}$

making them learn from each other in a peer-teaching manner. There are some benefits of such methods. First, these approaches merge the training processes of teachers and student models, and use peer networks to provide teaching knowledge. Second, these online distilling strategies can improve the performance of models of any capacity, leading to generic applications. Third, such a peer-distillation method can sometimes outperform teacher-based two-stage KD methods. For KD with mutual learning, the distillation loss of *two peers* is based on the KL divergence, which can be formulated as:

$$\mathcal{L}_{Peer}^{KD} = KL(z_1, z_2) + KL(z_2, z_1) \tag{15}$$

where KL is the KL divergence function, and z_1 and z_2 are the predictions of peer one and peer two, respectively.

In addition, Lan *et al*. [40] and Chen *et al*. [50] also constructed a multi-branch variant of a given target (student) network by adding auxiliary branches to create a local ensemble teacher (also called a group leader) model from all branches. Each branch was trained with a distillation loss that aligns the prediction of that branch with the teacher's prediction. Mathematically, the distillation loss can be formulated by minimizing the KL divergence of z_e (prediction of the ensemble teacher) and prediction z_i of the i-th branch peer:

$$\mathcal{L}_{Ens}^{KD} = \sum_{i=1}^{m} KL(z_e, z_i) \tag{16}$$

where the prediction $z_e = \sum_{i=1}^{m} g_i z_i$. g_i is the weighting score or attention-based weights [50] of the i-th branch peer z_i.

Although most of these methods only consider using logit information, some works also exploit feature information. Chung *et al*. [64] proposed a feature-map-level distillation by employing adversarial learning (discriminators). Kim *et al*. [110] introduce a feature fusion module to form an ensemble teacher. However, the fusion is based on the concatenation of the features (output channels) from the branch peers. Moreover, Liu *et al*. [18] presented a knowledge flow

framework which moves the knowledge from the features of multiple teacher networks to a student.

Summary: Compared to two-stage KD methods using pre-trained teachers, distillation from student peers has many merits. The methods are based on mutual learning of peers, and sometimes on ensembles of peers. Most studies rely on logit information; however, some works also exploit feature information via adversarial learning or feature fusion. There is room for improvement in this direction. For instance, the number of peers most optimal for KD processing is worth investigating. In addition, the possibility of using both the online and offline methods simultaneously when the teacher is available is intriguing. Reducing the computation cost without sacrificing accuracy and generalization is also an open issue. We will discuss the advantages and disadvantages of online and offline KD in Sec. 6.1.

Potentials Table. 3 summarizes the KD methods with multiple instructors. Overall, most methods rely on the ensemble of logits. However, the knowledge of feature representations has not been taken into account much. Therefore, it is possible to exploit the knowledge of the ensemble of feature representations by designing better gating mechanisms. Unifying data sources and extending teacher models are two effective methods for reducing individual teacher models; however, their performances are degraded. Thus, overcoming this issue needs more research. Customizing a versatile student is a valuable idea, but existing methods are limited by network structures, diversity, and computation costs, which must be improved in future works.

5 DISTILLATION BASED ON DATA FORMAT

5.1 Data-free distillation

Overall insight: *Can we achieve KD when the original data for the teacher or (un)labeled data for training student are not available?*

One major limitation of most KD methods such as [1, 28, 48, 52] is that they assume the training samples of the original networks (teachers) or of target networks (students) to be available. However, the training dataset is sometimes unknown in real-world applications owing to privacy and transmission concerns [112]. To handle this problem, some representative data-free KD paradigms [112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122] are newly developed. A taxonomy of these methods are summarized in Table. 4, and detailed technical analysis is provided as follows.

5.1.1 Distillation based on metadata

To the best of our knowledge, Lopes *et al.* [112] initially proposed to reconstruct the original training dataset using only teacher model and it is *metadata* recorded in the form of *precomputed activation statistics*. Thus, the objective is to find the set of images whose representation best matches the one given by the teacher network. Gaussian noise is randomly passed as input to the teacher, and the gradient descent (GD) is made to minimize the difference between the metadata and the representations of noise input. To better constrain the reconstruction, the metadata of *all* layers of the teacher model are used and recorded to train the student model with high accuracy. Bhardwaj *et al.* [113] demonstrated that

metadata from a *single* layer (average-pooling layer) using k-means clustering is sufficient to achieve high student accuracy. In contrast to [112, 113] requiring sampling the activations generated by real data, Haroush *et al.* [114] proposed using metadata (*e.g.*, channel-wise mean and standard deviation) from Batch Normalization (BN) [74] layer with synthetic samples. The objective of metadata-based distillation can be formulated as:

$$X^* = \arg \min_{X \sim R^{H \times W}} L(\Phi(X), \Phi_0) \qquad (17)$$

where X^* is the image (with width W and height H) to be found, Φ is the representation of X, Φ_0 is the representation of the metadata, and L is the loss function (*e.g.*, l_2).

5.1.2 Distillation based on class-similarities

Nayak *et al.* [115] argued that the metadata used in [112, 113] are actually not complete data-free approaches, since the metadata is formed using the training data itself. They instead proposed a zero-shot KD approach, in which no data samples and no metadata information are used. In particular, the approach obtains useful prior information about the underlying data distribution in the form of *class similarities* from the model parameters of the teacher. This prior information can be further utilized for crafting data samples (also called data impressions (DIs)) by modeling the output space of the teacher model as a Dirichlet distribution. The class similarity matrix, similar to [61], is calculated based on the softmax layer of the teacher model. The objective of the data impression X_i^k can be formulated based on the cross-entropy loss:

$$X_i^k = \arg \min_X L_{CE}(y_i^k, T(X, \theta_T, \tau)) \qquad (18)$$

where y_i^k is the sampled i-th softmax vector, and k is the certain class.

5.1.3 Distillation using generator

Considering the limitation of metadata and similarity-based distillation methods, some works [116, 117, 118, 119, 120, 121] propose novel data-free KD methods via adversarial learning [78, 123, 124]. Although the tasks and network structures vary in these methods, most are built on a common framework. That is, the pretrained teacher network is fixed as a discriminator, while a generator is designed to synthesize training samples, given various input source (*e.g.*, noise [116, 118, 119, 120]). However, slight differences exist in some studies. Fang *et al.* [117] point out the problem of taking the teacher as the discriminator since the information of the student is ignored, and generated samples cannot be customized without the student. Thus, they take both the teacher and the student as the discriminator to reduce the discrepancy between them, while a generator is trained to generate some samples to adversarially enlarge the discrepancy. In contrast, Ye *et al.* [118] focus more on strengthening the generator structure, and three generators are designed and subtly used. Specifically, a group-stack generator is trained to generate the images originally used for pre-training the teachers, and the intermediate activations. Then, a dual generator takes the generated image as the input, the dual part is taken as the target network (student), and regrouped for multi-label classifications. To

TABLE 4
A taxonomy of data-free knowledge distillation.

Method	Original data needed	Metadata or prior info.	Number of generators	Inputs	Discriminator	Multi-task distillation
Lopes [112]	✓	Activations of all layers	✗	Image shape	✗	✗
Bhardwaj [113]	✓	Activations of pooling layer	✗	Image shape	✗	✗
Haroush [114]	✓	Batch normalization layer	✗	Image shape	✗	✗
Nayak [115]	✗	Class similarities	✗	Class label+ Number of DIs	✗	✗
Chen [116]	✗	✗	One	Noise	Teacher	✗
Fang [117]	✗	✗	One	Noise/images	Teacher + student	✗
Ye [118]	✗	✗	Three	Noise	Teachers	✓
Yoo [119]	✗	✗	One	Noise + class labels	Teacher	✗
Yin [120]	✗	✗	One	Noise	Teacher	✗
Micaelli [121]	✗	✗	One	Noise	Teacher	✗

compute the adversarial loss for both the generated image and the intermediate activations, multiple group-stack discriminators (multiple teachers) are also designed to amalgamate multi-knowledge into the generator. Yoo *et al*. [119] make the generator take two inputs: a sampled class label y, and noise z. Meanwhile, a decoder is also applied to reconstruct the noise input z' and class label y' from the fake data x' generated by the generator from the noise input z, and class label y. Thus, by minimizing the errors between y and y' and between z and z', the generator generates more reliable data. Although adversarial loss is not used in [120], the generator (called DeepInversion) taking an image prior regularization term to synthesize images is modified from DeepDream [125].

5.1.4 Open challenges for data-free distillation
Although data-free KD methods have shown considerable potential and new directions for KD, there still exist many challenges. First, the recovered images are unrealistic and low-resolution, which may not be utilized in some data-captious tasks (*e.g*., semantic segmentation). Second, training and computation of the existing methods might be complicated due to the utilization of many modules. Third, diversity and generalization of the recovered data are still limited, compared with the methods of data-driven distillation. Fourth, the effectiveness of such methods for low-level tasks (*e.g*., image super-resolution) needs to be studied further.

5.2 Distillation with a few data samples
Overall insight: *How to perform efficient knowledge distillation with only a small amount of training data?*

Most KD methods with S-T structures, such as [1, 48, 54, 64], are based on matching information (*e.g*., logits, hints) and optimizing the KD loss with the fully annotated large-scale training dataset. As a result, the training is still data-heavy and processing-inefficient. To enable efficient learning of the student while using small amount of training data, some works [122, 126, 127, 128, 129] propose few-sample KD strategies. The technical highlight of these methods is based on generating pseudo training examples, or aligning the teacher and the student with layer-wise estimation metrics.

5.2.1 Distillation via pseudo examples
Insight: *If training data is insufficient, try to create pseudo examples for training the student.*

[122, 126, 128] focus on creating pseudo training examples when training data is scarce and leading to overfitting of the student network. Specifically, Kimura *et al*. [128] adopt the idea of inducing points [130] to generate pseudo training examples, which are then updated by applying adversarial examples [131, 132], and further optimized by an imitation loss. Liu *et al*. [126] generate pseudo ImageNet [51] labels from a teacher model (trained with ImageNet), and also utilize the semantic information (*e.g*., words) to add a supervision signal for the student. Interestingly, Kulkarni *et al*. [122] create a 'mismatched' unlabeled stimulus (*e.g*., soft labels of MNIST dataset [133] provided by the teacher trained on CIFAR dataset [134]), which are used for augmenting a small amount of training data to train the student.

5.2.2 Distillation via layer-wise estimation
Insight: *Layer-wise distillation from the teacher network via estimating the accumulated errors on the student network can also achieve the purpose of few-example KD.*

In Bai *et al*. [129] and Li *et al*. [127], the teacher network is first compressed to create a student via network pruning [135], and layer-wise distillation losses are then applied to reduce the estimation error on given limited samples. To conduct layer-wise distillation, Li *et al*. [127] add a 1×1 layer after each pruned layer block in the student, and estimate the least-squared error to align the parameters with the student. Bai *et al*. [129] employ cross distillation losses to mimic the behavior of the teacher network, given its current estimations.

5.2.3 Challenges and potentials
Although KD methods with a small number of examples inspired by the techniques of data augmentation and layer-wise learning are convincing, these techniques are still confined by the structures of teacher networks. This is because most methods rely on network pruning from teacher networks to create student networks. Besides, the performance of the student is heavily dependent on the amount of the crafted pseudo labels, which may impede the effectiveness
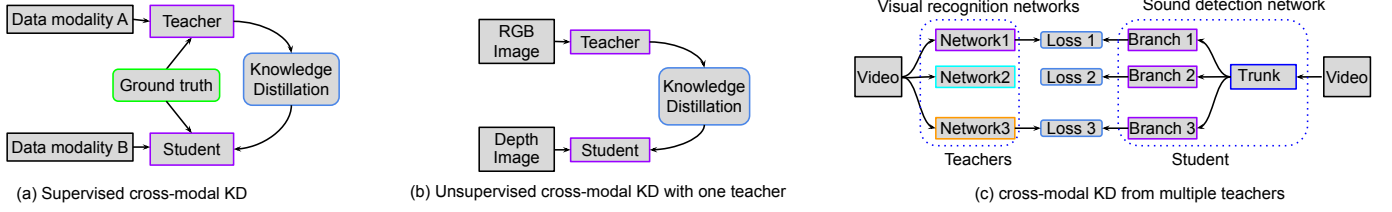
Fig. 5. Graphical illustration of cross-modal KD methods. (a) supervised cross-modal KD from the teacher with one modality to the student with another modality. (b) unsupervised cross-modal KD with one teacher. (c) unsupervised cross-modal KD with multiple teachers, each of which is transferring the discriminative knowledge to the student.

of these methods. Lastly, most works focus on generic classification tasks, and it is unclear whether these methods are effective for tasks without class labels (*e.g.*, low-level vision tasks).

5.3 Cross-modal distillation

Overall insight: *KD for cross-modal learning is typically performed with network architectures containing modal-specific representations or shared layers, utilizing the training images in correspondence of different domains.*

One natural question we ask is if it is possible to transfer knowledge from a pre-trained teacher network for one task to a student learning another task, while the training examples are in correspondence across domains. *Note that KD for cross-modal learning is essentially different from that for domain adaptation, in which data are drawn independently from different domains, but the tasks are the same.*

Compared to previously mentioned KD methods focused on transferring supervision within the same modality between the teacher and the student, cross-modal KD uses the teacher's representation as a supervision signal to train the student learning another task. In this problem setting, the student needs to rely on the visual input of the teacher to accomplish its task. Following this, many novel cross-modal KD methods [136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 152, 153, 156] have been proposed. We now provide a systematic analysis of the technical details, and point the challenges and potential of cross-domain distillation.

5.3.1 Supervised cross-modal distillation

Using the ground truth labels for the data used in the student network is the common way of cross-modal KD, as shown in Fig. 5(a). Do *et al*. [149], Su *et al*. [137], Nagrani *et al*. [138], Nagrani *et al*. [139] and Hoffman *et al*. [140] rely on supervised learning for cross-modal transfer. Several works [138, 139, 141, 155] leverage the synchronization of visual and audio information in the video data, and learn a joint embedding between the two modalities. Afouras *et al*. [141] and Nagrani *et al*. [139] transfer the voice knowledge to learn a visual detector, while Nagrani *et al*. [138] utilize visual knowledge to learn a voice detector (student). In contrast, Hoffman *et al*. [140], Do *et al*. [149] and Su *et al*. [137] focus on different modalities in the visual domain only. In particular, Hoffman *et al*. [140] learn a depth network by transferring the knowledge from an RGB network, and fuse the information across modalities. This improves the object recognition performance during the test time. Su *et al*. [137]

utilize the knowledge from high-quality images to learn a classifier with better generalization on low-quality image (paired).

5.3.2 Unsupervised cross-modal distillation

Most cross-modal KD methods exploit unsupervised learning, since the labels in target domains are hard to get. Thus, these methods are also called distillation '*in the wild*'. In this setting, the knowledge from the teacher's modality provides *supervision* for the student network. To this end, some works [136, 141, 142, 143, 144, 145, 146, 147, 148, 150, 151, 152, 153, 156] aimed for cross-modal distillation in an unsupervised manner.

5.3.3 Learning from one teacher

Afouras *et al*. [141], Albanie *et al*. [142], Gupta *et al*. [143], Thoker *et al*. [145], Zhao *et al*. [146], Owens *et al*. [147], Kim *et al*. [151], Arandjelovic *et al*. [148], Gan *et al*. [154], and Hafner *et al*. [153] focus on distilling knowledge from one teacher (see Fig. 5(b)), and mostly learn a single student network. Thoker *et al*. [145] and Zhao *et al*. [146] learn two students. Especially, Thoker *et al*. refer to mutual learning [34], where two students learn from each other based on two KL divergence losses. In addition, Zhao *et al*. [146] exploit the feature fusion strategy, similar to [110, 157] to learn a more robust decoder. Do *et al*. [149] focuses on unpaired images of two modalities, and learns a semantic segmentation network (student) using the knowledge from the other modality (teacher).

5.3.4 Learning from multiple teachers

Aytar *et al*. [136], Salem *et al*. [144], Aytar *et al*. [150] and Do *et al*. [149] exploit the potential of distilling from multiple teachers as mentioned in Sec. 4.2. Most methods rely on concurrent knowledge among visual, audio, and textual information, as shown in Fig. 5(c). However, Salem *et al*. [144] focus on the visual modality only, where teachers learn the information of object detection, image classification, and scene categorization via a multi-task approach, and distill the knowledge to a single student.

5.3.5 Potentials and open challenges

Potentials: Based on the analysis of the existing cross-modal KD techniques in Table. 5, we can see that cross-modal KD expands the generalization capability of the knowledge learned from the teacher models. Cross-domain KD has considerable potential in *relieving the dependence* for a large amount of labeled data in one modality or both. In addition,

TABLE 5
A taxonomy of cross-modal knowledge distillation methods.

Method	Use GT	Source modality	Target modality	Number of teachers	Online KD	Knowledge	Model compression
Ayter [136]	✗	RGB frames	Sound	Two	✗	Logits	✗
Su [137]	✓	HR image map	LR image	One	✗	Soft labels	✓
Nagrani [138]	✓	RGB frames	Voice	One	✓	Soft labels	✗
Nagrani [139]	✓	Voice/face	Face/voice	Multiple	✓	Features	✗
Hoffman [140]	✓	RDG images	Depth images	One	✗	Features	✗
Afouras [141]	✗	Audio	Video	One	✗	Soft labels	✗
Albanie [142]	✗	Video frames	Sound	One	✗	Logits	✗
Gupta [143]	✗	RGN images	Depth images	One	✗	Soft labels	✗
Salem [144]	✗	Scene classification, object detection	Localization	Three	✗	Soft labels	✗
Thoker [145]	✗	RGB video	Skeleton data	One	✗	Logits	✗
Zhao [146]	✗	RGB frames	Heatmaps	One	✗	Confidence maps	✗
Owens [147]	✗	Sound	Video frames	One	✗	Soft labels	✗
Arandjelovic [148]	✗	Video frames	Audio	One	✗	Features	✗
Do [149]	✓	Image, Questions, Answer info.	Image questions	Three	✗	Logits	✓
Aytar [150]	✓	Image	Sound, Image, Text	Three	✗	Features	✗
Kim [151]	✗	Sound/images	Images/sound	One	✗	Features	✗
Dou [152]	✗	CT images	MRI images	One	✓	Logits	✓
Hafner [153]	✗	Depth images	RGB images	One	✗	Embeddings	✗
Gan [154]	✓	Video frame	Sound	One	✗	Feature soft labels	✗
Perez [155]	✓	RGB video Acoustic images	Audio	One	✗	Soft labels	✗

cross-domain KD is more *scalable*, and can be *easily* applied to *new* distillation tasks. Moreover, it is advantageous for learning multiple modalities of data 'in the wild', since it is relatively easy to get data with one modality based on other data. In visual applications, cross-modal KD has the potential to distill knowledge among images taken from different types of cameras. For instance, one can distill knowledge from an RGB image to event streams (stacked event images from event cameras) [137, 158].

Open challenges: Since the knowledge is the transferred representations (*e.g.,* logits, features) of teacher models, ensuring the robustness of the transferred knowledge is crucial. We hope to transfer the good representations, but negative representations do exist. Thus, it is imperative that the supervision provided by the teachers is complementary to the target modality. Moreover, existing cross-modal KD methods are highly dependent on data sources (*e.g.,* video, images), but finding data with paired (*e.g.,* RGB image with depth pair) or multiple modalities (class labels, bounding boxes and segmentation labels) is not always an easy task. We are compelled to ask if it is possible to come up with a way for data-free distillation or distillation with a few examples? In other words, is it possible to just learn a student model with the data from the target modality based on the knowledge of the teacher, without referencing the source modality?

Moreover, existing cross-modal KD methods are mostly offline methods, which are computation-heavy and memory-intensive. Thus, it would be better if an online KD strategy is considered. Lastly, some works (*e.g.,* [144, 150]) learn a student model using the knowledge from multiple teachers. However, the student is less versatile or modality-dependent. Inspired by the analysis of Sec. 4.2.5, we open a research question: Is it possible to learn a versatile student that can perform tasks from multiple modalities?

6 ONLINE AND TEACHER-FREE DISTILLATION

6.1 Online distillation

Overall insight: *With the absence of a pre-trained powerful teacher, simultaneously training a group of student models by learning from peers' predictions is an effective substitute for two-stage (offline) KD*

In this section, we provide a deeper analysis of online (one-stage) KD methods in contrast to the previously discussed offline (two-stage) KD methods. Offline KD methods often require pre-trained high-capacity teacher models to perform one-way transfer [1, 8, 37, 54, 91, 111, 159, 160, 161]. However, it is sometimes difficult to get such 'good' teachers, and the performance of the student gets degraded when the gap of network capacity between the teacher and the student is significant. In addition, two-stage KD requires many parameters, resulting in higher computation costs. To overcome these difficulties, some studies focus on online KD that simultaneously trains a group of student peers by learning from the peers' predictions.
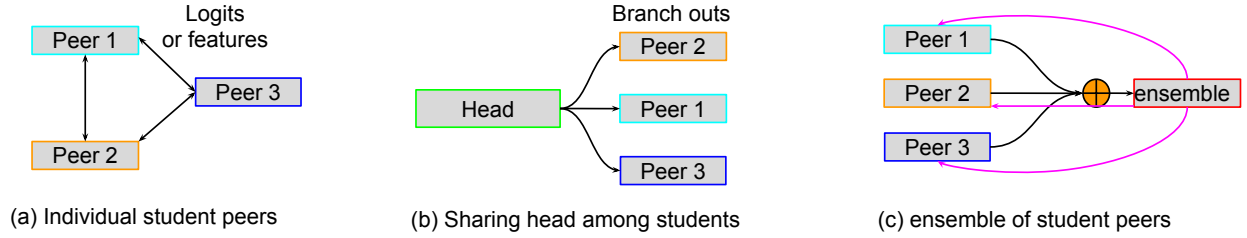
Fig. 6. An illustration of online KD methods. (a) online KD with individual student peer learning from each other, (b) online KD with student peers sharing trunk (head) structure, (c) online KD by assembling the weights of each student to form a teacher or group leader.

6.1.1 Individual student peers

Zhang *et al.* [34], Gao *et al.* [160] and Anil *et al.* [101] focus on online *mutual learning* [34] (also called codistillation) in which a pool of untrained student networks with the same network structure simultaneously learns the target task. In such a peer-teaching environment, each student learns the average class probabilities from the other (see Fig. 6(a)). However, Chung *et al.* [64] also employ individual students, and additionally design a feature map-based KD loss via adversarial learning. Hou *et al.* [111] proposed DualNet, where two individual student classifiers were fused into one fused classifier. During training, the two student classifiers are locally optimized, while the fused classifier is globally optimized as a mutual learning method. Other methods such as [159, 162], focus on online video distillation by periodically updating the weights of the student, based on the output of the teacher. Although codistillation achieves parallel learning of students, [34, 64, 101, 111] do not consider the ensemble of peers' information as done in other works such as [50, 160].

6.1.2 Sharing blocks among student peers

Considering the training cost of employing individual students, some works propose sharing network structures (*e.g.*, head sharing) of the students with branches as shown in Fig. 6(b). Song *et al.* [41] and Lan *et al.* [40] build the student peers on multi-branch architectures [131]. In such a way, all structures together with the shared trunk layers (often use head layers) can construct individual student peers, and any target student peer network in the whole multi-branch can be optimized.

6.1.3 Ensemble of student peers

While using codistillation and multi-architectures can facilitate online distillation, knowledge from all student peers is not accessible. To this end, some studies [40, 50, 110, 160, 161] proposed using the assembly of knowledge (logits information) of all student peers to build an on the fly teacher or group leader, which is in turn distilled back to all student peers to enhance student learning in a closed-loop form, as shown in Fig. 6(c). Note that in ensemble distillation, the student peers can either be independent, or share the same head structure (trunk). The ensemble distillation loss is given by Eqn. 12 of Sec. 4.2, where a gating component g_i is added to balance the contribution of each student. Chen *et al.* [50] obtain the gating component g_i based on the self-attention mechanism [72].

6.1.4 Summary and open challenges

Summary: Based on the above analysis, we have determined that codistillation, multi-architectures, and ensemble learning are three main techniques for online distillation. There are some advantages of online KD compared with offline KD. Firstly, online KD does not require pre-training teachers. Secondly, online learning provides a simple but effective way to improve the learning efficiency and generalizability of the network, by training together with other student peers. Thirdly, online learning with student peers often results in better performance than offline learning.

Open challenges: There are some challenges in online KD. Firstly, there is a lack of theoretical analysis for why online learning is sometimes better than offline learning. Secondly, in online ensemble KD, simply aggregating students' logits to form an ensemble teacher restrains the diversity of student peers, thus limiting the effectiveness of online learning. Thirdly, existing methods are confined problems in which ground truth (GT) labels exist (*e.g.*, classification). However, for some problems (*e.g.*, low-level vision problems), ways for the student peers to form effective ensemble teachers need to be exploited.

6.2 Teacher-free distillation

Overal insight: *Is it possible to enable the student to distill knowledge by itself to achieve plausible performance?*

The conventional KD approaches [1, 52, 57, 61, 110] still have many setbacks to be tackled, although significant performance boost has been achieved. First of all, these approaches have low efficiencies, since student models scarcely exploit all knowledge from the teacher models. Secondly, designing and training high-capacity teacher models still face many obstacles. Thirdly, two-stage distillation requires high computation and storage costs. To tackle these challenges, several novel self-distillation frameworks [25, 30, 80, 163, 164, 165, 166, 167, 168, 169, 170, 171] have been proposed recently. The goal of self-distillation is to learn a student model by distilling knowledge in itself without referring to other models. We now provide a detailed analysis of the technical details for self-distillation.

6.2.1 Born-again distillation

Insight: *Sequential self-teaching of students enables them to become masters, and outperform their teachers significantly.*

Furlanello *et al.* [30] initializde the concept of self-distillation, in which the students are parameterized identically to their teachers, as shown in Fig. 7(a). Through sequential teaching, the student is continuously updated,
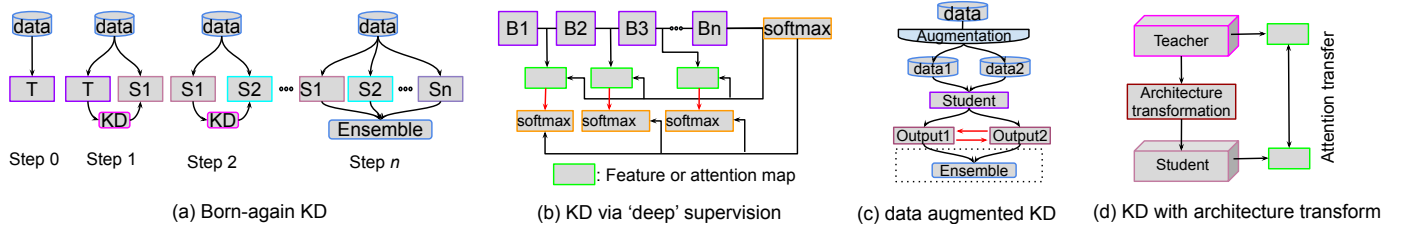
Fig. 7. An illustration of self-distillation methods. (a) born-again distillation. Note that T and $S_1, \cdots, S_n$ can be multi-tasks. (b) distillation via 'deep' supervision where the deepest branch (B_n) is used to distill knowledge to shallower branches. (c) distillation via data augmentation (*e.g.*, rotation, cropping). (d) distillation with network architecture transformation (*e.g.*, changing convolution filters).

TABLE 6
A taxonomy of self-distillation methods. Logits and hints indicate the knowledge to be distilled. 'Deep' supervision is for self-distillation from deepest branch (or layer) of the student network. One-stage KD is checking whether self-distillation is achieved in one step. ✓ / ✗ is for yes/no.

Method	Logits	Hints	Data augmentation	'Deep' supervision	One-stage KD	Multi-task KD	Architecture transformation
Clarm [80]	✓	✗	✗	✗	✗	✓	✗
Chowley [167]	✗	Attention map	✗	✗	✗	✗	✓
Furlanello [30]	✓	✗	✗	✗	✗	✗	✗
Hahn [165]	✓	✗	✗	✗	✗	✗	✗
Hou [169]	✗	Attention maps	✗	✓	✓	✗	✗
Luan [170]	✓	Feature maps	✗	✓	✓	✓	✗
Lee [166]	✓	✗	✓	✗	✗	✗	✗
Xu [164]	✓	Feature maps	✓	✗	✓	✗	✗
Zhang [168]	✓	Feature maps	✗	✓	✓	✗	✗
Yang [25]	✓	✗	✗	✗	✗	✗	✗

and at the end of the procedure, additional performance gains are achieved by an ensemble of multiple student generations. Hahn *et al.* [165] apply born-again distillation [30] to natural language processing. Yang *et al.* [25] observe that it remains unclear how S-T optimization works, and they then focus on putting strictness (adding an extra term to the standard cross-entropy loss) to the teacher model, such that the student can better learn inter-class similarity, and potentially prevent over-fitting. Instead of learning a single task, Clark *et al.* [80] extend [30] to the multi-task setting, where single-task models are distilled sequentially to teach a multi-task model. Since the born-again distillation approach is based on the multi-stage training, it is less efficient and computation-heavy compared to the following methods.

6.2.2 Distillation via 'deep' supervision

Insight: *The deeper layers (or branches) in the student model contains more useful information than those of shallower layers.*

Among the methods, Hou *et al.* [169], Luan *et al.* [170] and Zhang *et al.* [168] propose similar approaches where the target network (student) is divided into several shallow sections (branches) according to their depths and original structures (see Fig. 7(b)). As the deepest section may contain more useful and discriminative feature information than shallower sections, the deeper branches can be used to distill knowledge to the shallower branches. In contrast, in [169], instead of directly distilling features, attention-based methods used in [36] are adopted to force shallower layers to mimic the attention maps of deeper layers. Luan *et al.* [170] make each layer branch (ResNet block) a classifier. Thus, the deepest classifier is used to distill earlier the classifiers' feature maps and logits.

6.2.3 Distillation based on data augmentation

Insight: *Data augmentation (e.g., rotation, flipping, cropping, etc) during training forces the student network to be invariant to augmentation transformations via self-distillation.*

Although most methods focus on how to better supervise student in self-distillation, data representations for training the student are not fully excavated and utilized. To this end, Xu *et al.* [164] and Lee *et al.* [166] focus on self-distillation via data augmentation of the training samples, as shown in Fig. 7(c). There are some advantages to such a framework. First, it is efficient and effective to optimize a single student network without branching or the assistance of other models. Second, with data-to-data self-distillation, the student learns more inherent representations for generalization. Third, the performance of the student model is significantly enhanced with relatively low computation cost and memory load.

Xu *et al.* [164] apply random mirror and cropping to the batch images from the training data. Besides, inspired by mutual learning [34], the last feature layers and softmax outputs of the original batch image and distorted batch images are mutually distilled via MMD loss [55] and KL divergence loss, respectively. In contrast, Lee *et al.* [166] consider two types of data augmentation (rotation and color permutation to the same image), and the ensemble method used in [40, 50, 85] is employed to aggregate all logits of the student model to one, which is in turn is used by the student to transfer the knowledge to itself.

6.2.4 Distillation with architecture transformation

Insight: *A student model can be derived by changing the convolution operators in the teacher model with any architecture change.*

In contrast with all the above-mentioned self-distillation methods, Crowley *et al.* [167] proposes structure model distillation for memory reduction by replacing standard convolution blocks with cheaper convolutions, as shown in Fig. 7(d). In such a way, a student model that is a simple transformation of the teacher's architecture is produced. Then, attention transfer (AT) [55] is applied to align the teacher's attention map with that of the student's.

6.2.5 Summary and open challenges

Summary: In Table. 6, we summarize and compare different self-distillation approaches. Overall, using logits/feature information and two-stage training for self-distillation with 'deep' supervision from the deepest branch are main stream. Besides, data augmentation and attention-based self-distillation approaches are promising. Lastly, it is shown that multi-task learning with self-distillation is also a valuable direction, deserving more research.

Open challenges: There still exist many challenges to tackle. First, theoretical support laks in explaining why self-distillation works better. Mobahi *et al.* [163] provide theoretical analysis for born-again distillation [30] and find that self-distillation may reduce over-fitting by loop-over training, thus leading to good performance. However, it is still unclear why other self-distillation methods (*e.g.*, online 'deep' supervision [168, 169, 170]) work better.

In addition, existing methods focus on self-distillation with certain types of group-based network structures (*e.g.*, ResNet group). Thus, the generalization and flexibility of self-distillation methods need to be probed further. Lastly, all existing methods focus on classification-based tasks, and it is not clear whether self-distillation is effective for other tasks (*e.g.*, low-level vision tasks).

7 LABEL-REQUIRED/-FREE DISTILLATION

Overall Insight: *It is possible to learn a student without referring to the labels of training data?*

7.1 Label-required distillation

The success of KD relies on the assumption that labels provide the required level of semantic description for the task at hand [1, 4]. For instance, in most existing KD methods for classification-related tasks [1, 8, 9, 28, 48, 131, 172], image-level labels are required for learning student network. Meanwhile, some works exploit the pseudo labels when training data are scarce. We now provide a systematic analysis for these two types of methods.

7.1.1 KD with original labels.

Using the ground truth labels for the data used in the student network is the common way for KD. As depicted in Eq.2, the overall loss function is composed of the student loss and the distillation loss. The student loss is heavily dependent on the ground truth label y. Following this fashion, main-stream methods mostly utilize the original labels and design better distillation loss terms to achieve better performances [50, 55, 173, 174, 175, 176]. This convention has also been continually adopted in recent KD methods, such as online distillation [50, 159], teacher-free distillation [82, 154, 168], and even cross-modal learning

[137, 138, 139, 143, 149, 150]. While using labels expand the generalization capability of knowledge for learning student network, such approaches fail when labels are scarce or unavailable.

7.1.2 KD with pseudo labels.

Some works also exploit the pseudo labels. The most common methods can be discomposed into two groups. The first one aims to create noisy labels. [20, 21, 22, 23] propose to leverage large number of noisy labels to augment small amount of clean labels, which turns to improve the generalization and robustness of student network. The second group of methods focus on creating pseudo labels via metadata [112], class similarities [115] or generating labels [117, 118], etc.

7.2 Label-free distillation

However, in real-world applications, labels for the data used in the student network is not always easy to obtain. Hence, some attempts have been taken for label-free distillation. We now provide more detailed analysis for these methods.

7.2.1 KD with dark knowledge.

This has inspired some works to exploit KD without the requirement of labels. Based on our review, label-free distillation is mostly achieved in cross-modal learning, as discussed in Sec. 5.3. With paired modality data (*e.g.*, video and audio), where the label of modality (*e.g.*, video) is available, the student learns the end tasks only based on the distillation loss in Eq.2 [136, 141, 144, 147, 148, 153]. That is, in this situation, the dark knowledge of teacher provides 'supervision' for the student network.

7.2.2 Creating meta knowledge.

Recently, a few methods [117, 118, 120] propose data-/label-free frameworks for KD. The core technique is to craft samples with labels by using either feature or logits information, which are also called meta knowledge. Although these methods point out an interesting direction for KD, there still exist many challenges to achieve reasonable performance.

7.3 Potential and challenges.

Label-free distillation is a promising since it relieves the need for data annotation. However, the current status of research shows that there still exist many uncertainties and challenges in this direction. The major concern is how to ensure that the 'supervision' provided by teacher is reliable enough. As some works interpret the knowledge as a way a label regularization [172] or class similarities [177], it is crucial to guarantee that the knowledge can be captured by the student.

Another critical challenge of label-required distillation is that, in Eq. 2, the KD loss term never involves any label information although the student loss (*e.g.*, cross-entropy loss) uses labels. As labels provide informative knowledge for the student learning, it is worthwhile to find a way to use labels for the KD loss to further improve the performance. While it is generally acknowledged that a pretrained teacher has already mastered sufficient knowledge about the label information, its predictions still have a considerable gap

with the ground truth labels. Based on our literature review, there exists some difficulties to bring label information to the distillation loss. That is, to bring the label information, the teacher might need to be updated or fine-tuned, which may cause additional computation cost. However, with some recent attempts based on meta learning or continual learning, it is possible to learn the label information with only a few examples. Besides, it might be possible to learn a bootstrap representation based on the labels, as done in [178], and further incorporate the information to the KD loss. We do believe this direction is promising in real-world applications and thus expect future research might move towards this direction.

8 KD WITH NOVEL LEARNING METRICS

8.1 Distillation via adversarial learning

Overall Insight: *GAN can help learn the correlation between classes and preserve the multi-modality of S-T framework, especially when student has relatively small capacity.*

In Sec. 4.1, we have discussed the two most popular approaches for KD. However, the key problem is that it is difficult for the student to learn the true data distribution from the teacher, since the teacher can not perfectly model the real data distribution. Generative adversarial networks (GANs) [78, 123, 124, 158, 180] have been proven to have potential in learning the true data distribution in image translation. To this end, recent works [62, 64, 116, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199] have tried to explore adversarial learning to improve the performance of KD. These works are, in fact, built on three fundamental prototypes of GANs [20, 78, 179]. Therefore, we formulate the principle of these three types of GANs, as illustrated in Fig. 8, and analyze the existing GAN-based KD methods.

8.1.1 A basic formulation of GANs in KD

The first type of GAN, as shown in Fig. 8(a), is proposed to generate continuous data by training a generator G and discriminator D, which penalizes the generator G for producing implausible results. The generator G produces synthetic examples $G(z)$ (*e.g.*, images) from the random noise z sampled using a specific distribution (*e.g.*, normal) [78]. These synthetic examples are fed to the discriminator D along with the real examples sampled from the real data distribution $p(x)$. The discriminator D attempts to distinguish the two inputs, and both the generator G and discriminator D improve their respective abilities in a minmax game until the discriminator D is unable to distinguish the fake from the real. The objective function can be written as follows:

$$\min_G \max_D J(G, D) = \mathbb{E}_{x \sim p(x)}[log(D(x))]+$$
$$\mathbb{E}_{z \sim p(z)}[log(1 - D(G(z)))] \quad (19)$$

where $p_z(z)$ is the distribution of noise (*e.g.*, uniform or normal).

The second type of GAN for KD is built on conditional GAN (CGAN) [123, 124, 179, 200], as shown in Fig. 8(b). CGAN is trained to generate samples from a class conditional distribution c. The generator is replaced by useful information rather than random noise. Hence, the objective

of the generator is to generate realistic data, given the conditional information. Mathematically, the objective function can be written as:

$$\min_G \max_D J(G, D) = \mathbb{E}_{x \sim p(x)}[log(D(x|c))]+$$
$$\mathbb{E}_{z \sim p(z)}[log(1 - D(G(z|c)))] \quad (20)$$

Unlike the above-mentioned GANs, triple-GAN [20] (the third type) introduces a three-player game where there is a classifier C, generator G, and discriminator D, as shown in Fig. 8(c). Adversarial learning of generators and discriminators overcomes some difficulties [78], such as not having a good optimization and the failure of the generator to control the semantics of generated samples. We assume that there is a pair of data (x, y) from the true distribution $p(x, y)$. After a sample x is sampled from $p(x)$, C assigns a pseudo label y following the conditional distribution $p_c(y|x)$, that is, C characterizes the conditional distribution $p_c(y|x) \approx p(y|x)$. The aim of the generator is to model the conditional distribution in the other direction $p_g(x|y) \approx p(x|y)$, while the discriminator determines whether a pair of data (x,y) is from the true distribution $p(x, y)$. Thus, the minmax game can be formulated as:

$$\min_{C,G} \max_D J(C, G, D) = \mathbb{E}_{(x,y) \sim p(x,y)}[log(D(x, y))]+$$
$$\alpha \, \mathbb{E}_{(x,y) \sim p_c(x,y)}[log(1 - D(x, y))]+ \quad (21)$$
$$(1 - \alpha) \, \mathbb{E}_{(x,y) \sim p_g(x,y)}[log(1 - D(G(y, z), y))]$$

where α is a hyper-parameter that controls the relative importance of C and G.

8.1.2 How does GAN boost KD?

Based on the aforementioned formulation of GANs, we analyze how they are applied to boost the performance of KD with S-T learning.

KD based on the conventional GAN (first type) Chen *et al.*[116] and Fang *et al.*[117] focused on distilling the knowledge of *logits* from teacher to student via the first type of GAN, as depicted in Fig. 8(a). [1] There are several benefits of predicting logits based on the discriminator. First, the learned loss, as described using Eqn 19, can be effective in image translation tasks [123, 124, 200]. The second benefit is closely related to the multi-modality of the network output; therefore, it is not necessary to exactly mimic the output of one teacher network to achieve good student performance as it is usually done [1, 52]. However, the low-level feature alignment is missing because the discriminator only captures the high-level statistics of the teacher and student outputs (logits).

In contrast, Belagiannis *et al.* [183], Liu *et al.* [185], Hong *et al.* [192], Aguinaldo *et al.* [198], Chung *et al.* [64], Wang *et al.* [194], Wang *et al.* [186], Chen *et al.* [203], and Li *et al.* [199] aimed to distinguish whether the *features* come from the teacher or student via adversarial learning, which effectively pushes the two distributions close to each other. [2] The features of the teacher and student are used as inputs to the discriminator because of their *dimensionality*. The feature

---

1. [116, 117] are data-free KD methods, which will be explicitly discussed in Sec. 5.1.

2. Note that in [64], least squares GAN (LSGAN) [201] loss was used and in [194], Wasserstein GAN-gradient penalty (WGAN-GP) loss [202] was used to stabilize training.
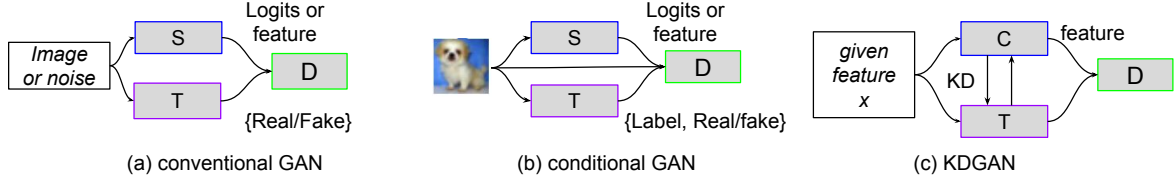
Fig. 8. An illustration of GAN-based KD methods. (a) KD based on GAN [78] where discriminator D discerns the feature/logits of T and S; (b) KD based on conditional GAN (CGAN) [179] where the input also functions as a condition to D; (c) KD based on TripleGAN [180] where the classifier C, teacher T and D play a minmax game.

TABLE 7
Taxonomy of KD based on adversarial learning.

Method	GAN type	Purpose	Inputs of D	Number of D	Online KD
Chen [116]	First type	Classification	Logits	One	No
Belagiannis [183]	First type	Classification	Features	One	No
Liu [185]	First type	Classification Object detection	Features	One	No
Hong [192]	First type	Object detection	Features	Six	No
Wang [186]	First type	Classification	Features	One	No
Aguinaldo [198]	First type	Classification	Features	One	No
Chung [64]	First type (LSGAN [201])	Classification	Features	Two/Three	Yes
Wang [194]	First type(WGAN-GP [202])	Image generation	Features	One/Multiple	Yes
Chen [203]	First/Second type	Image translation	Features	Two	No
Liu [197]	Second type (WGAN-GP [202])	Semantic segmentation	Features	One	No
Xu [182]	Second type	Classification	Logits	One	No
Roheda [187]	Second type	Cross-domain surveillance	Features	One	Yes
Zhai [193]	Second type (BicyleGAN [204])	Image translation	Features	One	Yes
Liu [184]	Second (AC-GAN [205])	Image translation	Features	One	No
Wang [181]	Third type	Image translation	Features	One	No
Li [199]	First/Second type	Image translation	Features	One	No
Fang [117]	First type	Classification Semantic segmentation	Logits	One	No
Yoo [119]	Second type	Classification	Logits	One	No

representations extracted from the teacher are high-level abstract information and easy for classification, which lowers the probability for the discriminator to make a mistake [185]. However, the GAN training in this setting is sometimes unstable and even difficult to converge, particularly when the model capacity between the student and teacher is large. To address this problem, some regularization techniques such as dropout [24] or l_2 or l_1 regularization [183] are added to Eqn. 19 to confine the weights.

KD based on CGAN (second type) Xu *et al.* [206] and Yoo *et al.* [119] employed CGAN [179] for KD, where the discriminator was trained to distinguish whether the *label distribution* (logits) was from the teacher or the student. The student, which was regarded as the generator, was adversarially trained to deceive the discriminator. Liu *et al.* [184] also exploited CGAN for compressing image generation networks. However, the discriminator predicted the class label of the teacher and student, together with an auxiliary classifier GAN [205].

In contrast, Roheda *et al.* [187], Zhai *et al.* [193], Li *et al.* [199], Chen *et al.* [203], and Liu *et al.* [197] focused on discriminating the *feature* space of the teacher and student in the CGAN framework. Interestingly, Chen *et al.* [203] deployed two discriminators, namely, the teacher and student

discriminators,for compressing image translation networks. To avoid model collapse, Liu *et al.* [197] used Wasserstein loss [202] to stabilize training.

KD based on TripleGAN (third type) In contrast to the distillation methods based on conventional GAN and CGAN , Wang *et al.* [181] proposed a three-player game named KDGAN, consisting of a classifier (tent), a teacher, and a discriminator (similar to the prototype in TripleGAN [20]), as shown in Fig. 8(c). The classifier and the teacher learn from each other via distillation losses, and are adversarially trained against the discriminator via the adversarial loss defined in Eqn. 21. By simultaneously optimizing the distillation and the adversarial loss, the classifier (student) learns the true data distribution at equilibrium.

8.1.3 Summary and open challenges

In Table 7, we summarize existing GAN-based knowledge distillation methods regarding the practical applications, input features of the discriminator D, the number of discriminators used, and whether it is one-stage (without the need for the teacher to be trained first). In general, most methods focus on classification tasks based on the first type of GAN (conventional GAN) [78] and use the features as the inputs to the discriminator D. Besides, it is worth noting that

most methods use only one discriminator for discerning the student from the teacher. However, some works such as [64], [194] and [203] employ multiple discriminators in their KD frameworks. One can see that most methods follow a two-stage KD paradigm where the teacher is trained first, and then knowledge is transferred to the student via KD loss. In contrast, studies such as [64, 187, 193, 194] also exploit online (one-stage) KD, without the necessity of pre-trained teacher networks. *More detailed analyses of KD methods with respect to online/two-stage distillation and image translation are described in Sec. 6.1 and Sec. 9.5, respectively.*

Open challenges: The first challenge for GAN-based KD is the stability of training, especially when the capacity between the teachers and the students is large. Secondly, it is less intuitive whether using only logits or only features or both as inputs to the discriminator is good because there lacks theoretical support. Thirdly, the advantages of using multiple discriminators are less clear and what features in which position are suitable for training GAN also needs to be further studied.

8.2 Distillation with graph representations

Overall insight: *Graphs are the most typical locally connected structures that capture the features and hierarchical patterns for KD.*

Up to now, we have categorized and analyzed the most common KD methods using either logits or feature information. However, one critical issue regarding KD is data. In general, training a DNN requires embedding a high-dimensional dataset to facilitate data analysis. Thus, the optimal goal of training a teacher model is not only to transform the training dataset into a low-dimensional space, but also to analyze the intra-data relations [207, 208]. However, most KD methods do not consider such relations. Here, we introduce the definitions of the basic concepts of graph embedding and knowledge graph based on [209, 210]. We provide an analysis of existing graph-based KD methods, and discuss new perspectives about KD.

8.2.1 Notation and definition

Definition 1. *A **graph** can be depicted as $\mathcal{G} = (V, E)$, where $v \in V$ is a node and $e \in E$ is an edge. A graph $\mathcal{G}$ is associated with a node type mapping function $F_v: V \rightarrow \mathcal{T}^v$, and an edge type mapping function $F_e: E \rightarrow \mathcal{T}^e$.*

Here, $\mathcal{T}^v$ and $\mathcal{T}^e$ denote the node types and edge types, respectively. For any $v_i \in V$, there exists a particular mapping type: $F_v(v_i) \in \mathcal{T}^v$. Similar mapping comes to any $e_{ij} \in E$, which is mapped as $F_e(e_{ij}) \in \mathcal{T}^e$, where i and j indicate the i-th and j-th nodes.

Definition 2. *A **homogeneous graph (directed graph)**, depicted as $\mathcal{G}_{hom} = (V, E)$, is a type of graph in which $|\mathcal{T}^v| = |\mathcal{T}^e| = 1$. All nodes and edges in this graph embedding are of one type.*

Definition 3. *A **knowledge graph**, defined as $\mathcal{G}_{kn} = (V, E)$, is an instance of a directed heterogeneous graph whose nodes are entities, and edges are subject-property-object triplets. Each edge has the form: head entity, relation, tail entity, denoted as $< h, r, t >$, indicating a relationship from a head h to a tail t.*

TABLE 8
A summary of notations used in Sec. 8.2.

Notations	Descriptions
$|\cdot|$	The cardinally of a set
$\mathcal{G} = (V, E)$	Graph $\mathcal{G}$ with a set of node V and set of edge E
v_i, e_{ij}	A node $v_i \in V$ and an edge e_{ij} linking v_i and v_j
$\mathbf{x}_{v_i}, \mathbf{x}_{e[v_i]}$	Features of v_i and features of edges of v_i
$\mathbf{h}_{ne[v_i]}, \mathbf{x}_{ne[v_i]}$	Features of states and of neighboring nodes of v_i
$F_v(v_i), F_e(e_{ij})$	Mapping of node type v_i and edge type e_{ij}
$\mathcal{T}^v, \mathcal{T}^e$	The set of node types and set of edge types
$< h, r, t >$	Head, relation and tail in knowledge graph
N	Number of nodes in the graph
$\mathbf{h}_{v_i}$	Hidden state of i-th node v
f_t, f_o	local transition and output functions
F_t, F_o	global transition and output functions
$\mathbf{H}, \mathbf{O}, \mathbf{X}$	Stack of all hidden states, outputs, features
$\mathbf{H}^t$	Hidden state of t-th iteration of $\mathbf{H}$

Fig. 9. An example of knowledge graph.

$h, t \in V$ are entities and $r \in E$ is the relation. Hereby, we note $< h, r, t >$ as a triplet for knowledge graph. An example is shown in Fig. 9. The knowledge graph includes two triplets $< h, r, t >$: $< Los Angeles, IsCityOf, California >$ and $< California, isStateOf, US >$.

Graph neural networks. A graph neural network (GNN) is a type of DNN that operates directly on the graph structure. A typical application is about node classification [211]. In the node classification problem, the i-th node v_i is characterized by its feature x_{v_i}, and ground truth t_{v_i}. Thus, given a labeled graph $\mathcal{G}$, the goal is to leverage the labeled nodes to predict the unlabeled ones. It learns to represent each node with a d dimensional vector state h_{v_i} containing the information of its neighborhood. Specifically speaking, h_{v_i} can be mathematically described as [174]:

$$\mathbf{h}_{v_i} = f_t(\mathbf{x}_{v_i}, \mathbf{x}_{co[v_i]}, \mathbf{h}_{ne[v_i]}, \mathbf{x}_{ne[v_i]}) \tag{22}$$

$$\mathbf{o}_{v_i} = f_o(\mathbf{h}_{v_i}, \mathbf{x}_{v_i}) \tag{23}$$

where $\mathbf{x}_{co[v_i]}$ denotes the feature of the edges connected with v_i, $\mathbf{h}_{ne[v_i]}$ denotes the embedding of the neighboring nodes of $\mathbf{v}_i$, and $\mathbf{x}_{ne[v_i]}$ denotes the features of the neighboring nodes of v_i. The function f_t is a transition function that projects these inputs onto a d-dimensional space, and f_o is the local output function that produces the output. Note that f_t and f_o can be interpreted as the feedforward neural networks. If we denote $\mathbf{H}, \mathbf{O}, \mathbf{X}$ and $\mathbf{X}_N$ as the concatenation of the outputs of stacking all the states, all the outputs, all the features, and all the node features, respectively, then $\mathbf{H}$ and $\mathbf{O}$ can be formulated as:

$$\mathbf{H} = F_t(\mathbf{H}, \mathbf{X}) \tag{24}$$

$$\mathbf{O} = F_o(\mathbf{H}, \mathbf{X}) \tag{25}$$

where F_t is the global transition function and F_o is the global output function. Note that F_t and F_o are the stacked functions of f_t and f_o, respectively, in all nodes V in the graph.

Since we are aiming to get a unique solution for $\mathbf{h}_{v_i}$, in [174, 211], a neighborhood aggregation algorithm is applied, such that:

$$\mathbf{H}^{t+1} = F_t(\mathbf{H}^t, \mathbf{X}) \tag{26}$$

where $\mathbf{H}^t$ denotes t-th iteration of $\mathbf{H}$.

Given any initial state $\mathbf{H}(0)$, $\mathbf{H}^{t+1}$ in Eqn. 26 convergences exponentially to the solution in Eqn. 24. Based on the framework, f_t and f_o can be optimized via supervised loss when the target information t_v^i is known:

$$\mathcal{L} = \sum_{i=1}^{N} (t_v^i - o_v^i) \tag{27}$$

where N is the total number of supervised nodes in the graph.

8.2.2 Graph-based distillation

Based on the above explanation regarding the fundamentals of graph representations and GNN, we now delve into the existing graph-based distillation techniques. To our knowledge, Liu *et al.* [216] first introduced a graph-modeling approach for visual recognition task in videos. Videos are action models modeled initially as bag of visual words (BoVW), which is sensitive to visual changes. However, some higher-level features are shared across views, and enable connecting the action models of different views. To better capture the relationship between two vocabularies, they construct a bipartite graph $\mathcal{G} = (V, E)$ to partition them into visual-word clusters. Note that V is the union of vocabularies V_1 and V_2, and E are the weights attached to nodes. In this way, knowledge from BoVW can be transferred to visual-word clusters, which are more discriminative in the presence of view changes. Luo *et al.* [217] consider incorporating rich, privileged information from a large-scale multimodal dataset in the source domain, and improve the learning in the target domain where training data and modalities are scarce. Regarding using S-T structures for KD, to date, there are several works such as [173, 207, 208, 212, 213, 214, 215, 218].

GKD [212] and IRG [208] consider the geometry of the perspective feature spaces by reducing intra-class variations, which allow for dimension-agnostic transfer of knowledge. This perspective is the opposite of Liu *et al.* [208] and RKD [28]. Specifically, instead of directly exploring the mutual relation between data points in students and teachers, GKD [212] regards this relation as a geometry of data space (see Fig. 10(a)). Given a batch of inputs $\mathbf{X}$, we can compute the inner representation $\mathbf{X}_l^S = [\mathbf{x}_l^S, \mathbf{x} \in \mathbf{X}]$ and $\mathbf{X}_l^T = [\mathbf{x}_l^T, \mathbf{x} \in \mathbf{X}]$ at layer l ($l \in \Lambda$) of the teacher and student networks. Using cosine similarity metric, these representations can be used to build a k-nearest neighbor similarity graph for the teacher $\mathcal{G}_l^T(\mathbf{X}) = <\mathbf{X}_l^T, \mathbf{W}_l^T>$, and for the student $\mathcal{G}_l^S(\mathbf{X}) = <\mathbf{X}_l^S, \mathbf{W}_l^S>$. Note that $\mathbf{W}_l^T$ and $\mathbf{W}_l^S$ represent the edge weights, which represent the similarity between the i-th and j-th elements of $\mathbf{X}_l^T$ and $\mathbf{X}_l^S$. Based on graph representation for both the teacher and the student, the KD loss in Eqn. 10 can be updated as follows:

$$\mathcal{L} = \sum_{l \in \Lambda} D\left(\mathcal{G}_l^S(\mathbf{X}), \mathcal{G}_l^T(\mathbf{X})\right) \tag{28}$$

where the distance metric D is based on L_2 distance.

IRG [208] essentially is similar to GKD [212] in the construction of the graph, however, IRG also takes into account the instance of graph transformations. The aim of introducing feature space transformation across layers is because there may be too tight or dense constraints and descriptions fitting on the teacher's instance features at intermediate layers. The transformation of the instance relation graph is composed of vertex transformation and edge transformation from the $l_1 l$-th layer to l_2-th layer, as shown in Fig. 10 (b). Thus the loss in Eqn. 28 can be extended to:

$$\mathcal{L} = \sum_{l \in \Lambda} D_1(\mathcal{G}_l^S(\mathbf{X}), \mathcal{G}_l^T(\mathbf{X})) + \\ D_2\left((\Theta_T(\mathcal{G}_l^S(\mathbf{X})), \Theta_S(\mathcal{G}_l^T(\mathbf{X}))\right) \tag{29}$$

where Θ_T and Θ_S are the transformation functions for the teacher and the student, respectively, and D_1 and D_2 are the distance metrics for instance relation and instance translation.

MHKD [207] is a method that enables distilling data based knowledge from a teacher network to a graph using an attention network (see Fig. 10(d)). Like IRG [18], feature transformation is also considered to capture the intra-data relations. The KD loss is based on the KL-divergence loss using the embedded graphs from the teacher and the student. KTG [213] also exploits graph representation; however, it focuses on a different perspective of KD. The knowledge transfer graph provides a unified view of KD, and has the potential to represent diverse knowledge patterns. Interestingly, each node in the graph represents the direction of knowledge transfer. On each edge, a loss function is defined for transferring knowledge between two nodes linked by each edge. Thus, combining different loss functions can represent collaborative knowledge learning with pair-wise knowledge transfer. Fig. 10(c) shows the knowledge graph of diverse collaborative distillation with three nodes, where $L_{s,t}$ represents the loss function used for the training node.

In addition, GFL [214], HGKT [173], GRL [215] and MHGD [207] all resort to GNN for the purpose of KD. HGKT and GFL focus on transferring knowledge from seen classes to unseen classes in few-shot learning [223, 224]. GFL [214] leverages the knowledge learned by the auxiliary graphs to improve semi-supervised node classification in the target graph. As shown in Fig. 10(e), GFL learns the representation of a whole graph, and ensures the transfer of a similarly structured knowledge. Auxiliary graph reconstruction is achieved by using a graph autoencoder. HGTK aims to build a heterogeneous graph focusing on transferring intra-class and inter-class knowledge simultaneously. Inspired by modeling class distribution in adversarial learning [78, 123, 181, 182, 189, 225], in which instances with the same class are expected to have the same distribution, the knowledge is transferred from seen classes to new unseen classes based on learned aggregation and embedding functions, as shown in Fig. 10 (f). GRL [215] builds a multi-task KD method for representation learning based on DeepGraph [222]. This knowledge is based on GNN, and maps raw graphs to metric values. The learned graph metrics are then used as auxiliary tasks, and the knowledge
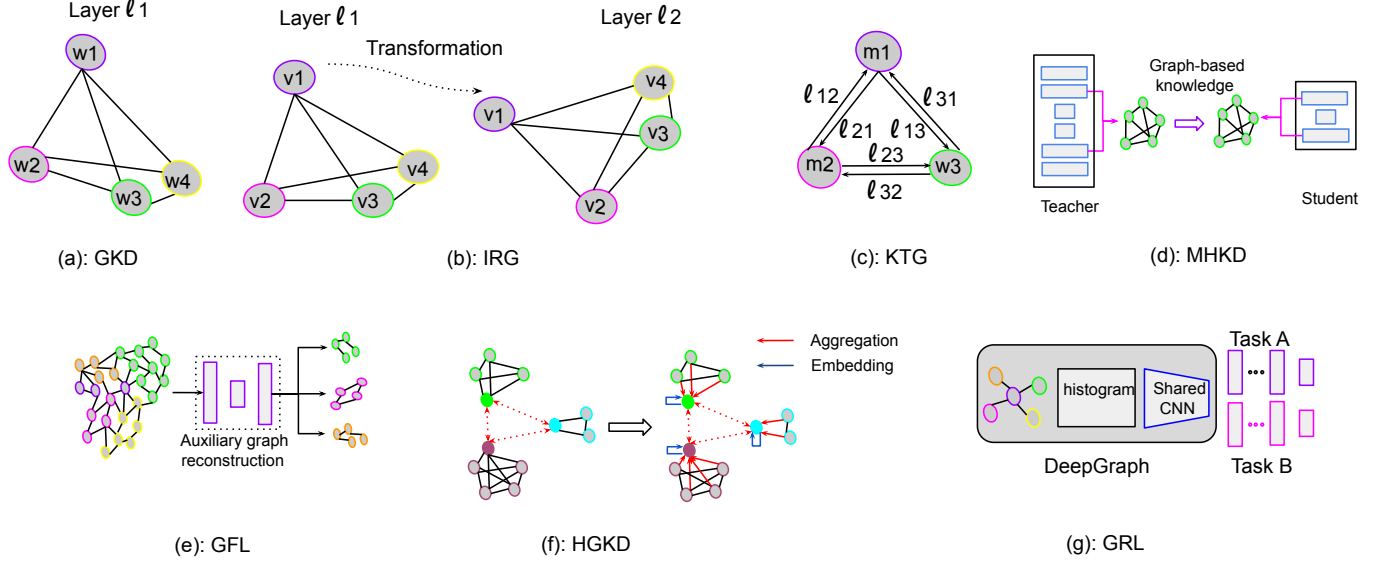
Fig. 10. A graphical illustration of graph-based KD methods. GKD [212], IRG [208], KTG [213], MHKD [207] all focus on graph-based knowledge distillation for model compression. GFL [214] and HGKD [173] aim to improve semi-supervised node classification via graph-based knowledge transfer, whereas GRL [215] exploits graph-based knowledge for multi-task learning.

TABLE 9
A summary of KD methods via graph representations.

Method	Purpose	Graph type	Knowledge type	Distance metric	Graph embedding
GKD [212]	Model compression	Heterogeneous graph	Layer-wise feature	L_2	GSP [219]
IRG [208]	Model compression	Knowledge graph	Middle layers	L_2	Instance relations
MHKD [207]	Model compression	Knowledge graph	Middle layers	KL	SVD [71] + Attention
KTG [213]	Model compression	Directed graph	Network model	L_1 + KP loss	–
GFL [214]	Few-shot learning	GNN	Class of nodes	Frobenius norm	HGR [220]
HGKD [173]	Few-shot learning	GNN	Class of nodes	Wasserstein	GraphSAGE [221]
GRL [215]	Multi-task leaning	GNN	Class of nodes	Cross-entropy	HKS [222]
Yang [218]	Model compression	GNN	Topological info.	KL	Attention

of the network is distilled into graph representations (see Fig. 10(g)). The graph representation structure is learned via a CNN by feeding the graph descriptor to it. We denote pairs of graph and graph-level labels as $\{(G_i, y_i)\}_{i=1}^{N}$, where $G_i \in \mathcal{G}$, $y_i \in \mathcal{Y}$, and $\mathcal{G}$, $\mathcal{Y}$ are the cardinal of all possible graphs and labels respectively. Then, the loss for learning the model parameters are described as:

$$\mathcal{L} = \mathbb{E}[D(y_i, f(G_i; \theta))] \qquad (30)$$

where θ are the model parameters.

Open challenges Graph representations are of significant importance for tackling KD problems because they better capture the hierarchical patterns in locally connected structures. However, there are some challenges. Firstly, graph representations are difficult to generalize because they are limited to structured data or specific types of data. Secondly, it is challenging to measure graph distances appropriately, since existing distance measure (*e.g.*, l_2) may not fit well. Thirdly, layer-wise distillation is difficult to achieve in graph KD, because graph representation models and network structures in such cases are limited.

8.3 KD for semi-/self-supervised learning

Overall insight: *KD with S-T learning aims to learn a rich representation by training a model with a large number of unlabeled*

datasets, and limited amount of labeled data.

Semi-supervised learning usually handles the problem of over-fitting due to the lack of high-quality labels of training data. To this end, most methods apply S-T learning that assumes a dual role as a teacher and a student. The student model aims to learn the given data as before, and the teacher learns from the noisy data and generates predicted targets. These are then transferred to the student model via consistency cost. In self-supervised learning, the student itself generates knowledge to be learned via various approaches, and the knowledge is then transferred by the student to itself via distillation losses. We now provide a detailed analysis of the technical details of the existing methods.

8.3.1 Semi-supervised learning

The baseline S-T frameworks for semi-supervised learning was initialized by Laine *et al.* [226] and Tarvainen *et al.* [2], as illustrated in Fig. 1(b). The student and the teacher models have the same structures, and the teacher learns from noise and transfers knowledge to the student via consistency cost. Interestingly, in [2], the teacher's weights are updated using the earth moving average (EMA) of the student's weights. Inspired by [2], Luo *et al.* [227], Zhang *et al.* [228], French *et al.* [229], Choi *et al.* [230], Cai *et al.* [231] and Xu *et al.* [232]

all employ similar frameworks where the teacher's weights are updated using exponential moving average (EMA) of the student. However, Ke *et al.*[157] mention that using a coupled EMA teacher is not sufficient for the student, since the degree of coupling increases as the training goes on. To tackle this problem, the teacher is replaced with another student, and two students are optimized individually during training while a stabilization constraint is provided for knowledge exchange (similar to mutual learning [34]).

Instead of taking independent weights between the teacher and the student, Hailat *et al.* [233] employ weight-sharing, in which the last two fully connected layers of the teacher and the student are kept independent. The teacher model plays the role of teaching the student, stabilizing the overall model, and attempting to clean the noisy labels in the training dataset. In contrast, Gong *et al.* [104] and Xie *et al.* [21] follow the conventional distillation strategy proposed by [1], where a pretrained teacher is introduced to generate learnable knowledge using unlabeled data, and utilizes it as privileged knowledge to teach the student on labeled data. However, during learning of the student, Xie *et al.* inject noise (*e.g.*, dropout) to the student such that it learns better than the teacher. Papernot *et al.* [38] propose to distill from multiple teachers (an ensemble of teachers) on a disjoint subset of sensitive data (augmented with noise) and to aggregate the knowledge of teachers to guide the student on query data.

8.3.2 Self-supervised learning

Distilling knowledge for self-supervised learning aims to preserve the learned representation for the student itself, as depicted in Fig. 1(c). Using pseudo labels is the most common approach, as done in [71, 234]. Specifically, Lee *et al.* [71] adopt self-supervised learning for KD, which not only ensures the transferred knowledge does not vanish, but also provides an additional performance improvement. In contrast, Noroozi *et al.* [234] propose to transfer knowledge by reducing the learned representation (from a pretrained teacher model) to *pseudo-labels* (via clustering) on the unlabeled dataset, which are then utilized to learn a smaller student network. Another approach is based on data augmentation (*e.g.*, rotation, cropping, color permutation) [164, 166, 235], which has been mentioned in Sec. 6.2.3. In contrast to making the 'positive' and 'negative' (augmented) examples, BYOL [178] directly bootstraps the representations with two neural networks, referred to as online and target networks, that interact and learn from each other. This spirit is somehow similar to mutual learning [34]; however, BYOL trains its online network to predict the target network's representation of another augmented view of the same image. The promising performance of BYOL might point out a new direction of KD with self-supervised learning via representation boostrap rather than using negative examples.

8.3.3 Potentials and open challenges

Based on the technical analysis for the KD methods in semi-/self-supervised learning, it is noticeable that online distillation is the mainstream. However, there are several challenges. First, as pointed by [157], using EMA for updating teacher's weights might lead to less optimal learning of knowledge. Second, no methods attempt to exploit the rich feature knowledge from teacher models. Third, data augmentation methods in these distillation methods are less effective compared to those proposed in Sec. 6.2, in which the advantages of adversarial learning are distinctive. Fourth, the representations of knowledge in these methods are limited and less effective. BYOL [178] opens a door for representation boost, and there exists a potential to further bind this idea with KD in the further research. Moreover, it has the potential to exploit a better-structured data representation approach, such as in GNNs. With these challenges, the future directions of KD for semi-/self-supervised learning could gain inspirations from exploiting feature knowledge and more sophisticated data augmentation methods together with more robust representation approaches.

8.4 Few-shot learning

Insight:*Is it possible possible to learn an effective student model to classify unseen classes (query sets) by distilling knowledge from a teacher model with the support set?*

In contrast to the methods discussed in Sec. 5.2 focusing on distillation with a few samples for training a student network (without learning to generalize to new classes), this section stresses on analyzing the technical details of few-shot learning with KD. Few-shot learning is to classify new data, having seen from only a few training examples. Few-shot learning itself is a meta-learning problem in which the DNN learns how to learn to classify, given a set of training tasks, and evaluate using a set of test tasks. Here, the goal is to discriminate between N classes with K examples of each (so-called N-way-K-shot classification). In this setting, these training examples are known as the *support set*. In addition, there are further examples of the same classes, known as a *query set*. The approaches for learning prior knowledge of a few-shot are usually based on *three* types: prior knowledge about similarity, prior knowledge about learning procedure, and prior knowledge about data. We now analyze the KD methods for few-shot learning [28, 46, 126, 236, 237] that have been recently proposed.

Prior knowledge about similarity: Park *et al.* [28] propose distance-wise and angle-wise distillation losses. The aim is to penalize the structural differences in relation to the learned representations between the teacher and the student for few-shot learning.

Prior knowledge about learning procedure: [236, 237] tackle the second type of prior knowledge, namely learning procedure. To be specific, Flennerhag *et al.* [236] focuses on transferring knowledge across the learning process, in which the information from previous tasks is distilled to facilitate the learning on new tasks. However, Jin *et al.* [237] address the problem of learning a meta-learner that can automatically learn what knowledge to transfer from the source network to where in the target network.

Prior knowledge about data: Dvornik *et al.* [46] and Liu *et al.* [126] address the third type of prior knowledge, namely data variance. To be specific, in [46], en ensemble of several teacher networks is elaborated to leverage the variance of the classifiers and encouraged to cooperate while encouraging the diversity of prediction. However, in [126], the goal is to preserve the knowledge of the teacher (*e.g.*, intra-class

relationship) learned at the pretraining stage by generating pseudo labels for training samples in the fine-tuning set.

8.4.1 What's challenging?

Based on our analysis, the existing techniques actually expose crucial challenges. First, the overall performance of KD-based few-shot learning is convincing, but the power of meta-learning is somehow degraded or exempted. Second, transferring knowledge from multi-source networks is a potential, but identifying what to learn and where to transfer is heavily based on the meta-learner, and selecting which teacher to learn is computation-complex. Third, all approaches focus on a task-specific distillation, but the performance drops as the domain shifts. Thus, future works may focus more on handling these problems.

8.5 Incremental Learning

Overall insight: *KD for incremental learning mainly deals with two challenges: maintaining the performance on old classes, and balancing between old and new classes.*

Incremental learning investigates learning the new knowledge continuously to update the model's knowledge, while maintaining the existing knowledge [238]. Many attempts [102, 169, 193, 238, 239, 240, 241] have been made to utilize KD in addressing the challenge of maintaining the old knowledge. Based on the number of teacher networks used for distillation, these methods can be categorized into two types: distillation from a single teacher and distillation from multiple teachers.

8.5.1 Distillation from a single teacher

Shmelkov *et al*. [241], Wu *et al*. [238], Michieli *et al*. [239] and Hou *et al*. [169] focus on learning student networks for new classes, by distilling knowledge (logits information) from pretrained teachers on old-class data. Although these methods vary in tasks and distillation process, they follow similar S-T structures. Usually, the pretrained model is taken as the teacher, and the same network or a different network is employed to adapt for new classes. Michieli *et al*. exploit the intermediate feature representations and transfer them to the student.

8.5.2 Distillation from multiple teachers

Castro *et al*. [240], Zhou *et al*. [102] and Ammar *et al*. [242] concentrate on learning an incremental model with multiple teachers. Specifically, Castro *et al*. share the same feature extractor between teachers and the student. The teachers contain old classes, and their logits are used for distillation and classification. Interestingly, Zhou *et al*. propose a multi-model and multi-level KD strategy in which all previous model snapshots are leveraged to learn the last model (student). This approach is similar to born-again KD methods, as mentioned in Sec. 6.2, where the student model at the last step, and is updated using the assembled knowledge from all previous steps. However, the assembled knowledge also depends on the intermediate feature representations. Ammar *et al*. develop a cross-domain incremental RL framework, in which the transferable knowledge is shared and projected to different task domains of the task-specific student peers.

8.5.3 Open challenges

The existing methods rely on multi-step training (offline). However, it will be more significant if the online (one-step) distillation approaches can be utilized to improve the learning efficiency and performance. Moreover, existing methods require accessing the previous data to avoid ambiguities between the update steps. However, the possibility of data-free distillation methods remains open. Furthermore, existing methods only tackle the incremental learning of new classes in the same data domain, but it will be fruitful if cross-domain distillation methods can be applied in this direction.

8.6 Reinforcement learning

Overall insight: *KD in reinforcement learning is to encourage polices (such as students) in the ensemble to learn from the best policies (such as teachers), thus enabling rapid improvement and continuous optimization.*

Reinforcement learning (RL) is a learning problem that trains a policy to interact with the environment in way that yields maximal reward. To use the best policy to guide other policies, KD has been employed in [18, 107, 236, 243, 244, 245, 246, 247]. Based on the specialties of these methods, we divide them into three categories, and provide an explicit analysis. *We assume familiarity with basics of RL, and skip the definitions of Deep Q-network and A3C.*

8.6.1 Collaborative distillation

Xue *et al*. [246], Hong *et al*. [245], and Lin *et al*. [247] focus on collaborative distillation, which is similar to mutual learning [34]. In Xue *et al*., the agents teach each other based on the reinforcement rule, and teaching occurs between the value function of the agents (students and teachers). Note that the knowledge is provided by a group of student peers periodically, and assembled to enhance the learning speed and stability as in [245]. However, Hong *et al*.[245] periodically distill the best-performing policy to the rest of the ensemble. Lin *et al*. stress on collaborative learning among heterogeneous learning agents, and incorporate the knowledge into online training.

8.6.2 Model compression with RL-based distillation

Ashok *et al*. [243] tackle the problem of model compression via RL. The method takes a larger teacher network, and outputs a compressed student network derived from it. In particular, two recurrent policy networks are employed to aggressively remove layers from the teacher network, and to carefully reduce the size of each remaining layer. The learned student network is evaluated by a reward, which is a score based on the accuracy and compression of the teacher.

8.6.3 Random network distillation

Burda *et al*. [244] focus on a different perspective where the prediction problem is randomly generated. The approach involves two networks: the target (student) network, which is fixed and randomly initialized, and a predictor (teacher) network trained on the data collected by the agent. With the knowledge distilled from the predictor, the target network tends to have lower prediction errors. Rusu *et al*. [107]

also apply random initialization for the target network. However, they focus more on online learning action policies, which can be either single-task or muti-task.

8.6.4 Potentials of RL-based KD

We have analyzed existing RL-based KD methods in detail. Especially, we notice that model compression via RL-based KD is promising due to its extraordinary merits. First, RL-based KD better addresses the problem of *scalability* of network models. This is similar to neural architecture search (NAS). Moreover, the reward functions in RL-based KD *better balance* the accuracy-size trade-off. It is also possible to transfer knowledge from a *smaller model to a larger model*, which is a distinctive advantage over other KD methods.

9 APPLICATIONS FOR VISUAL INTELLIGENCE

9.1 Semantic and motion segmentation

Insight: *Semantic segmentation is a structured problem, and structure information (e.g., spatial context structures) needs to be taken into account when distilling knowledge for semantic segmentation networks.*

Semantic segmentation is a special classification problem that predicts the category label in a pixel-wise manner. As existing the state-of-the-art (SOTA) methods such as fully convolutional networks [248] have large model sizes and high computation costs, some methods [117, 152, 159, 197, 239, 249, 250, 251, 252] have been proposed to train lightweight networks via KD. Although these methods vary in their learning methods, most of them share the same distillation frameworks. Particularly, Xie *et al*. [252], Shan *et al*. [249], and Michieli *et al*. [239] focused on pixel-wise, feature-based distillation methods. Moreover, Liu *et al*.[197] and He *et al*. [250] both exploited affinity-based distillation strategy using intermediate features. Liu *et al*. also employed pixel-wise and holistic KD losses via adversarial learning. In contrast, Dou *et al*. [152] focused on unpaired multi-modal segmentation, and proposed an online KD method via mutual learning [34]. Chen *et al*. [251] proposed a target-guided KD approach to learn the real image style by training the student to imitate a teacher trained with real images. Mullapudi *et al*. [159] trained a compact video segmentation model via online distillation, in which a teacher's output was used as a learning target to adapt the student and select the next frame for supervision.

9.2 KD for visual detection and tracking

Insight: *Challenges such as regression, region proposals, and less voluminous labels must be considered when distilling visual detectors.*

Visual detection is a crucial high-level task in computer vision. Speed and accuracy are two key factors for visual detectors. KD is a potential choice to achieve sped-up and lightweight network models. However, applying distillation methods to detection is more challenging than applying classification methods. First, detection performance degrades seriously after compression. Second, detection classes are not equally important, and special considerations for distillation have to be taken into account. Third, domain and data generalization has to be considered for a distilled detector. To overcome these challenges, several impressive KD methods [65, 175, 192, 237, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264] have been proposed for compressing visual detection networks. We categorize these methods according to their specialties (*e.g.*, pedestrian detection).

9.2.1 Generic object detection

[65, 175, 192, 253, 254, 256, 258, 265, 266] aimed to learn lightweight object detectors with KD. Among these works, Chen *et al*. [254] and Hao *et al*. [253] highlighted learning a class-incremental student detector by following the generic KD framework (from a pretrained teacher). However, novel object detection losses were adopted as strong impetus for learning new classes. These losses handled classification results, location results, the detected region of interest, and all intermediate region proposals. Moreover, Chen *et al*. [175] learned a student detector by distilling knowledge from the intermediate layer, logits, and regressor of the teacher, in contrast to [65], in which only the intermediate layer of the teacher was utilized based on fine-grained imitation masks to identify informative locations. Jin *et al*. [258], Tang *et al*. [256], and Hong *et al*.[192] exploited multiple intermediate layers as useful knowledge. Jin *et al*. designed an uncertainty-aware distillation loss to learn the multiple-shot features from the teacher network. However, Hong *et al*.and Tang *et al*. were based on one-stage KD (online) via adversarial learning and semi-supervised learning, respectively. In contrast, Liu *et al*. [266] combined single S-T learning and mutual learning of students for learning lightweight tracking networks.

9.2.2 Pedestrian detection

While pedestrian detection is based on generic object detection, various sizes and aspect ratios of pedestrians under extreme illumination conditions are challenges. To learn an effective lightweight detector, Chen *et al*. [255] suggested using the unified hierarchical knowledge via multiple intermediate supervisions, in which not only the feature pyramid (from low-level to high-level features) and region features, but also the logits information were distilled. Kruthiventi *et al*. [261] learned an effective student detector in challenging illumination conditions by extracting dark knowledge (both RGB and thermal-like hint features) from a multi-modal teacher network.

9.2.3 Face detection

Ge *et al*. [262] and Karlekar *et al*. [267] compressed face detectors to recognize low-resolution faces via selective KD (last hidden layer) from teachers which were initialized to recognize high-resolution faces. In contrast, Jin *et al*. [237], Luo *et al*. [263], and Feng *et al*. [264] used single type of image. Jin *et al*.focused on compressing face detectors by using the supervisory signal from the classification maps of teacher models and regression maps of the ground truth. They identify that it is better to learn a classification map of a larger model than that of smaller models. Feng *et al*.presented a triplet KD method to transfer knowledge from a teacher model to a student model, in which a triplet of samples, the anchor image, positive image, and negative

image, was used. The purpose of the triplet loss was to minimize the feature similarity between the anchor and positive images, while maximizing that between the anchor and negative images. Luo *et al.*addressed the importance of neurons at the higher hidden layer of the teacher, and a neuron selection method was applied to select neurons that were crucial for teaching the student. Dong *et al.* [268] concentrated on the interaction between the teacher and the students. Two students learned to generate pseudo facial landmark labels, which were filtered and selected as the qualified knowledge by the teacher.

9.2.4 Vehicle detection and driving learning

Lee *et al.* [259], Saputra *et al.* [257], and Xu *et al.* [182] focused more on detection tasks for autonomous driving. In particular, Lee *et al.*focused on compressing a vehicle maker classification system based on cascaded CNNs (teacher) into a single CNN structure (student). The proposed distillation method used the feature map as the transfer medium, and the teacher and student were trained in parallel (online distillation). Although the detection task was different, Xu *et al.* build a binary weight Yolo vehicle detector by mincing the feature maps of the teacher network from easy tasks to difficult ones progressively. Zhao *et al.* [269] exploited an S-T framework to encourage the student to learn the teacher's sufficient and invariant representation knowledge (based on semantic segmentation) for driving.

9.2.5 Pose detection

Distilling human pose detectors has several challenges. First, lightweight detectors have to deal with arbitrary person images/videos to determine joint locations with unconstrained human appearances. Second, the detectors must be robust in viewing conditions and background noises. Third, the detectors should have fast inference speeds, and be memory-efficient. To this end, [145, 270, 271, 272, 273, 274, 275] formulated various distillation methods. Zhang *et al.* [270] achieved effective knowledge transfer by distilling the joint confidence maps from a pre-trained teacher model, whereas Huang *et al.* [272] exploited the heat map and location map of a pretrained teacher as the knowledge to be distilled. Furthermore, Xu *et al.* [273], Thoker *et al.* [145], and Martinez *et al.* [271] focused on multi-person pose estimation. Thoker *et al.* addressed cross-modality distillation problems, in which a novel framework based on mutual learning [34] of two students supervised by one teacher was initialized. Xu *et al.* [273] learned the integral knowledge, namely, the feature, logits, and structured information via a discriminator under the standard S-T framework. Martinez *et al.* [271] trained the student to mimic the confidence maps, feature maps, and inner-stage predictions of a pre-trained teacher with depth images. Wang *et al.* [274] trained a 3D pose estimation network by distilling knowledge from non-rigid structure from motion using only 2D landmark annotations. In contrast, Nie *et al.* [275] introduced an online KD in which the pose kernels in videos were distilled by leveraging the temporal cues from the previous frame in a one-shot learning manner.

9.3 Domain adaptation

Insight: *Is it possible to distill knowledge of a teacher in one domain to a student in another domain?*

Domain adaptation (DA) addresses the problem of learning a target domain with the help of a different but related source domain [276]. Since Lopez *et al.* [277] and Gupta *et al.* [143] initially proposed the technique of transferring knowledge between images from different modalities (called generalized distillation), it is natural to ask if this novel technique be used to address the problem of DA. The challenge of DA usually comes with transferring knowledge from the source model (usually with labels) to the target domain with unlabeled data. To address the problem, several KD methods based on S-T frameworks [230, 231, 232, 276, 278, 279, 280, 281, 282] have been proposed recently. Although these methods are focused on diverse tasks, technically, they can be categorized into two types: unsupervised and semi-supervised DA via KD.

9.3.1 Semi-supervised DA

French *et al.* [229], Choi *et al.* [230], Cai *et al.* [231], Xu *et al.* [232], and Cho *et al.* [283] proposed similar S-T frameworks for semantic segmentation and object detection. These frameworks were the updated methods of Mean-Teacher [2], which is based on self-ensemble of the student networks (teacher and student models have the same structure). Note that the weights of the teacher models in these methods are the EMAs of the weights of the student models. In contrast, Choi *et al.*added a target-guided generator to produce augmented images instead of stochastic augmentation, as in [229, 231, 232]. Cai *et al.* also exploited the feature knowledge from the teacher model, and applied region-level and intra-graph consistency losses instead of the mean square error loss.

In contrast, Ao *et al.* [276] proposed a generalized distillation DA method by applying the generalized distillation information [277] to multiple teachers to generate soft labels, which were then used to supervise the student model (this framework is similar to online KD from multiple teachers as mentioned in Sec. 4.2). Cho *et al.* [283] proposed an S-T learning framework, in which a smaller depth prediction network was trained based on the supervision of the auxiliary information (ensemble of multiple depth predictions) obtained from a larger stereo matching network (teacher).

9.3.2 Unsupervised DA

Some methods such as [278, 280] distill the knowledge from the source domain to the target domain based on adversarial learning [78] and image translation [123, 124, 200]. Technically, images in the source domain are translated to images in the target domain as data augmentation, and cross-domain consistency losses are adopted to force the teacher and student models to produce consistent predictions. Tsai *et al.* [281] and Deng *et al.* [282] focused on aligning the feature similarities between teacher and student models, compared with Meng *et al.* [279], who focused on aligning softmax outputs.

9.4 Depth and scene flow estimation

Insight: *The challenges for distilling depth and flow estimation tasks come with transferring the knowledge of data and labels.*

Depth and optical flow estimations are low-level vision tasks aiming to estimate the 3D structure and motion of the scene. There are several challenges. First, in contrast to other tasks (*e.g.*, semantic segmentation), depth and flow estimations do not have class labels. Thus, applying existing KD techniques directly may not work well. Moreover, learning a lightweight student model usually requires a large amount of labeled data to achieve robust generalization capability. However, acquiring these data is very costly.

To address these challenges, Guo *et al*. [176], Pilzer *et al*. [284], and Tosi *et al*. [285] proposed distillation-based approaches to learn monocular depth estimation. These methods were focused on handling the second challenge, namely, data distillation. Specifically, Pilzer *et al*. [284] proposed an unsupervised distillation approach, where the left image was translated to the right via the image translation framework [123, 200]. The inconsistencies between the left and right images were used to improve depth estimation, which was finally used to improve the student network via KD. In contrast, Guo *et al*. and Tosi *et al*. focused on cross-domain KD, which aimed to distill the *proxy* labels obtained from the stereo network (teacher) to learn a student depth estimation network. Choi *et al*. [283] learned a student network for monocular depth inference by distilling the knowledge of depth predictions from a *stereo* teacher network via the data ensemble strategy.

Liu *et al*. [286] and Aleotti *et al*. [287] proposed data-distillation methods for scene flow estimation. Liu *et al*. distilled reliable predictions from a teacher network with unlabeled data, and used these predictions (for non-occluded pixels) as annotations to guide a student network to learn the optical flow. They proposed to leverage on the knowledge learned by the teacher networks specialized in stereo to distill proxy annotations, which is similar to the KD method for depth estimation in [176, 285]. Tosi *et al*. [288] learned a compact network for predicting holistic scene understanding tasks including depth, optical flow, and motion segmentation, based on distillation of proxy semantic labels and semantic-aware self-distillation of optical information.

9.5 Image translation

Insight: *Distilling GAN frameworks for image translation has to consider three factors: large number of parameters of the generators, no ground truth labels for training data, and complex framework (both generator and discriminator).*

Attempts were made in several works to compress GANs for image translation with KD. Aguinaldo *et al*. [198] focused on unconditional GANs, and proposed to learn a smaller student generator by distilling knowledge from the generated images of a larger teacher generator using mean squared error (MSE). However, the knowledge incorporated in the teacher discriminator was not investigated. In contrast, Chen *et al*. [203] and Li *et al*. [199] focused on conditional GANs, and exploited the knowledge from the teacher discriminator. Specifically, Chen *et al*.included a student discriminator to measure the distances between real images and images generated by the student and teacher generators. The student GAN was then trained under the supervision of the teacher GAN. In particular, Li *et al*. [199] adopted the discriminator of the teacher as the student

discriminator, and fine-tuned the discriminator together with the compressed generator, which was automatically found with significantly lower computation cost and fewer parameters, by using NAS. In contrast, Wang *et al*. [289] focused on compressing encoder-decoder based neural style transfer network via collaborative distillation (between the encoder and its decoder), where the student was restricted to learn the linear embedding of the teacher's output.

9.6 KD for Video understanding

9.6.1 Video classification and recognition

Bhardwaj *et al*. [290] and Wang *et al*. [291] employed the general S-T learning framework for video classification. The student was trained with processing only a few frames of the video, and produced a representation similar to that of the teacher. Gan *et al*. [292] focused on video concept learning for action recognition and event detection by using web videos and images. The learned knowledge from teacher network (Lead network) was used to filter out the noisy images. These were then used to fine-tune the teacher network to obtain a student network (Exceeding network). Gan *et al*. [293] explored geometry as a new type of practical auxiliary knowledge for self-supervised learning of video representations. Fu *et al*. [294] propose focused on video attention prediction by leveraging both spatial and temporal knowledge. Farhadi *et al*. [295] distill the temporal knowledge from a teacher model over the selected video frames to a student model.

9.6.2 Video captioning

[296, 297] exploited the potential of graph-based S-T learning for image captioning. Specifically, Zhang *et al*. [296] leveraged the object-level information (teacher) to learn the scene feature representation (student) via a spatio-temporal graph. Pan *et al*. [297] highlighted the importance of the relational graph connecting for all the objects in the video, and forced the caption model to learn the abundant linguistic language via teacher-recommended learning.

10 DISCUSSIONS

In this section, we discuss some fundamental questions and challenges that are crucial for better understanding and improving KD.

10.1 Are bigger models better teachers?

The early assumption and idea behind KD are that soft labels (probabilities) from a trained teacher reflect more about the distribution of data than the ground truth labels [1]. If this is true, then it is expected that as the teacher becomes more robust, the knowledge (soft labels) provided by the teacher would be more reliable and better capture the distribution of classes. That is, a more robust teacher provides constructive knowledge and supervision to the student. Thus, the intuitive approach for learning a more accurate student is to employ a bigger and more robust teacher. However, based on the experimental results in [16], it is found out that a bigger and more robust model does not always make a better teacher. As the teacher's capacity grows, the student's accuracy rises to some extent, and then

begins to drop. We summarize two crucial reasons behind the lack of theoretical support for KD, based on [16, 35].

- The student is able to follow the teacher, but it cannot absorb useful knowledge from the teacher. This indicates that there is a mismatch between the KD losses and accuracy evaluation methods. As pointed in [35], the optimization method used could have a large impact on the distillation risk. Thus, optimization methods might be crucial for significant KD to the student.
- Another reason comes from when the student is unable to follow the teacher due to the large model capacity between the teacher and the student. It is stated in [1, 53] that the S-T similarity is highly related to how well the student can mimic the teacher. If the student is similar to the teacher, it will produce outputs similar to the teacher.

Intermediate feature representations are also effective knowledge that can be used to learn the student [52, 54]. The common approach for feature-based distillation is to transfer the features into a type of representation that the student can easily learn. In such a case, are bigger models are better teachers? As pointed in [52], feature-based distillation is better than the distillation of soft labels, and deeper students perform better than shallower ones. In addition, the performance of the student increases upon increasing the number of layers (feature representations) [54]. However, when the student is fixed, a bigger teacher does not always teach a better student. When the similarity between the teacher and student is relatively high, the student tends to achieve plausible results.

10.2 Is a pretrained teacher important?

While most works focus on learning a smaller student based on the pretrained teacher, the distillation is not always efficient and effective. When the model capacity between the teacher and the student is large, it is hard for the student to follow the teacher, thus inducing the difficulty of optimization. Is a pretrained teacher important for learning a compact student with plausible performance? [34, 40] propose learning from student peers, each of which has the same model complicity. The greatest advantage of this distillation approach is efficiency, since the pretraining of a high capacity teacher is exempted. Instead of teaching, the student peers learn to cooperate with each other to obtain an optimal learning solution. Surprisingly, learning without the teacher even enables improving the performance. The question of why learning without the teacher is better has been studied in [2]. Their results indicate that the compact student may have a less chance of overfitting. Moreover, [16] suggests that early stopping of training on ImageNet [51] achieves better performance. The ensemble of students pool their collective predictions, thus helping to converge at a more robust minima as pointed in [34].

10.3 Is born-again self-distillation better?

Born-again network [30] is the initial self-distillation method in which the student is trained sequentially, and the later step is supervised by the earlier generation. At the end of the procedure, all the student generations are assembled together to get an additional gain. So is self-distillation in the *generations* better? [16] finds that network architecture heavily determines the success of KD in generations. Although the ensemble of the student models from all the generations outperforms a single model trained from scratch, the ensemble does not outperform an ensemble of an equal number of models trained from scratch.

Instead, recent works [163, 164, 168] shift the focus from sequential self-distillation (multiple stages) to the one-stage (online) manner. The student distills knowledge to itself without resorting to the teacher and heavy computation. These methods show more efficiency, less computation costs, and higher accuracy. The reason for such better performance has been pointed out in [163, 168]. They have figured out that that online self-distillation can help student models converge to flat minima. Moreover, self-distillation prevents student models from the 'vanishing gradient' problem. Lastly, self-distillation helps to extract more discriminative features. In summary, online self-distillation shows significant advantages than sequential distillation methods and is more generalizable.

10.4 Single teacher vs multiple teachers

It is noticeable that recent distillation methods turn to exploit the potential of learning from multiple teachers. Is learning from multiple teachers really better than learning from a single teacher? To answer this question, [37] intuitively identified that the student can fuse different predictions from multiple teachers to establish its own comprehensive understanding of the knowledge. The intuition behind this is that by unifying the knowledge from the ensemble of teachers, the relative similarity relationship among teachers is maintained. This provides a more integrated dark knowledge for the student. Similar to mutual learning [34, 40], the ensemble of teachers collects the individual predictions (knowledge) together, thus converging at minima that are more robust. Lastly, learning from multiple teachers relieves training difficulties such as vanishing gradient problems.

10.5 Is data-free distillation effective enough?

In the absence of training data, some novel methods [112, 116, 118, 119] have been proposed to achieve plausible results. A theoretical explanation for why such methods are robust enough for learning a portable student has yet to be proposed. These methods are focused only on classification, and the generalization capability of such methods is still low. Most works employ generators to generate the 'latent' images from noise via adversarial learning [123, 132], but such methods are relatively hard to train and computationally expensive.

10.6 Logits vs features

The knowledge defined in existing KD methods is from three aspects: logits, feature maps (intermediate layers), and both. However, it is still unclear which one of these represents better knowledge. While works such as [52, 53, 54, 55, 61] focus on better interpretation of feature representations and claim that features might contain richer information;

some other works [1, 15, 34, 115] mention that softened labels (logits) could represent each sample by a class distribution, and a student can easily learn the intra-class variations. However, it is noticeable that KD via logits has obvious drawbacks. First, its effectiveness is limited to the softmax loss function, and it relies on the number of classes (cannot be applied to low-level vision tasks). Secondly, when the capacity between the teacher and the student is big, it is hard for the student to follow the teacher's class probabilities [16]. Moreover, as studied in [61], semantically similar inputs tend to elicit similar activation patterns in teacher networks, indicating that the similarity-preserving knowledge from intermediate features express not only the representation space, but also the activations of object category (similar to class distributions). Thus, we can clearly see that features provide more affluent knowledge than logits, and generalize better to the problems without class labels.

10.7 Interpretability of KD

In Sec. 3, we provided a theoretical analysis of KD based on the information maximization theory. It is commonly acknowledged that the teacher model's dark knowledge provides the privileged information on similarity of class categories to improve the learning of students [1, 4]. However, why KD works is also an important question. There are some methods that explore the principles of KD from the view of label smoothing [172], visual concepts [177], category similarity [1], etc. Specifically, [172] found that KD is a learned label smoothing regularization (LSR), and LSR is an ad-hoc KD. Even a poorly-trained teacher can improve the student's performance, and the weak student could improve the teacher. However, the findings in [172] only focus on classification-related tasks, and these intriguing results do not apply to the tasks without labels [199, 203]. In contrast, [177] claims that KD makes DNN learn more task-related visual concepts and discard task-irreverent concepts to learn discriminative features. From a general perspective, the quantification of visual concepts in [177] provides a more intuitive interpretation for the success of KD. However, there exists a strong need that more intensive research needs to be done in this direction.

10.8 Network architecture vs effectiveness of KD.

It has been demonstrated that distillation position has a significant impact on the effectiveness of KD [16, 53]. Most methods demonstrate this by deploying the same network for both teacher and student. However, many fail to transfer across very different teacher and student architectures. Recently, [10] found that [11, 52, 57] perform poorly even on very similar student and teacher architectures. [172] also reported an intriguing finding that a poorly trained teacher also can improve the student's performance. It is thus quite imperative to excavate how network architecture affects the effectiveness of KD and why KD fails to work when the network architectures of student and teacher are different.

11 NEW OUTLOOKS AND PERSPECTIVES

In this section, we provide some ideas and discuss future directions of knowledge distillation. We take the latest deep learning methods (*e.g.*, neural architecture search

(NAS), graph neural network (GNN)), novel non-Euclidean distances (*e.g.*, hypersphere), better feature representation approaches, and potential vision applications, such as 360° vision [298] and event-based vision [158] into account.

11.1 Potential of NAS

In recent years, NAS has become a popular topic in deep learning. NAS has the potential of automating the design of neural networks. Therefore, it can be efficient for searching more compact student models. In such a way, NAS can be incorporated with KD for model compression. This has been recently demonstrated for GAN compression [199, 299]. It is shown to be effective for finding efficient student model from the teacher with lower computation costs and fewer parameters. It turns out that NAS improves the compression ratio and accelerates the KD process. A similar approach is taken by [243] learning to remove layers of teacher network based on reinforcement learning (RL). Thus, we propose that NAS with RL can be a good direction of KD for model compression. This might significantly relieve the complexity and enhance the learning efficiency of existing methods, in which the student is manually designed based on the teacher.

11.2 Potential of GNN

Although GNN has brought progress in KD learning under the S-T framework, some challenges remain. This is because most methods rely on finding structured data on which graph-based algorithms can be applied. [18] considers the instance features and instance relationships as instance graphs, and [215] builds an input graph representation for multi-task knowledge distillation. However, in knowledge distillation, there exists *non-structural* knowledge in addition to the structural knowledge (*e.g.*, training data, logits, intermediate features, and outputs of teacher), and it is necessary to construct a flexible knowledge graph to tackle the *non-structural* distillation process.

11.3 Non-Euclidean distillation measure

Existing KD losses are mostly dependent on Euclidean loss (*e.g.*, l_1), and have their own limitations. [300] has shown that algorithms that regularize with Euclidean distance, (*e.g.*MSE loss) are easily confused by random features. The difficulty arises when the model capacity between the teacher and the student is large. Besides, l_2 regularization does not penalize small weights enough. Inspired by a recent work [301] on GAN training, we propose that it is useful to exploit the information of higher-order statistics of data in non-Euclidean spaces (*e.g.*, hypersphere). This is because geometric constraints induced by the non-Euclidean distance might make training more stable, thus improving the efficiency of KD.

11.4 Better feature representations

Existing methods that focus on KD with multiple teachers show potential for handling cross-domain problems or other problems where the ground truth is not available. However, the ensemble of feature representations [48, 64, 111] is still

challenging in some aspects. One critical challenge is fusing the feature representations and balancing each of them with robust gating mechanisms. Manually assigning weights to each component may hurt the diversity and flexibility of individual feature representations, thus impairing the effectiveness of ensemble knowledge. One possible solution is attention gates, as demonstrated in some detection tasks [302, 303]. The aim of this approach is to highlight the important feature dimensions and prune feature responses to preserve only the activations relevant to the specific task. Another approach is inspired by the gating mechanism used in long short-term memory(LSTM) [304, 305]. That is, this gate unit in KD is elaborately designed to remember features across different image regions and to control the pass of each region feature as a whole by their contribution to the task (*e.g.*, classification) with the weight of importance.

11.5 A more constructive theoretical analysis

While KD shows impressive performance improvements in many tasks, the intuition behind it is still unclear. Recently, [16] explained conventional KD [1] using linear models, and [8, 9, 10] focus on explaining feature-based KD. Mobahi *et al.*[163] provides theoretical analysis for self-distillation. However, the mechanism behind data-free KD and KD from multiple teachers is still unknown. Therefore, further theoretical studies on explaining the principles of these methods should be undertaken.

11.6 Potentials for special vision problems

While existing KD techniques are mostly developed based on vision problems (*e.g.*, classification), they rarely exploit some special vision fields such as 360° vision [298] and event-based vision [124, 158, 306]. The biggest challenge for both these vision fields is the lack of labeled data, and learning in these needs a special change of inputs for neural networks. Thus, the potential of KD, particularly cross-modal KD, for these two fields is promising. By distilling knowledge from the teacher trained with RGB images or frames to the student network specialized in learning to predict 360° images or stacked event images, it not only handles the problem of lack of data, but also achieves desirable results in the prediction tasks.

11.7 Integration of vision, speech and NLP.

As a potential, it is promising to apply KD to the integrated learning problems of vision, speech, and NLP. Although recent attempts of cross-modal KD [142, 145, 146, 147] focus on transferring the knowledge from one modality (*e.g.*, video) to the other (*e.g.*, sound) on the end tasks, it is still challenging to learn the end tasks for the integration of the three modalities. The major challenge may come from collecting paired data of three modalities; however, it is possible to apply GAN or representation learning methods to unsupervised cross-modal KD for learning effective end tasks.

12 CONCLUSION

This review of KD and S-T learning has covered major technical details and applications for visual intelligence. We provide a formal definition of the problem, and introduce the taxonomy methods for existing KD approaches. Drawing connections among these approaches, we identify a new active area of research that is likely to create new methods that take advantage of the strengths of each paradigm. Each taxonomy of the KD methods shows the current technical status regarding its advantages and disadvantages. Based on explicit analyses, we then discuss methods to overcome the challenges, and break the bottlenecks by exploiting new deep learning methods, new KD losses, and new vision application fields.

REFERENCES

[1] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *NIPS Deep Learning Workshop*, 2015.

[2] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[3] S. Gutstein, O. Fuentes, and E. Freudenthal, "Knowledge transfer in deep convolutional neural nets," *International Journal on Artificial Intelligence Tools*, vol. 17, no. 03, pp. 555–567, 2008.

[4] C. Bucilua, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 535–541.

[5] https://intellabs.github.io/distiller/knowledge_distillation.html.

[6] https://github.com/FLHonker/Awesome-Knowledge-Distillation.

[7] https://github.com/dkozlov/awesome-knowledge-distillation.

[8] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] S. Hegde, R. Prasad, R. Hebbalaguppe, and V. Kumar, "Variational student: Learning compact and sparser networks in knowledge distillation framework," *ICASSP*, 2020.

[10] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *International Conference on Learning Representations*, 2019.

[11] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in neural information processing systems*, 2014, pp. 2654–2662.

[12] K. Mangalam and M. Salzamann, "On compressing u-net using knowledge distillation," *arXiv preprint arXiv:1812.00249*, 2018.

[13] Q. Ding, S. Wu, H. Sun, J. Guo, and S.-T. Xia, "Adaptive regularization of labels," *arXiv preprint arXiv:1908.05474*, 2019.

[14] J. Cho and M. Lee, "Building a compact convolutional neural network for embedded intelligent sensor systems using group sparsity and knowledge distillation," *Sensors*, vol. 19, no. 19, p. 4307, 2019.

[15] T. Wen, S. Lai, and X. Qian, "Preparing lessons: Improve knowledge distillation with better supervision," *arXiv preprint arXiv:1911.07471*, 2019.

[16] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4794–4802.

[17] C. Yang, L. Xie, C. Su, and A. L. Yuille, "Snapshot distillation: Teacher-student optimization in one generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2859–2868.

[18] I.-J. Liu, J. Peng, and A. G. Schwing, "Knowledge flow: Improve upon your teachers," *Seventh International Conference on Learning Representations*, 2019.

[19] A. Mishra and D. Marr, "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," *arXiv preprint arXiv:1711.05852*, 2017.

[20] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1910–1918.

[21] Q. Xie, E. Hovy, M.-T. Luong, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *CVPR*, 2020.

[22] Y. Xu, Y. Wang, H. Chen, K. Han, X. Chunjing, D. Tao, and C. Xu, "Positive-unlabeled compression on the cloud," in *Advances in Neural Information Processing Systems*, 2019, pp. 2561–2570.

[23] F. Sarfraz, E. Arani, and B. Zonooz, "Noisy collaboration in knowledge distillation," *openreview.net*, 2019.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] C. Yang, L. Xie, S. Qiao, and A. L. Yuille, "Training deep neural networks in generations: A more tolerant teacher educates better students," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5628–5635.

[26] L. Yu, V. O. Yazici, X. Liu, J. v. d. Weijer, Y. Cheng, and A. Ramisa, "Learning metrics from teachers: Compact networks for image embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2907–2916.

[27] S. Arora, M. M. Khapra, and H. G. Ramaswamy, "On knowledge distillation from complex networks for response prediction," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 3813–3822.

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

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

[30] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," *ICML*, 2018.

[31] D. Wang, Y. Li, Y. Lin, and Y. Zhuang, "Relational knowledge transfer for zero-shot learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[32] S.-I. Mirzadeh, M. Farajtabar, A. Li, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher," *the Association for the Advancement of Artificial Intelligence*, 2019.

[33] A. Malinin, B. Mlodozeniec, and M. Gales, "Ensemble distribution distillation," *arXiv preprint arXiv:1905.00076*, 2019.

[34] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[35] M. Phuong and C. H. Lampert, "Distillation-based training for multi-exit architectures," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1355–1364.

[36] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *International Conference on Learning Representations*, 2016.

[37] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1285–1294.

[38] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *ICLR*, 2017.

[39] B. B. Sau and V. N. Balasubramanian, "Deep model compression: Distilling knowledge from noisy teachers," *arXiv preprint arXiv:1610.09650*, 2016.

[40] X. Lan, X. Zhu, and S. Gong, "Knowledge distillation by on-the-fly native ensemble," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, pp. 7528–7538.

[41] G. Song and W. Chai, "Collaborative learning for deep neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 1832–1841.

[42] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He, "Data distillation: Towards omni-supervised learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4119–4128.

[43] X. Tan, Y. Ren, D. He, T. Qin, Z. Zhao, and T.-Y. Liu, "Multilingual neural machine translation with knowledge distillation," *ICLR*, 2019.

[44] J. Vongkulbhisal, P. Vinayavekhin, and M. Visentini-Scarzanella, "Unifying heterogeneous classifiers with distillation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[45] A. Wu, W.-S. Zheng, X. Guo, and J.-H. Lai, "Distilled person re-identification: Towards a more scalable system," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1187–1196.

[46] N. Dvornik, C. Schmid, and J. Mairal, "Diversity with cooperation: Ensemble methods for few-shot classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3723–3731.

[47] Z. Yang, L. Shou, M. Gong, W. Lin, and D. Jiang, "Model compression with two-stage multi-teacher knowledge distillation for web question answering system," *WSDM*, 2019.

[48] S. Park and N. Kwak, "Feed: Feature-level ensemble for knowledge distillation," *European Conference on Artificial Intelligence*, 2019.

[49] K. Lee, L. T. Nguyen, and B. Shim, "Stochasticity and skip connections improve knowledge transfer," 2019.

[50] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, "Online knowledge distillation with diverse peers," *Association for the Advancement of Artificial Intelligence*, 2020.

[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[52] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *Inter-

national Conference on Learning Representations, 2014.

[53] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1921–1930.

[54] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Advances in Neural Information Processing Systems*, 2018, pp. 2760–2769.

[55] Z. Huang and N. Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," *International Conference on Learning Representations*, 2017.

[56] Z. Zhang, G. Ning, and Z. He, "Knowledge projection for effective design of thinner and faster deep neural networks," 2017.

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

[58] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *arXiv preprint arXiv:2002.10957*, 2020.

[59] S. Srinivas and F. Fleuret, "Knowledge transfer with jacobian matching," *International Conference on Machine Learning*, 2018.

[60] X. Jin, B. Peng, Y. Wu, Y. Liu, J. Liu, D. Liang, J. Yan, and X. Hu, "Knowledge distillation via route constrained optimization," *ICCV*, 2019.

[61] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1365–1374.

[62] Z. Shen, Z. He, and X. Xue, "Meal: Multi-model ensemble via adversarial learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4886–4893.

[63] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," 2018.

[64] I. Chung, S. Park, J. Kim, and N. Kwak, "Feature-map-level online adversarial knowledge distillation," 2020.

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

[66] S. Changyong, L. Peng, X. Yuan, Q. Yanyun, D. Longquan, and M. Lizhuang, "Knowledge squeezed adversarial network compression," *arXiv preprint arXiv:1904.05100*, 2019.

[67] A. Kulkarni, N. Panchi, and S. Chiddarwar, "Stagewise knowledge distillation," *arXiv preprint arXiv:1911.06786*, 2019.

[68] G. Aguilar, Y. Ling, Y. Zhang, B. Yao, X. Fan, and E. Guo, "Knowledge distillation from internal representations," *AAAI*, 2019.

[69] M. Gao, Y. Shen, Q. Li, and C. C. Loy, "Residual knowledge distillation," *arXiv preprint arXiv:2002.09168*, 2020.

[70] H.-T. Li, S.-C. Lin, C.-Y. Chen, and C.-K. Chiang, "Layer-level knowledge distillation for deep neural network learning," *Applied Sciences*, vol. 9, no. 10, p. 1966, 2019.

[71] S. H. Lee, D. H. Kim, and B. C. Song, "Self-supervised knowledge distillation using singular value decomposition," in *European Conference on Computer Vision*. Springer, 2018, pp. 339–354.

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[73] N. Passalis and A. Tefas, "Learning deep representations

with probabilistic knowledge transfer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 268–284.

[74] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[75] S. Zagoruyko and N. Komodakis, "Wide residual networks," *BMVC*, 2016.

[76] D. Barber and F. V. Agakov, "The im algorithm: a variational approach to information maximization," in *Advances in neural information processing systems*, 2003, p. None.

[77] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*.   ACM, 2009, pp. 41–48.

[78] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[79] F. Horn and K.-R. Müller, "Learning similarity preserving representations with neural similarity and context encoders," 2016.

[80] K. Clark, M.-T. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, "Bam! born-again multi-task networks for natural language understanding," *ACL*, 2019.

[81] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[82] C. Zhang and Y. Peng, "Better and faster: knowledge transfer from multiple self-supervised learning tasks via graph distillation for video classification," *arXiv preprint arXiv:1804.10069*, 2018.

[83] C. Shen, M. Xue, X. Wang, J. Song, L. Sun, and M. Song, "Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3504–3513.

[84] S. Ruder, P. Ghaffari, and J. G. Breslin, "Knowledge adaptation: Teaching to adapt," *arXiv preprint arXiv:1702.02052*, 2017.

[85] X. Zhu, S. Gong *et al.*, "Knowledge distillation by on-the-fly native ensemble," in *Advances in neural information processing systems*, 2018, pp. 7517–7527.

[86] J. Vongkulbhisal, P. Vinayavekhin, and M. Visentini-Scarzanella, "Unifying heterogeneous classifiers with distillation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3175–3184.

[87] Z. Yang, L. Shou, M. Gong, W. Lin, and D. Jiang, "Model compression with two-stage multi-teacher knowledge distillation for web question answering system," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 690–698.

[88] L. Tran, B. S. Veeling, K. Roth, J. Swiatkowski, J. V. Dillon, J. Snoek, S. Mandt, T. Salimans, S. Nowozin, and R. Jenatton, "Hydra: Preserving ensemble diversity for model distillation," *arXiv preprint arXiv:2001.04694*, 2020.

[89] A. Ruiz and J. Verbeek, "Distilled hierarchical neural ensembles with adaptive inference cost," *arXiv preprint arXiv:2003.01474*, 2020.

[90] M.-C. Wu and C.-T. Chiu, "Multi-teacher knowledge distillation for compressed video action recognition based on deep learning," *Journal of Systems Architecture*, vol. 103, p. 101695, 2020.

[91] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers." in *Interspeech*, 2017, pp. 3697–3701.

[92] M. Mehak and V. N. Balasubramanian, "Knowledge dis-

tillation from multiple teachers using visual explanations," Ph.D. dissertation, Indian Institute of Technology Hyderabad, 2018.

[93] J.-w. Jung, H. Heo, H.-j. Shim, and H.-J. Yu, "Distilling the knowledge of specialist deep neural networks in acoustic scene classification," 2019.

[94] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for bert model compression," *EMNLP*, 2019.

[95] L. Liu, H. Wang, J. Lin, R. Socher, and C. Xiong, "Attentive student meets multi-task teacher: Improved knowledge distillation for pretrained models," *arXiv preprint arXiv:1911.03588*, 2019.

[96] X. Liu, P. He, W. Chen, and J. Gao, "Improving multi-task deep neural networks via knowledge distillation for natural language understanding," *arXiv preprint arXiv:1904.09482*, 2019.

[97] X. He, Z. Zhou, and L. Thiele, "Multi-task zipping via layer-wise neuron sharing," in *Advances in Neural Information Processing Systems*, 2018, pp. 6016–6026.

[98] C. Shen, X. Wang, J. Song, L. Sun, and M. Song, "Amalgamating knowledge towards comprehensive classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3068–3075.

[99] J. Ye, Y. Ji, X. Wang, K. Ou, D. Tao, and M. Song, "Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2829–2838.

[100] S. Luo, X. Wang, G. Fang, Y. Hu, D. Tao, and M. Song, "Knowledge amalgamation from heterogeneous networks by common feature learning," *IJCAI*, 2019.

[101] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, "Large scale distributed neural network training through online distillation," *ICLR*, 2018.

[102] P. Zhou, L. Mai, J. Zhang, N. Xu, Z. Wu, and L. S. Davis, "M2kd: Multi-model and multi-level knowledge distillation for incremental learning," *The British Machine Vision Conference (BMVC)*, 2019.

[103] L. Xiang and G. Ding, "Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification," *ECCV*, 2020.

[104] C. Gong, X. Chang, M. Fang, and J. Yang, "Teaching semi-supervised classifier via generalized distillation." in *IJCAI*, 2018, pp. 2156–2162.

[105] J. Ye, X. Wang, Y. Ji, K. Ou, and M. Song, "Amalgamating filtered knowledge: learning task-customized student from multi-task teachers," *IJCAI*, 2019.

[106] J. Gao, Z. Li, R. Nevatia *et al.*, "Knowledge concentration: Learning 100k object classifiers in a single cnn," *arXiv preprint arXiv:1711.07607*, 2017.

[107] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," *arXiv preprint arXiv:1511.06295*, 2015.

[108] K. Ahmed, M. H. Baig, and L. Torresani, "Network of experts for large-scale image categorization," in *European Conference on Computer Vision*.   Springer, 2016, pp. 516–532.

[109] I. M. Galván, P. Isasi, R. Aler, and J. M. Valls, "A selective learning method to improve the generalization of multi-layer feedforward neural networks," *International journal of neural systems*, vol. 11, no. 02, pp. 167–177, 2001.

[110] J. Kim, M. Hyun, I. Chung, and N. Kwak, "Feature fusion for online mutual knowledge distillation," *International Conference on Pattern Recognition*, 2019.

[111] S. Hou, X. Liu, and Z. Wang, "Dualnet: Learn complementary features for image recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 502–510.

[112] R. G. Lopes, S. Fenu, and T. Starner, "Data-free knowl-edge distillation for deep neural networks," *NIPS Workshop*, 2017.

[113] K. Bhardwaj, N. Suda, and R. Marculescu, "Dream distillation: A data-independent model compression framework," *ICML Joint Workshop*, 2019.

[114] M. Haroush, I. Hubara, E. Hoffer, and D. Soudry, "The knowledge within: Methods for data-free model compression," *CVPR*, 2020.

[115] G. K. Nayak, K. R. Mopuri, V. Shaj, R. V. Babu, and A. Chakraborty, "Zero-shot knowledge distillation in deep networks," *ICML*, 2019.

[116] H. Chen, Y. Wang, C. Xu, Z. Yang, C. Liu, B. Shi, C. Xu, C. Xu, and Q. Tian, "Data-free learning of student networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3514–3522.

[117] G. Fang, J. Song, C. Shen, X. Wang, D. Chen, and M. Song, "Data-free adversarial distillation," *arXiv preprint arXiv:1912.11006*, 2019.

[118] J. Ye, Y. Ji, X. Wang, X. Gao, and M. Song, "Data-free knowledge amalgamation via group-stack dual-gan," *CVPR*, 2020.

[119] J. Yoo, M. Cho, T. Kim, and U. Kang, "Knowledge extraction with no observable data," in *Advances in Neural Information Processing Systems*, 2019, pp. 2701–2710.

[120] H. Yin, P. Molchanov, Z. Li, J. M. Alvarez, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," *CVPR*, 2020.

[121] P. Micaelli and A. J. Storkey, "Zero-shot knowledge transfer via adversarial belief matching," in *Advances in Neural Information Processing Systems*, 2019, pp. 9547–9557.

[122] M. Kulkarni, K. Patil, and S. Karande, "Knowledge distillation using unlabeled mismatched images," *arXiv preprint arXiv:1703.07131*, 2017.

[123] L. Wang, W. Cho, and K.-J. Yoon, "Deceiving image-to-image translation networks for autonomous driving with adversarial perturbations," *IEEE Robotics and Automation Letters (RAL)*, 2020.

[124] L. Wang, Y.-S. Ho, K.-J. Yoon *et al.*, "Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 081–10 090.

[125] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," 2015.

[126] Q. Liu, L. Xie, H. Wang, and A. L. Yuille, "Semantic-aware knowledge preservation for zero-shot sketch-based image retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3662–3671.

[127] T. Li, J. Li, Z. Liu, and C. Zhang, "Few sample knowledge distillation for efficient network compression," *arXiv preprint arXiv:1812.01839*, 2018.

[128] A. Kimura, Z. Ghahramani, K. Takeuchi, T. Iwata, and N. Ueda, "Few-shot learning of neural networks from scratch by pseudo example optimization," *BMVC*, 2018.

[129] H. Bai, J. Wu, I. King, and M. Lyu, "Few shot network compression via cross distillation," *AAAI*, 2020.

[130] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in neural information processing systems*, 2006, pp. 1257–1264.

[131] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[132] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[133] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Pro-*

ceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

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

[135] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *ICLR Workshop*, 2018.

[136] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in neural information processing systems*, 2016, pp. 892–900.

[137] J.-C. Su and S. Maji, "Adapting models to signal degradation using distillation," *BMVC*, 2016.

[138] A. Nagrani, S. Albanie, and A. Zisserman, "Learnable pins: Cross-modal embeddings for person identity," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 71–88.

[139] ——, "Seeing voices and hearing faces: Cross-modal biometric matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8427–8436.

[140] J. Hoffman, S. Gupta, J. Leong, S. Guadarrama, and T. Darrell, "Cross-modal adaptation for rgb-d detection," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5032–5039.

[141] T. Afouras, J. S. Chung, and A. Zisserman, "Asr is all you need: cross-modal distillation for lip reading," *ICASSP*, 2020.

[142] S. Albanie, A. Nagrani, A. Vedaldi, and A. Zisserman, "Emotion recognition in speech using cross-modal transfer in the wild," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 292–301.

[143] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2827–2836.

[144] T. Salem, C. Greenwell, H. Blanton, and N. Jacobs, "Learning to map nearly anything," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 4803–4806.

[145] F. M. Thoker and J. Gall, "Cross-modal knowledge distillation for action recognition," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 6–10.

[146] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human pose estimation using radio signals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7356–7365.

[147] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient sound provides supervision for visual learning," in *European conference on computer vision*. Springer, 2016, pp. 801–816.

[148] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 609–617.

[149] T. Do, T.-T. Do, H. Tran, E. Tjiputra, and Q. D. Tran, "Compact trilinear interaction for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 392–401.

[150] Y. Aytar, C. Vondrick, and A. Torralba, "See, hear, and read: Deep aligned representations," *arXiv preprint arXiv:1706.00932*, 2017.

[151] C. Kim, H. V. Shin, T.-H. Oh, A. Kaspar, M. Elgharib, and W. Matusik, "On learning associations of faces and voices," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 276–292.

[152] Q. Dou, Q. Liu, P. A. Heng, and B. Glocker, "Unpaired multi-modal segmentation via knowledge distillation," *IEEE TMI*, 2020.

[153] F. Hafner, A. Bhuiyan, J. F. Kooij, and E. Granger, "A cross-modal distillation network for person re-identification in rgb-depth," *arXiv preprint arXiv:1810.11641*, 2018.

[154] C. Gan, H. Zhao, P. Chen, D. Cox, and A. Torralba, "Self-supervised moving vehicle tracking with stereo sound," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7053–7062.

[155] A. Perez, V. Sanguineti, P. Morerio, and V. Murino, "Audio-visual model distillation using acoustic images," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2854–2863.

[156] L. Zhao, X. Peng, Y. Chen, M. Kapadia, and D. N. Metaxas, "Knowledge as priors: Cross-modal knowledge generalization for datasets without superior knowledge," *CVPR*, 2020.

[157] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. W. Lau, "Dual student: Breaking the limits of the teacher in semi-supervised learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6728–6736.

[158] L. Wang, T.-K. Kim, and K.-J. Yoon, "Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[159] R. T. Mullapudi, S. Chen, K. Zhang, D. Ramanan, and K. Fatahalian, "Online model distillation for efficient video inference," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3573–3582.

[160] L. Gao, X. Lan, H. Mi, D. Feng, K. Xu, and Y. Peng, "Multistructure-based collaborative online distillation," *Entropy*, vol. 21, no. 4, p. 357, 2019.

[161] R. Lin, J. Xiao, and J. Fan, "Mod: A deep mixture model with online knowledge distillation for large scale video temporal concept localization," *ICCV YouTube8M workshop*, 2019.

[162] A. Cioppa, A. Deliege, M. Istasse, C. De Vleeschouwer, and M. Van Droogenbroeck, "Arthus: Adaptive real-time human segmentation in sports through online distillation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[163] H. Mobahi, M. Farajtabar, and P. L. Bartlett, "Self-distillation amplifies regularization in hilbert space," *arXiv preprint arXiv:2002.05715*, 2020.

[164] T.-B. Xu and C.-L. Liu, "Data-distortion guided self-distillation for deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5565–5572.

[165] S. Hahn and H. Choi, "Self-knowledge distillation in natural language processing," *arXiv preprint arXiv:1908.01851*, 2019.

[166] H. Lee, S. J. Hwang, and J. Shin, "Rethinking data augmentation: Self-supervision and self-distillation," *ICML*, 2020.

[167] E. J. Crowley, G. Gray, and A. J. Storkey, "Moonshine: Distilling with cheap convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 2888–2898.

[168] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3713–3722.

[169] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 831–839.

[170] Y. Luan, H. Zhao, Z. Yang, and Y. Dai, "Msd: Multi-self-distillation learning via multi-classifiers within deep neural networks," *arXiv preprint arXiv:1911.09418*, 2019.

[171] X. Lan, X. Zhu, and S. Gong, "Self-referenced deep learning," in *Asian conference on computer vision*. Springer,

2018, pp. 284–300.

[172] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, "Revisit knowledge distillation: a teacher-free framework," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[173] J. Wang, X. Wang, B. Jin, J. Yan, W. Zhang, and H. Zha, "Heterogeneous graph-based knowledge transfer for generalized zero-shot learning," 2019.

[174] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[175] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Advances in Neural Information Processing Systems*, 2017, pp. 742–751.

[176] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, "Learning monocular depth by distilling cross-domain stereo networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 484–500.

[177] X. Cheng, Z. Rao, Y. Chen, and Q. Zhang, "Explaining knowledge distillation by quantifying the knowledge," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 925–12 935.

[178] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *arXiv preprint arXiv:2006.07733*, 2020.

[179] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[180] L. Chongxuan, T. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *Advances in neural information processing systems*, 2017, pp. 4088–4098.

[181] X. Wang, R. Zhang, Y. Sun, and J. Qi, "Kdgan: knowledge distillation with generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 775–786.

[182] Z. Xu, Y.-C. Hsu, and J. Huang, "Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks," *ICLR workshop*, 2017.

[183] V. Belagiannis, A. Farshad, and F. Galasso, "Adversarial network compression," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[184] R. Liu, N. Fusi, and L. Mackey, "Teacher-student compression with generative adversarial networks," *arXiv preprint arXiv:1812.02271*, 2018.

[185] P. Liu, W. Liu, H. Ma, T. Mei, and M. Seok, "Ktan: knowledge transfer adversarial network," *International Joint Conference on Neural Networks (IJCNN)*, 2020.

[186] Y. Wang, C. Xu, C. Xu, and D. Tao, "Adversarial learning of portable student networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[187] S. Roheda, B. S. Riggan, H. Krim, and L. Dai, "Cross-modality distillation: A case for conditional generative adversarial networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2926–2930.

[188] Z. Xu, Y.-C. Hsu, and J. Huang, "Training student networks for acceleration with conditional adversarial networks." in *BMVC*, 2018, p. 61.

[189] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge distillation with adversarial samples supporting decision boundary," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3771–3778.

[190] J. Liu, Y. Chen, and K. Liu, "Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6754–6761.

[191] M. Goldblum, L. Fowl, S. Feizi, and T. Goldstein, "Adversarially robust distillation," *AAAI*, 2020.

[192] W. Hong and J. Yu, "Gan-knowledge distillation for one-stage object detection," *arXiv preprint arXiv:1906.08467*, 2019.

[193] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong gan: Continual learning for conditional image generation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2759–2768.

[194] Y. Wang, A. Gonzalez-Garcia, D. Berga, L. Herranz, F. S. Khan, and J. van de Weijer, "Minegan: effective knowledge transfer from gans to target domains with few images," *CVPR*, 2019.

[195] L. Gao, H. Mi, B. Zhu, D. Feng, Y. Li, and Y. Peng, "An adversarial feature distillation method for audio classification," *IEEE Access*, vol. 7, pp. 105 319–105 330, 2019.

[196] Z. Shen, Z. He, W. Cui, J. Yu, Y. Zheng, C. Zhu, and M. Savvides, "Adversarial-based knowledge distillation for multi-model ensemble and noisy data refinement," *AAAI*, 2019.

[197] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2604–2613.

[198] A. Aguinaldo, P.-Y. Chiang, A. Gain, A. Patil, K. Pearson, and S. Feizi, "Compressing gans using knowledge distillation," *arXiv preprint arXiv:1902.00159*, 2019.

[199] M. Li, J. Lin, Y. Ding, Z. Liu, J.-Y. Zhu, and S. Han, "Gan compression: Efficient architectures for interactive conditional gans," *CVPR*, 2020.

[200] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[201] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[202] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[203] H. Chen, Y. Wang, H. Shu, C. Wen, C. Xu, B. Shi, C. Xu, and C. Xu, "Distilling portable generative adversarial networks for image translation," *AAAI*, 2020.

[204] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Advances in neural information processing systems*, 2017, pp. 465–476.

[205] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2642–2651.

[206] Z. Xu, Y.-C. Hsu, and J. Huang, "Learning loss for knowledge distillation with conditional adversarial networks," *arXiv preprint arXiv:1709.00513*, 2017.

[207] S. Lee and B. C. Song, "Graph-based knowledge distillation by multi-head self-attention network," *BMVC*, 2019.

[208] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan, "Knowledge distillation via instance relationship graph," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7096–7104.

[209] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.

[210] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE*

Data Engineering Bulletin, 2017.

[211] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[212] C. Lassance, M. Bontonou, G. B. Hacene, V. Gripon, J. Tang, and A. Ortega, "Deep geometric knowledge distillation with graphs," *ICASSP*, 2020.

[213] S. Minami, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Knowledge transfer graph for deep collaborative learning," 2019.

[214] H. Yao, C. Zhang, Y. Wei, M. Jiang, S. Wang, J. Huang, N. V. Chawla, and Z. Li, "Graph few-shot learning via knowledge transfer," *arXiv preprint arXiv:1910.03053*, 2019.

[215] J. Ma and Q. Mei, "Graph representation learning via multi-task knowledge distillation," *NeurIPS GRL Workshop*, 2019.

[216] J. Liu, M. Shah, B. Kuipers, and S. Savarese, "Cross-view action recognition via view knowledge transfer," in *CVPR 2011*. IEEE, 2011, pp. 3209–3216.

[217] Z. Luo, J.-T. Hsieh, L. Jiang, J. Carlos Niebles, and L. Fei-Fei, "Graph distillation for action detection with privileged modalities," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 166–183.

[218] Y. Yang, J. Qiu, M. Song, D. Tao, and X. Wang, "Distilating knowledge from graph convolutional networks," *arXiv preprint arXiv:2003.10477*, 2020.

[219] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[220] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Advances in Neural Information Processing Systems*, 2018, pp. 4800–4810.

[221] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1024–1034.

[222] C. Li, X. Guo, and Q. Mei, "Deepgraph: Graph structure predicts network growth," *arXiv preprint arXiv:1610.06251*, 2016.

[223] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR*, 2017.

[224] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[225] M. A. Haidar and M. Rezagholizadeh, "Textkd-gan: text generation using knowledge distillation and generative adversarial networks," in *Canadian Conference on Artificial Intelligence*. Springer, 2019, pp. 107–118.

[226] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[227] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8896–8905.

[228] S. Zhang, J. Li, and B. Zhang, "Pairwise teacher-student network for semi-supervised hashing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[229] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," *arXiv preprint arXiv:1706.05208*, 2017.

[230] J. Choi, T. Kim, and C. Kim, "Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation," in *Proceedings of the IEEE Interna-*

tional Conference on Computer Vision, 2019, pp. 6830–6840.

[231] Q. Cai, Y. Pan, C.-W. Ngo, X. Tian, L. Duan, and T. Yao, "Exploring object relation in mean teacher for cross-domain detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 457–11 466.

[232] Y. Xu, B. Du, L. Zhang, Q. Zhang, G. Wang, and L. Zhang, "Self-ensembling attention networks: Addressing domain shift for semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5581–5588.

[233] Z. Hailat and X.-W. Chen, "Teacher/student deep semi-supervised learning for training with noisy labels," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 907–912.

[234] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, "Boosting self-supervised learning via knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9359–9367.

[235] G. Xu, Z. Liu, X. Li, and C. C. Loy, "Knowledge distillation meets self-supervision," *European Conf. Comput. Vis.(ECCV)*, 2020.

[236] S. Flennerhag, P. G. Moreno, N. D. Lawrence, and A. Damianou, "Transferring knowledge across learning processes," *arXiv preprint arXiv:1812.01054*, 2018.

[237] H. Jin, S. Zhang, X. Zhu, Y. Tang, Z. Lei, and S. Z. Li, "Learning lightweight face detector with knowledge distillation," in *2019 International Conference on Biometrics (ICB)*. IEEE, 2019, pp. 1–7.

[238] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 374–382.

[239] U. Michieli and P. Zanuttigh, "Knowledge distillation for incremental learning in semantic segmentation," *arXiv preprint arXiv:1911.03462*, 2019.

[240] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 233–248.

[241] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3400–3409.

[242] H. B. Ammar, E. Eaton, J. M. Luna, and P. Ruvolo, "Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[243] A. Ashok, N. Rhinehart, F. Beainy, and K. M. Kitani, "N2n learning: Network to network compression via policy gradient reinforcement learning," *arXiv preprint arXiv:1709.06030*, 2017.

[244] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.

[245] Z.-W. Hong, P. Nagarajan, and G. Maeda, "Periodic intra-ensemble knowledge distillation for reinforcement learning," *arXiv preprint arXiv:2002.00149*, 2020.

[246] Z. Xue, S. Luo, C. Wu, P. Zhou, K. Bian, and W. Du, "Transfer heterogeneous knowledge among peer-to-peer teammates: A model distillation approach," *arXiv preprint arXiv:2002.02202*, 2020.

[247] K. Lin, S. Wang, and J. Zhou, "Collaborative deep reinforcement learning," *arXiv preprint arXiv:1702.05796*, 2017.

[248] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[249] Y. Shan, "Distilling pixel-wise feature similarities for

semantic segmentation," *arXiv preprint arXiv:1910.14226*, 2019.

[250] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 578–587.

[251] Y. Chen, W. Li, and L. Van Gool, "Road: Reality oriented adaptation for semantic segmentation of urban scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7892–7901.

[252] J. Xie, B. Shuai, J.-F. Hu, J. Lin, and W.-S. Zheng, "Improving fast segmentation with teacher-student learning," *arXiv preprint arXiv:1810.08476*, 2018.

[253] Y. Hao, Y. Fu, Y.-G. Jiang, and Q. Tian, "An end-to-end architecture for class-incremental object detection with knowledge distillation," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 1–6.

[254] L. Chen, C. Yu, and L. Chen, "A new knowledge distillation for incremental object detection," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–7.

[255] R. Chen, H. Ai, C. Shang, L. Chen, and Z. Zhuang, "Learning lightweight pedestrian detector with hierarchical knowledge distillation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1645–1649.

[256] S. Tang, L. Feng, W. Shao, Z. Kuang, W. Zhang, and Y. Chen, "Learning efficient detector with semi-supervised adaptive distillation," *arXiv preprint arXiv:1901.00366*, 2019.

[257] M. R. U. Saputra, P. P. de Gusmao, Y. Almalioglu, A. Markham, and N. Trigoni, "Distilling knowledge from a deep pose regressor network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 263–272.

[258] X. Jin, C. Lan, W. Zeng, and Z. Chen, "Uncertainty-aware multi-shot knowledge distillation for image-based object re-identification," *arXiv preprint arXiv:2001.05197*, 2020.

[259] Y. Lee, N. Ahn, J. H. Heo, S. Y. Jo, and S.-J. Kang, "Teaching where to see: Knowledge distillation-based attentive information transfer in vehicle maker classification," *IEEE Access*, vol. 7, pp. 86 412–86 420, 2019.

[260] J. Xu, Y. Nie, P. Wang, and A. M. López, "Training a binary weight object detector by knowledge transfer for autonomous driving," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2379–2384.

[261] S. S. Kruthiventi, P. Sahay, and R. Biswal, "Low-light pedestrian detection from rgb images using multi-modal knowledge distillation," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 4207–4211.

[262] S. Ge, S. Zhao, C. Li, and J. Li, "Low-resolution face recognition in the wild via selective knowledge distillation," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 2051–2062, 2018.

[263] P. Luo, Z. Zhu, Z. Liu, X. Wang, and X. Tang, "Face model compression by distilling knowledge from neurons," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[264] Y. Feng, H. Wang, R. Hu, and D. T. Yi, "Triplet distillation for deep face recognition," *ICML ODML-CDNNR Workshop*, 2019.

[265] H. Felix, W. M. Rodrigues, D. Macêdo, F. Simões, A. L. Oliveira, V. Teichrieb, and C. Zanchettin, "Squeezed deep 6dof object detection using knowledge distillation," *IJCNN*, 2020.

[266] Y. Liu, X. Dong, W. Wang, and J. Shen, "Teacher-students knowledge distillation for siamese trackers," *arXiv preprint arXiv:1907.10586*, 2019.

[267] J. Karlekar, J. Feng, Z. S. Wong, and S. Pranata, "Deep face recognition model compression via knowledge transfer and distillation," *arXiv preprint arXiv:1906.00619*, 2019.

[268] X. Dong and Y. Yang, "Teacher supervises students how to learn from partially labeled images for facial landmark detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 783–792.

[269] A. Zhao, T. He, Y. Liang, H. Huang, G. V. d. Broeck, and S. Soatto, "Lates: Latent space distillation for teacher-student driving policy learning," *arXiv preprint arXiv:1912.02973*, 2019.

[270] F. Zhang, X. Zhu, and M. Ye, "Fast human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3517–3526.

[271] A. Martínez-González, M. Villamizar, O. Canévet, and J.-M. Odobez, "Efficient convolutional neural networks for depth-based multi-person pose estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[272] D.-H. Hwang, S. Kim, N. Monet, H. Koike, and S. Bae, "Lightweight 3d human pose estimation network training using teacher-student learning," *arXiv preprint arXiv:2001.05097*, 2020.

[273] X. Xu, Q. Zou, X. Lin, Y. Huang, and Y. Tian, "Integral knowledge distillation for multi-person pose estimation," *IEEE Signal Processing Letters*, 2020.

[274] C. Wang, C. Kong, and S. Lucey, "Distill knowledge from nrsfm for weakly supervised 3d pose learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 743–752.

[275] X. Nie, Y. Li, L. Luo, N. Zhang, and J. Feng, "Dynamic kernel distillation for efficient pose estimation in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6942–6950.

[276] S. Ao, X. Li, and C. X. Ling, "Fast generalized distillation for semi-supervised domain adaptation," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[277] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik, "Unifying distillation and privileged information," *International Conference on Learning Representations*, 2016.

[278] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," *CVPR*, 2018.

[279] Z. Meng, J. Li, Y. Gaur, and Y. Gong, "Domain adaptation via teacher-student learning for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 268–275.

[280] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Cr-doco: Pixel-level domain transfer with cross-domain consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1791–1800.

[281] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7472–7481.

[282] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9944–9953.

[283] J. Cho, D. Min, Y. Kim, and K. Sohn, "A large rgb-d dataset for semi-supervised monocular depth estimation," *IEEE Transactions on Image Processing*, 2019.

[284] A. Pilzer, S. Lathuiliere, N. Sebe, and E. Ricci, "Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9768–9777.

[285] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, "Learning monocular depth estimation infusing traditional stereo

knowledge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9799–9809.

[286] P. Liu, I. King, M. R. Lyu, and J. Xu, "Ddflow: Learning optical flow with unlabeled data distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8770–8777.

[287] F. Aleotti, M. Poggi, F. Tosi, and S. Mattoccia, "Learning end-to-end scene flow by distilling single tasks knowledge," *arXiv preprint arXiv:1911.10090*, 2019.

[288] F. Tosi, F. Aleotti, P. Z. Ramirez, M. Poggi, S. Salti, L. Di Stefano, and S. Mattoccia, "Distilled semantics for comprehensive scene understanding from videos," *CVPR*, 2020.

[289] H. Wang, Y. Li, Y. Wang, H. Hu, and M.-H. Yang, "Collaborative distillation for ultra-resolution universal style transfer," *arXiv*, pp. arXiv–2003, 2020.

[290] S. Bhardwaj, M. Srinivasan, and M. M. Khapra, "Efficient video classification using fewer frames," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 354–363.

[291] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, and W.-S. Zheng, "Progressive teacher-student learning for early action prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3556–3565.

[292] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei, "You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 923–932.

[293] C. Gan, B. Gong, K. Liu, H. Su, and L. J. Guibas, "Geometry guided convolutional neural networks for self-supervised video representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5589–5597.

[294] K. Fu, J. Li, Y. Song, Y. Zhang, S. Ge, and Y. Tian, "Ultrafast video attention prediction with coupled knowledge distillation," *arXiv preprint arXiv:1904.04449*, 2019.

[295] M. Farhadi and Y. Yang, "Tkd: Temporal knowledge distillation for active perception," *WACV*, 2020.

[296] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z. Zha, "Object relational graph with teacher-recommended learning for video captioning," *CVPR*, 2020.

[297] B. Pan, H. Cai, D.-A. Huang, K.-H. Lee, A. Gaidon, E. Adeli, and J. C. Niebles, "Spatio-temporal graph for video captioning with knowledge distillation," *CVPR*, 2020.

[298] Y. Lee, J. Jeong, J. Yun, W. Cho, and K.-J. Yoon, "Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9181–9189.

[299] P. Bashivan, M. Tensen, and J. J. DiCarlo, "Teacher guided architecture search," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5320–5329.

[300] M. Derezinski and M. K. Warmuth, "The limits of squared euclidean distance regularization," in *Advances in Neural Information Processing Systems*, 2014, pp. 2807–2815.

[301] S. W. Park and J. Kwon, "Sphere generative adversarial network based on geometric moment matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4292–4301.

[302] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Attention bridging network for knowledge transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5198–5207.

[303] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, "Attention gated networks: Learning to leverage salient regions in medical images," *Medical image analysis*, vol. 53, pp. 197–207, 2019.

[304] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[305] R.-W. Zhao, J. Li, Y. Chen, J.-M. Liu, Y.-G. Jiang, and X. Xue, "Regional gating neural networks for multi-label image classification." in *BMVC*, 2016, pp. 1–12.

[306] M. Mostafavi, L. Wang, and K.-J. Yoon, "Learning to reconstruct hdr images from events, with applications to depth and flow prediction," *International Journal of Computer Vision*, pp. 1–21.
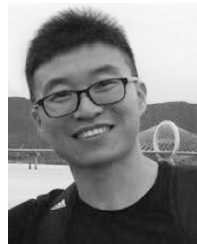
Lin Wang is a Ph.D. student in Visual Intelligence Lab., Dept. of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST). His research interests include deep learning (especially adversarial learning, transfer learning, semi-/self-supervised learning), event camera-based vision, low-level vision (image super-solution and deblurring, etc.) and computer vision for VR/AR.

Kuk-Jin Yoon received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology in 1998, 2000, and 2006, respectively. He is now an Associate Professor at the Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), South Korea, leading the Visual Intelligence Laboratory. Before joining KAIST, he was a Post-Doctoral Fellow in the PERCEPTION Team, INRIA, Grenoble, France, from 2006 to 2008, and was an Assistant/Associate Professor at the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, South Korea, from 2008 to 2018. His research interests include various topics in computer vision such as multi-view stereo, visual object tracking, SLAM and structure-from-motion, 360 camera and event-camera-base vision, sensor fusion.