Adversarial Distillation for Learning with Privileged Provisions

Xiaojie Wang, Rui Zhang*, Yu Sun, and Jianzhong Qi

Abstract—Knowledge distillation aims to train a student (model) for accurate inference in a resource-constrained environment. Traditionally, the student is trained by a high-capacity teacher (model) whose training is resource-intensive. The student trained this way is suboptimal because it is difficult to learn the real data distribution from the teacher. To address this issue, we propose to train the student against a discriminator in a minimax game. Such a minimax game has an issue that it can take an excessively long time for the training to converge. To address this issue, we propose adversarial distillation consisting of a student, a teacher, and a discriminator. The discriminator is now a multi-class classifier that distinguishes among the real data, the student, and the teacher. The student and the teacher aim to fool the discriminator via adversarial losses, while they learn from each other via distillation losses. By optimizing the adversarial and the distillation losses simultaneously, the student and the teacher can learn the real data distribution. To speed up the training, we propose to obtain low-variance gradient updates from the discriminator using a Gumbel-Softmax trick. We conduct extensive experiments to demonstrate the superiority of the proposed adversarial distillation under both accuracy and training speed.

Index Terms—Adversarial distillation, generative adversarial network, knowledge distillation, privileged information

1 Introduction

In many machine learning tasks, more resources are available at training (e.g., using labeled data to estimate a model's parameters) than at inference (e.g., fixing a model's parameters to predict unseen data) [18]. We refer to such extra resources available only at training as *privileged provisions*. Privileged provisions are not available at inference due to some requirement imposed by a specific task. An example task is to recommend tags for users to label their images, where extra textual features are privileged provisions [27], [39]. At training, textual features (i.e., titles and comments about images) are available in training data (see Fig. 1a). We can use these textual features to train more accurate tag recommendation models. However, such textual features cannot be used at inference. This is because this task requires a model to recommend tags even when users do not provide any textual features (see Fig. 1b). Another example task is to unlock mobile phones by face recognition, where intensive computational resources are privileged provisions [20], [41]. At training, we can use powerful servers with intensive computational resources (e.g., strong computation capability and large memory space) to train high-capacity face recognition models. Such computational resources are not available at inference. This is because this task requires a model to run on mobile phones with restricted computational resources, so that legit users can unlock their mobile phones without depending on remote services or internet connections. Here, a widely-recognized problem is called learning with privileged provisions, where the goal is to train an accurate model that satisfies stringent inference requirements [28].

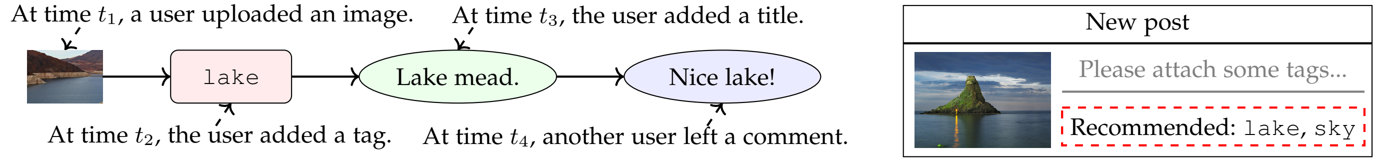
Knowledge distillation (KD), a typical approach to learning with privileged provisions, consists of a student and a teacher [7]. The student is meant for resource-constrained inference and hence cannot rely on privileged provisions. Compared with the student, the teacher makes use of privileged provisions by having a larger model capacity if intensive computational resources are available or by exploiting more features for learning if extra input features are available. KD first trains the teacher and then uses a different type of training, named *distillation*, to transfer the knowledge of the teacher into the student [9]. For example, Hinton et al. [18] treat the label distributions produced by the teacher as the "soft targets" and perform training by minimizing the Cross-Entropy measure between the soft targets and the label distributions produced by the student. Since the teacher often provides limited extra supervision signals on top of the real labels, it is difficult for the student to learn the real data distribution from the teacher [28].

To guarantee that the student can perfectly model the real data distribution in theory, we propose adapting generative adversarial network to learning with privileged provisions, where the student is trained against a discriminator in a minimax game. The student, serving as a generator, aims to generate fake labels that look like the real labels, whereas the discriminator aims to distinguish between the real and the fake labels. Such a naive adaptation provides theoretical guarantee, but has the issue of slow training speed: it usually requires a large number of training epochs for the training to converge with a good accuracy [14]. The training speed is slow because the gradients from the discriminator to update the student often vanish or explode during the training process [4]. Hence, it is challenging to theoretically guarantee the equilibrium, while empirically reducing the number of training instances and epochs required for training.

To tackle this challenge, we propose a novel training framework, named *adversarial distillation*, where a student and a teacher play a minimax game against a discriminator.

- *X. Wang, R. Zhang and J. Qi are with University of Melbourne. E-mail: xiaojiew1@student.unimelb.edu.au, rui.zhang@unimelb.edu.au, jianzhong.qi@unimelb.edu.au. *R. Zhang is the corresponding author.*
- *Y. Sun is with Twitter Inc. E-mail: ysun@twitter.com.*

Manuscript received May 07, 2019; revised October 04, 2019.


At time t_1, a user uploaded an image.
At time t_3, the user added a title.
lake
Lake mead.
Nice lake!
At time t_2, the user added a tag.
At time t_4, another user left a comment.
New post
Please attach some tags...
Recommended: lake, sky

(a) At training, we use historical data for training where extra texts, e.g., comments and titles (besides labeled tags) were accumulated over time $t_1 < t_2 < t_3 < t_4$.

(b) At inference, we aim to recommend `lake` and `sky` right after an image is uploaded.

Fig. 1: When we recommend tags, extra texts about images are available at training (a) but not available at inference (b).

We propose two variants of adversarial distillation: one uses a binary-class discriminator for less costly training and the other uses a multi-class discriminator for more accurate inference. We start with the one using a binary-class discriminator, called *adversarial distillation with a binary-class discriminator* (ADIB). The student and the teacher compete against the discriminator via adversarial losses: the student and the teacher aim to generate fake labels that resemble the real labels, whereas the discriminator is a binary-class classifier that aims to distinguish between the real and the fake labels. Meanwhile, the student and the teacher learn from each other via distillation losses to reach an agreement on generating what fake labels. By optimizing both the adversarial and the distillation losses, the student and the teacher can learn the real data distribution with less training epochs and instances. However, a limitation of ADIB is that the adversarial losses drive a mixture distribution of the student and the teacher, rather than each of the student and the teacher, towards the real data distribution. The adversarial losses may even nudge the teacher away from the real data distribution if the student is still far from the real data distribution, which leads to less accurate inference. To overcome this limitation, we further propose *adversarial distillation with a multi-class discriminator* (ADIM), where the discriminator is a multi-class classifier that distinguishes whether a label is generated by the real data, the student, or the teacher. The multi-class discriminator has more parameters and operations than the binary-class discriminator does, which results in more costly training. The adversarial losses associated with the multi-class discriminator drive each of the student and the teacher towards the real data distribution. The adversarial losses are general in the sense that the distillation losses for training the student and the teacher are special cases of the adversarial losses under certain conditions.

We further consider reducing the variance of the gradients from the discriminator to accelerate the training of adversarial distillation. The gradients from the discriminator may have high variance when obtained through the widely used policy gradient methods [40], [45]. It is non-trivial to obtain low-variance gradients from the discriminator because the student and the teacher generate discrete samples, which are not differentiable. We propose to relax the discrete distributions learned by the student and the teacher into concrete distributions by applying the Gumbel-Softmax trick [22], [29]. We use the concrete distributions for generating continuous samples to enable end-to-end differentiability and sufficient control over the variance of gradients. Given the continuous samples, we obtain low-variance gradients from the discriminator to accelerate the training.

The main contributions of this paper are listed as follows.

C1: To our knowledge, we are the first to adapt generative adversarial network to addressing the issue of knowledge distillation in learning with privileged provisions.

C2: We propose a training framework, named ADIB, for using privileged provisions available only at training to learn a student suitable for resource-constrained inference.

C3: We propose a training framework, named ADIM, to overcome the limitation that the binary-class discriminator of ADIB cannot guarantee the equilibrium (Section 4.2).

C4: We propose a Gumbel-Softmax trick to train ADIB, which yields low-variance gradient updates and reduces the training epochs required for a good convergence accuracy.

C5: We also apply the proposed Gumbel-Softmax trick to speeding up the training of ADIM by deriving the formulas for computing low-variance gradients (Section 4.3).

C6: We prove that the adversarial losses of ADIM can approximate the distillation losses, which shows the generality of the adversarial losses for distillation (Section 4.4).

C7: We empirically investigate the superiority of ADIM, the benefits of the Gumbel-Softmax trick, the effects of the teacher learning from the student, and the gradient variance of the adversarial and the distillation losses (Section 5).

We have presented the contributions **C1**, **C2** and **C4** in our previous conference paper [42]. In this journal extension, we make new contributions **C3** and **C5** to **C7**. The main challenges of the journal extension include: (1) Imposing several distributions (i.e., the distributions underlying the real data, the student, and the teacher) to be identical with a single discriminator, which is tackled by **C3**. (2) Obtaining low-variance gradients of the adversarial losses associated with the multi-class discriminator to update the student and the teacher, which is tackled by **C5**. The rest of the paper is organized as follows. Section 2 provides a review of the related work. Section 3 introduces the problem of learning with privileged provisions. Section 4 proposes the adversarial distillation and describes the theoretical results in detail. Section 5 presents experiments in two real-world tasks. Section 6 concludes the paper.

2 Related Work

Our work is closely related to existing work on knowledge distillation (KD) and generative adversarial network (GAN).

The goal of KD is to train a lightweight student that satisfies the requirements of low memory use and fast running time at inference [9]. Early studies on KD adopt two-phase training, where training a high-capacity teacher is followed by training a student to match soft targets or feature representations produced by the teacher. For example, Ba and Caruana [7] train a shallow student network to mimic a pretrained teacher network by matching logits via a L2 loss.

Hinton et al. [18] generalize this work by training a student to predict soft targets produced by a teacher. To simplify the complex procedure of the two-phase training, recent studies on KD develop one-phase training. For example, Zhang et al. [49] simultaneously train a student and a teacher to match soft targets produced by each other [50]. Lopez-Paz et al. unify KD and learning using privileged information (LUPI) as generalized distillation where a teacher can be trained by taking privileged information as input [28]. Compared with KD, we introduce a discriminator to guarantee that the student can learn the real data distribution in theory.

Previous studies mostly formulate GAN as a two-player framework with a generator and a discriminator. Initially, GAN is proposed to generate continuous data by training a generator and a discriminator adversarially in a minimax game [16]. Since discrete data makes it difficult to obtain gradients from a discriminator, GAN has only recently been introduced to generate discrete data [47], [48]. For example, sequence GAN (SeqGAN) [45] models token sequence generation as a stochastic policy and updates a generator by Monte Carlo search. Recently, several multi-player frameworks have been proposed to enhance the learning capacity of the two-player frameworks [15]. For example, Pu et al. propose JointGAN which uses multiple generators and a discriminator to learn a joint distribution of multiple random variables [32]. Our framework also has multiple players including two generators and a discriminator, but differs from existing work [12] in that both generators learn a conditional distribution over labels given features and hence can learn from each other to improve their accuracy through KD. There is also a rich body of studies on improving the training of GAN by, e.g., feature matching [34]. These studies focus on generating continuous data and avoiding the problem of mode collapse [5], whereas we aim at generating discrete data and reducing the number of training epochs.

We explore the idea of retaining advantages and avoiding disadvantages of KD and GAN in a single framework. Similar ideas have been explored in recent studies. For example, Xu et al. introduce a discriminator to distinguish logits produced by the student and the teacher [44], whereas Chen et al. introduce a discriminator to distinguish shared embeddings in the student and the teacher [11]. Our work differs from these studies mainly in that we introduce a discriminator to distinguish among the real data, the student, and the teacher. This way, the optimal student and the optimal teacher are guaranteed to fit the real data distribution perfectly.

3 Preliminaries

We focus on multi-label classification [25], [43], [46], although the same ideas can be applied to other problems with a discrete output space, e.g., webpage ranking [40]. We study the problem of learning with privileged provisions: it makes use of privileged provisions ϱ, available only at training and not available at inference, to learn a student S (a multi-label classifier) that satisfies stringent inference requirements [28]. The inference requirements can be running in real time with restricted computational resources where privileged provisions are intensive computational resources [18], or lacking a certain type of input features where privileged provisions are extra input features, a.k.a, privileged information [39]. Let $\boldsymbol{x}$ be a random vector in a feature (input) space $\mathcal{X}$ and y be a random variable in a label (output) space $\mathcal{Y}$. Let $p_r(y|\boldsymbol{x})$ be the real data distribution from which a real label y is sampled given a feature vector $\boldsymbol{x}$. Let $p_s(y|\boldsymbol{x})$ parameterized by θ_s be the categorical distribution defined by the student. The goal is to learn the optimal student S^* that fits the real data distribution perfectly $p_s(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x})$ and does not require privileged provisions to perform inference.

A typical approach to learning with privileged provisions is knowledge distillation (KD), which consists of a student S and a teacher T. The teacher is a multi-label classifier that makes use of privileged provisions by having a larger model capacity or by taking more features as input. Let $p_t(y|\boldsymbol{x})$ parameterized by θ_t be the categorical distribution defined by the teacher. Here, with a slight abuse of notation, we also use $\boldsymbol{x}$ to denote a feature vector including privileged information, which should be clear from context. First, the teacher is typically trained by minimizing the Kullback-Leibler divergence, which is equivalent to minimizing the Cross-Entropy measure between the distributions of the real data and the teacher

$$\min_{\theta_t} \mathcal{L}_{\mathrm{KL}}(p_r \,\|\, p_t) = \sum_{y\in\mathcal{Y}} p_r(y|\boldsymbol{x}) \log \frac{p_r(y|\boldsymbol{x})}{p_t(y|\boldsymbol{x})} . \tag{1}$$

Once the teacher has been trained, the student is typically trained by learning from both the teacher and the real data, which are balanced by a hyper-parameter $\upsilon > 0$

$$\min_{\theta_s} \mathcal{L}_{\mathrm{KD}}(p_t \,\|\, p_s) = \mathcal{L}_{\mathrm{KL}}(p_t \,\|\, p_s) + \upsilon \mathcal{L}_{\mathrm{KL}}(p_r \,\|\, p_s) . \tag{2}$$

We refer to such losses used to characterize a distillation training process, e.g., in Eqn. (2) as the *distillation losses*.

An alternative approach naively adapts generative adversarial network (NGAN) to learning with privileged provisions, where a student S and a discriminator D play a minimax game. The discriminator is a binary-class classifier and also makes use of privileged provisions in the same way as the teacher. Given a feature vector $\boldsymbol{x}$, the discriminator parameterized by θ_d computes the probability $D(\boldsymbol{x}, y)$ of a label y being real. The student aims to generate fake labels that look like the real labels by sampling from its categorical distribution, whereas the discriminator aims to draw a clear distinction between the real and the fake labels. We define the value function for the minimax game of NGAN as

$$\begin{aligned} \min_{\theta_s} \max_{\theta_d} \mathcal{V}_{\mathrm{NGAN}} = \mathbb{E}_{y\sim p_r}&[\log D(\boldsymbol{x}, y)] \\ &+ \mathbb{E}_{y\sim p_s}[\log(1 - D(\boldsymbol{x}, y))] . \end{aligned} \tag{3}$$

We refer to such losses used to characterize an adversarial training process, e.g., in Eqn. (3) as *adversarial losses*. The adversarial losses differ from the distillation losses in that the former are related to a certain discriminator but the latter are not. The minimax game of NGAN has an equilibrium, where the student perfectly fits the real data distribution while the discriminator does no better than random guesses at deciding whether a label is real or fake [6]. Adversarial training often proceeds by updating the student and the discriminator alternately until convergence [40].

Our key observation is that the advantages and the disadvantages of KD and NGAN are complementary: (1) KD usually requires a small number of training instances and epochs for a good convergence accuracy, but does not

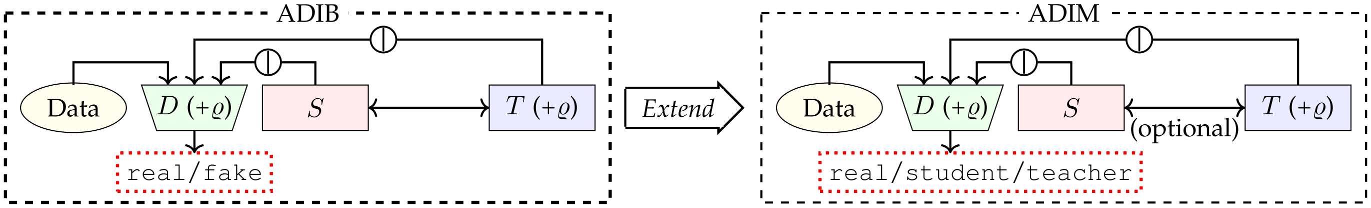


Fig. 2: Difference between ADIB and ADIM. The student S does not use privileged provisions, whereas the teacher T and the discriminator D do $(+\varrho)$. Adversarial and distillation losses are denoted by lines with single and double arrows, respectively. The equilibrium remains the same if the optional line is removed. Sampling with the GS trick is denoted by $\oplus$.

guarantee the student to learn the real data distribution [24]. (2) NGAN guarantees the student to learn the real data distribution, but normally requires a large number of training instances and epochs which may be difficult to satisfy in practice [4]. We aim to retain the advantages and avoid the disadvantages of the two approaches in a single framework.

4 Proposed Frameworks

To speed up the training while preserving the equilibrium, we first propose *adversarial distillation with a binary-class discriminator* (ADIB), a three-player framework with a student, a teacher, and a binary-class discriminator [42]. The student and the teacher are trained against the discriminator via adversarial losses, which is regularized by distillation losses between the student and the teacher. The distillation-regularized adversarial training can theoretically guarantee the equilibrium where both the student and the teacher perfectly fit the real data distribution. A limitation of ADIB is that by optimizing the adversarial losses, the binary-class discriminator pushes a mixture distribution of the student and the teacher towards the real data distribution. This is not ideal because as the equilibrium suggests, we aim to push each of the student and the teacher (rather than their mixture) towards the real data distribution. To overcome this limitation, we further propose *adversarial distillation with a multi-class discriminator* (ADIM). Our key idea is to design a multi-class discriminator that can distinguish not only whether a label is real or fake, but also whether a fake label comes from the student or the teacher. Following this idea, we formulate the adversarial losses by training a multi-class discriminator to distinguish among the real data, the student, and the teacher. Moreover, we provide a rigorous theoretical interpretation that builds up the connection between the adversarial and the distillation losses of ADIM.

4.1 Binary-class Adversarial Distillation (ADIB)

We formulate ADIB as a minimax game with a student S, a teacher T, and a binary-class discriminator D, which is illustrated by the left half of Fig. 2. Given a feature vector $\boldsymbol{x}$, the student and the teacher generate fake labels by sampling from the categorical distributions $p_s(y|\boldsymbol{x})$ and $p_t(y|\boldsymbol{x})$, while the discriminator is a binary-class classifier that computes the probability $D(\boldsymbol{x}, y)$ of a label y being real. The discriminator aims to maximize the probability of correctly distinguishing the real and the fake labels, whereas the student and the teacher aim to minimize the probability that the discriminator rejects their fake labels. Meanwhile, the student and the teacher learn from each other by mimicking the categorical distributions learned by each other. Such mutual learning helps the student and the teacher avoid generating different fake labels to fool the discriminator. Formally, we define the value function for the minimax game of ADIB as

$$\begin{aligned}\min_{\theta_s,\theta_t}\max_{\theta_d} \mathcal{V}_{\text{ADIB}} &= \mathbb{E}_{y\sim p_r}[\log D(\boldsymbol{x}, y)] \\ &+ \omega_s \mathbb{E}_{y\sim p_s}[\log(1 - D(\boldsymbol{x}, y))] + \nu \mathcal{L}_{\text{KD}}(p_t \parallel p_s) \\ &+ \omega_t \mathbb{E}_{y\sim p_t}[\log(1 - D(\boldsymbol{x}, y))] + \mu \mathcal{L}_{\text{KD}}(p_s \parallel p_t)\,,\end{aligned} \tag{4}$$

where hyper-parameters $\omega_s, \omega_t, \nu, \mu > 0$ and $\omega_s + \omega_t = 1$. On the right hand side of Eqn. (4), we denote by $\mathcal{L}_{\text{BA}}$ the expectation terms (the adversarial losses) and by $\mathcal{L}_{\text{BD}}$ the other two terms (the distillation losses). $\mathcal{L}_{\text{KD}}(p_t \parallel p_s)$ and $\mathcal{L}_{\text{KD}}(p_s \parallel p_t)$ are the distillation losses for training the student and the teacher, respectively. The distillation losses can be defined in several ways, e.g., the L2 loss between logits [7].

We show that the student and the teacher perfectly fit the real data distribution at the equilibrium of ADIB. To see this, we first derive the optimal discriminator D^* as follows.

Lemma 4.1. *For any fixed student and teacher, the optimal discriminator that maximizes the value function of ADIB is*

$$D^*(\boldsymbol{x}, y) = \frac{p_r(y|\boldsymbol{x})}{p_r(y|\boldsymbol{x}) + p_\omega(y|\boldsymbol{x})}\,, \tag{5}$$

where $p_\omega(y|\boldsymbol{x}) = \omega_s p_s(y|\boldsymbol{x}) + \omega_t p_t(y|\boldsymbol{x})$ is a mixture distribution of the student and the teacher.

Proof. Since the distillation losses $\mathcal{L}_{\text{BD}}$ do not contain the parameters of the discriminator, we can maximize the adversarial losses $\mathcal{L}_{\text{BA}}$ to obtain the optimal discriminator

$$\begin{aligned}&\mathcal{V}^*_{\text{ADIB}} = \max_{\theta_d} \mathcal{L}_{\text{BA}}\,, \\ &= \sum_{y\in\mathcal{Y}} p_r(y|\boldsymbol{x}) \log D(\boldsymbol{x}, y) + \omega_s \sum_{y\in\mathcal{Y}} p_s(y|\boldsymbol{x}) \log(1 - D(\boldsymbol{x}, y)) \\ &\quad + \omega_t \sum_{y\in\mathcal{Y}} p_t(y|\boldsymbol{x}) \log(1 - D(\boldsymbol{x}, y))\,, \\ &= \sum_{y\in\mathcal{Y}} p_r(y|\boldsymbol{x}) \log D(\boldsymbol{x}, y) + \sum_{y\in\mathcal{Y}} p_\omega(y|\boldsymbol{x}) \log(1 - D(\boldsymbol{x}, y))\,.\end{aligned}$$

Hence, the optimal discriminator is given by Eqn. (5) because a function $h(z) = a \log z + b \log(1 - z)$ $(0 < z < 1)$ achieves the maximum at $z = \frac{a}{a+b}$. This completes the proof. □

Given the optimal discriminator, we show that both the student and the teacher fit the real data distribution perfectly when ADIB achieves its equilibrium in the minimax game.

Theorem 4.2. *In the minimax game of ADIB, the equilibrium is achieved if and only if $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x})$. At that point, the value function is equal to $-\log(4)$.*

Proof. Given the optimal discriminator in Lemma 4.1, the minimax game of ADIB can be reformulated as

$$\begin{aligned}
&\min_{\theta_s,\theta_t}\left\{\mathcal{V}^*_{\mathrm{ADIB}} + \nu\mathcal{L}_{\mathrm{KD}}(p_t\,\|\,p_s) + \mu\mathcal{L}_{\mathrm{KD}}(p_s\,\|\,p_t)\right\}\\
&= \sum_{y\in\mathcal{Y}} p_r(y|\boldsymbol{x}) \log \frac{p_r(y|\boldsymbol{x})}{p_r(y|\boldsymbol{x})+p_\omega(y|\boldsymbol{x})} + \nu\mathcal{L}_{\mathrm{KD}}(p_t\,\|\,p_s)\\
&\quad + \sum_{y\in\mathcal{Y}} p_\omega(y|\boldsymbol{x}) \log \frac{p_\omega(y|\boldsymbol{x})}{p_r(y|\boldsymbol{x})+p_\omega(y|\boldsymbol{x})} + \mu\mathcal{L}_{\mathrm{KD}}(p_s\,\|\,p_t)\,,\\
&= 2\mathcal{L}_{\mathrm{JS}}(p_r\,\|\,p_\omega) - \log(4) + \nu\mathcal{L}_{\mathrm{KD}}(p_t\,\|\,p_s) + \mu\mathcal{L}_{\mathrm{KD}}(p_s\,\|\,p_t)\,,
\end{aligned}$$

where the Jensen-Shannon divergence is given by

$$2\mathcal{L}_{\mathrm{JS}}(p_r\,\|\,p_\omega) = \mathcal{L}_{\mathrm{KL}}\left(p_r\,\middle\|\,\frac{p_r+p_\omega}{2}\right) + \mathcal{L}_{\mathrm{KL}}\left(p_\omega\,\middle\|\,\frac{p_r+p_\omega}{2}\right).$$

Note that the Jensen-Shannon divergence achieves zero (the minimum) if and only if $p_\omega(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x})$ while the distillation losses achieve zero (the minimum) if and only if $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$ [12]. Therefore, the equilibrium is reached if and only if $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x})$, where the value function is equal to $-\log(4)$, which completes the proof. □

4.2 Multi-class Adversarial Distillation (ADIM)

A limitation of ADIB is that the adversarial losses alone, i.e., setting $\nu = \mu = 0$ in Eqn. (6), cannot guarantee the student and the teacher to learn the real data distribution. This is because when $\nu = \mu = 0$, the equilibrium is reached if and only if $p_\omega(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x})$ (see Theorem 4.2), which is not equivalent to $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x})$. This limitation results from the fact that the binary-class discriminator of ADIB does not distinguish whether a fake label comes from the student or the teacher. Therefore, the binary-class discriminator cannot force the student and the teacher to move towards each other. We overcome this limitation by designing the discriminator as a multi-class classifier.

Specifically, we formulate ADIM as a minimax game with a student S, a teacher T, and a multi-class discriminator D, which is illustrated by the right half of Fig. 2. The discriminator is a multi-class classifier that computes the probability of label y coming from the real data $D_r(\boldsymbol{x}, y)$, the student $D_s(\boldsymbol{x}, y)$, or the teacher $D_t(\boldsymbol{x}, y)$. To simplify notation, let $\mathcal{I} = \{r, s, t\}$ be the set of subscripts corresponding to the real data, the student, and the teacher. We define the value function for the minimax game of ADIM as

$$\begin{aligned}
\min_{\theta_s,\theta_t}\max_{\theta_d}\mathcal{V}_{\mathrm{ADIM}} &= \sum_{i\in\mathcal{I}}\omega_i\mathbb{E}_{y\sim p_i}[\log D_i(\boldsymbol{x},y)] \qquad (6)\\
&+ \nu\mathcal{L}_{\mathrm{KD}}(p_t\,\|\,p_s) + \mu\mathcal{L}_{\mathrm{KD}}(p_s\,\|\,p_t)\,,\ \text{s.t.},\ \sum_{i\in\mathcal{I}} D_i(\boldsymbol{x},y) = 1\,,
\end{aligned}$$

where hyper-parameters $\omega_r, \omega_s, \omega_t > 0$ satisfy $\omega_r + \omega_s + \omega_t = 1$ and $\nu, \mu \geq 0$. On the right hand side of Eqn. (6), we denote by $\mathcal{L}_{\mathrm{MA}}$ the expectation terms (the adversarial losses) and by $\mathcal{L}_{\mathrm{MD}}$ the other two terms (the distillation losses).

We show that the student and the teacher perfectly fit the real data distribution at the equilibrium of ADIM by first deriving the optimal discriminator D^* as follows.

Lemma 4.3. *For any fixed student and teacher, the optimal discriminator that maximizes the value function of ADIM is*

$$D^*_i(\boldsymbol{x},y) = \frac{\omega_i p_i(y|\boldsymbol{x})}{\bar{p}_a(y|\boldsymbol{x})}\,,\ i\in\mathcal{I}\,, \qquad (7)$$

where $\bar{p}_a(y|\boldsymbol{x}) = \omega_r p_r(y|\boldsymbol{x}) + \omega_s p_s(y|\boldsymbol{x}) + \omega_t p_t(y|\boldsymbol{x})$.

Proof. Since the distillation losses $\mathcal{L}_{\mathrm{MD}}$ do not contain the parameters of the discriminator, we can maximize the adversarial losses $\mathcal{L}_{\mathrm{MA}}$ to obtain the optimal discriminator

$$\mathcal{V}^*_{\mathrm{ADIM}} = \max_{\theta_d}\mathcal{L}_{\mathrm{MA}} = \sum_{i\in\mathcal{I}}\omega_i\sum_{y\in\mathcal{Y}} p_i(y|\boldsymbol{x})\log D_i(\boldsymbol{x},y)\,.$$

After introducing a Lagrange multiplier λ_y for each $y \in \mathcal{Y}$, we compute a Lagrange function as

$$\begin{aligned}
\mathcal{L}_{\mathrm{ADIM}} &= \mathcal{L}_{\mathrm{MA}} + \sum_{y\in\mathcal{Y}}\lambda_y\left(\sum_{i\in\mathcal{I}} D_i(\boldsymbol{x},y) - 1\right),\\
&= \sum_{y\in\mathcal{Y}}\sum_{i\in\mathcal{I}}\omega_i p_i(y|\boldsymbol{x})\log D_i(\boldsymbol{x},y) + \sum_{y\in\mathcal{Y}}\lambda_y\left(\sum_{i\in\mathcal{I}} D_i(\boldsymbol{x},y) - 1\right),\\
&= \sum_{y\in\mathcal{Y}}\left(\sum_{i\in\mathcal{I}}\omega_i p_i(y|\boldsymbol{x})\log D_i(\boldsymbol{x},y) + \lambda_y\left(\sum_{i\in\mathcal{I}} D_i(\boldsymbol{x},y) - 1\right)\right).
\end{aligned}$$

We set gradients of the Lagrange function w.r.t. the discriminator's distribution and the Lagrange multipliers to zero

$$\begin{aligned}
\frac{\partial\mathcal{L}_{\mathrm{ADIM}}}{\partial D_i(\boldsymbol{x},y)} &= \frac{\omega_i p_i(y|\boldsymbol{x})}{D_i(\boldsymbol{x},y)} - \lambda_y = 0\,,\ y\in\mathcal{Y},\ i\in\mathcal{I},\\
\frac{\partial\mathcal{L}_{\mathrm{ADIM}}}{\partial\lambda_y} &= \sum_{i\in\mathcal{I}} D_i(\boldsymbol{x},y) - 1 = 0\,,\ y\in\mathcal{Y}\,.
\end{aligned}$$

Solving these equations yields the optimal discriminator given by Eqn. (7), which completes the proof. □

Given the optimal discriminator, we show that both the student and the teacher fit the real data distribution perfectly when ADIM achieves its equilibrium in the minimax game.

Theorem 4.4. *In the minimax game of ADIM, the equilibrium is achieved if and only if* $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x})$. *At that point, the value function is equal to* $\log(\omega_r^{\omega_r}\omega_s^{\omega_s}\omega_t^{\omega_t})$.

Proof. Given the optimal discriminator in Lemma 4.3, the minimax game of ADIM can be reformulated as

$$\begin{aligned}
&\min_{\theta_s,\theta_t}\left\{\mathcal{V}^*_{\mathrm{ADIM}} + \nu\mathcal{L}_{\mathrm{KD}}(p_t\,\|\,p_s) + \mu\mathcal{L}_{\mathrm{KD}}(p_s\,\|\,p_t)\right\}\\
&= \sum_{i\in\mathcal{I}}\omega_i\sum_{y\in\mathcal{Y}} p_i(y|\boldsymbol{x})\log\frac{\omega_i p_i(y|\boldsymbol{x})}{\bar{p}_a(y|\boldsymbol{x})} + \nu\mathcal{L}_{\mathrm{KD}}(p_t\,\|\,p_s)\\
&\quad + \mu\mathcal{L}_{\mathrm{KD}}(p_s\,\|\,p_t)\,,\\
&= \sum_{i\in\mathcal{I}}\omega_i\sum_{y\in\mathcal{Y}} p_i(y|\boldsymbol{x})\left(\log\omega_i + \log\frac{p_i(y|\boldsymbol{x})}{\bar{p}_a(y|\boldsymbol{x})}\right) + \nu\mathcal{L}_{\mathrm{KD}}(p_t\,\|\,p_s)\\
&\quad + \mu\mathcal{L}_{\mathrm{KD}}(p_s\,\|\,p_t)\,,\\
&= \log(\omega_r^{\omega_r}\omega_s^{\omega_s}\omega_t^{\omega_t}) + \sum_{i\in\mathcal{I}}\omega_i\mathcal{L}_{\mathrm{KL}}(p_i\,\|\,\bar{p}_a) + \nu\mathcal{L}_{\mathrm{KD}}(p_t\,\|\,p_s)\\
&\quad + \mu\mathcal{L}_{\mathrm{KD}}(p_s\,\|\,p_t)\,.
\end{aligned}$$

The second term, i.e., the summation over three Kullback-Leibler divergences, achieves zero (the minimum) if and only if $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x}) = \bar{p}_a(y|\boldsymbol{x})$ [32]. Therefore, even when the distillation losses are not used ($\nu = \mu = 0$), the equilibrium is still achieved if and only if the student and the teacher perfectly fit the real data distribution. The distillation losses are usually defined as the Kullback-Leibler divergences between the real data, the student, and the teacher. These Kullback-Leibler divergences achieve zero (the minimum) if and only if $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x}) = p_r(y|\boldsymbol{x})$. Hence, adding the distillation losses does not change the equilibrium, at which point the value function is equal to $\log(\omega_r^{\omega_r}\omega_s^{\omega_s}\omega_t^{\omega_t})$. This completes the proof. □

To achieve the equilibrium, we should be able to sample sufficient training instances from the real data distribution, but in practice we often have a finite set of training instances. Besides, the equilibrium requires the student and the teacher to have enough capability to represent the real data distribution, but these models are often implemented by a certain family of distributions. Our empirical results suggest that while the theoretical guarantee may not hold in practice, we can achieve a reasonable approximation to the equilibrium by training limited-capacity models on a finite training set.

4.3 Training Acceleration via the Gumbel-Softmax Trick

Next, we discuss how to accelerate the training speed of the proposed frameworks in terms of reducing the number of training epochs required to converge. The training speed is closely related to the variance of gradients: high-variance gradients usually make the training process oscillate and slow down the training speed [8], [38]. Compared with the two-player framework NGAN, the proposed three-player frameworks ADIB and ADIM by design can already reduce the variance of gradients. This is because the three-player frameworks introduce the teacher, and the gradients from the teacher often have lower variance than those from the discriminator. Moreover, we propose to obtain gradients with even lower-variance by smoothing the discrete samples (i.e., fake labels) propagated between the student (or teacher) and the discriminator into continuous samples with a reparameterization trick [22], [29]. The reparameterization trick allows us to attain sufficient control over the variance, and hence reduces the variance of gradients from the discriminator. In the section, we will focus on ADIM and the same techniques can also be applied to ADIB [42].

We begin by showing that the high-variance of a random variable can be reduced by a low-variance random variable.

Lemma 4.5. *Random variables X and Y have finite variance and satisfy an inequality $\mathrm{Var}(X) \le \mathrm{Var}(Y)$. For any random variable $Z = \lambda X + (1-\lambda)Y$ $(0 \le \lambda \le 1)$, we have $\mathrm{Var}(Z) \le \mathrm{Var}(Y)$.*

Proof. Given $\mathrm{Var}(X) \le \mathrm{Var}(Y)$, the covariance $\mathrm{Cov}(X,Y)$ is no more than the variance $\mathrm{Var}(Y)$ because

$$\mathrm{Cov}(X,Y) \le |\mathrm{Cov}(X,Y)| \le \sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)} \le \mathrm{Var}(Y) .$$

Since $Z = \lambda X + (1-\lambda)Y$, we rewrite the variance $\mathrm{Var}(Z)$ as

$$\begin{aligned}&\lambda^2\,\mathrm{Var}(X) + 2\lambda(1-\lambda)\,\mathrm{Cov}(X,Y) + (1-\lambda)^2\,\mathrm{Var}(Y)\\ \le\ &\lambda^2\,\mathrm{Var}(Y) + 2\lambda(1-\lambda)\,\mathrm{Var}(Y) + (1-\lambda)^2\,\mathrm{Var}(Y) .\end{aligned}$$

This right hand side is $\mathrm{Var}(Y)$, completing the proof. □

Given the above lemma, we show that ADIM by design can reduce the variance of gradients by introducing the teacher into NGAN. In NGAN, the student only receives gradients $\nabla_{\theta_s}\mathcal{V}_{\mathrm{NGAN}}$ from the discriminator, whereas in ADIM the student receives gradients $\nabla_{\theta_s}\mathcal{V}_{\mathrm{ADIM}}$ from both the discriminator and the teacher, i.e.,

$$\nabla_{\theta_s}\mathcal{V}_{\mathrm{NGAN}} = \nabla^D_{\theta_s}\,, \quad \nabla_{\theta_s}\mathcal{V}_{\mathrm{ADIM}} = \omega\nabla^T_{\theta_s} + (1-\omega)\nabla^D_{\theta_s}\,, \tag{8}$$

where $0 \le \omega \le 1$, $\nabla^T_{\theta_s}$ and $\nabla^D_{\theta_s}$ are the gradients of the distillation and the adversarial losses w.r.t. the parameters of the student. In practice, we observe that the gradients of the distillation loss usually have lower variance than those of the adversarial loss (see Section 5 for more details), which is also consistent with previous findings [18], [35], Therefore, according to Lemma 4.5, it can be shown that the gradients w.r.t. the parameters of the student in ADIM have lower variance than those in NGAN, i.e.,

$$\begin{aligned}&\mathrm{Var}(\nabla^T_{\theta_s}) \le \mathrm{Var}(\nabla^D_{\theta_s})\\ \Rightarrow\ &\mathrm{Var}(\nabla_{\theta_s}\mathcal{V}_{\mathrm{ADIM}}) \le \mathrm{Var}(\nabla_{\theta_s}\mathcal{V}_{\mathrm{NGAN}}) .\end{aligned} \tag{9}$$

We further reduce the variance of gradients with a reparameterization trick, in particular, the Gumbel-Softmax (GS) trick [22], [29]. The essence of the GS trick is to reparameterize generating discrete samples from the student into a differentiable function of the original parameters and an additional random variable that obeys a Gumbel distribution. Since the student defines a categorical distribution, we adopt a concrete distribution $q_s(y|\boldsymbol{x})$ to perform the GS trick [22]. Specifically, we define the concrete distribution as

$$q_s(y|\boldsymbol{x}) = \mathrm{softmax}\left(\frac{\log p_s(y|\boldsymbol{x}) + g}{\tau}\right), \tag{10}$$

where $g \sim \mathrm{Gumbel}(0,1)$ is a sample from the Gumbel distribution [1] and $\tau > 0$ is a temperature hyper-parameter. We apply the concrete distribution $q_s(y|\boldsymbol{x})$ to generate continuous samples and use the continuous samples to compute the gradients via the REINFORCE algorithm [40]

$$\begin{aligned}\nabla_{\theta_s}\mathcal{V}_{\mathrm{ADIM}} &= \nabla_{\theta_s}(\omega_s\mathbb{E}_{y\sim p_s}[\log D_s(\boldsymbol{x},y)])\,,\\ &= \omega_s\mathbb{E}_{y\sim p_s}[\nabla_{\theta_s}\log p_s(y|\boldsymbol{x})\log D_s(\boldsymbol{x},y)]\,,\\ &\approx \omega_s\mathbb{E}_{y\sim q_s}[\nabla_{\theta_s}\log q_s(y|\boldsymbol{x})\log D_s(\boldsymbol{x},y)]\,.\end{aligned} \tag{11}$$

We leverage the temperature hyper-parameter τ to control the variance of gradients. With a high temperature, the continuous samples from the concrete distribution are smooth, which results in low-variance gradient estimates [29]. A limitation of the concrete distribution is that with a high temperature, it becomes a less accurate approximation to the original categorical distribution, which leads to biased gradient estimates. We overcome this limitation by annealing the concrete distribution into the original categorical distribution: we use a high temperature at the beginning and anneal the temperature to a small and non-zero value during training.

Besides improving the training of the student, we also apply the GS trick to improve the training of the teacher. We use the standard back-propagation to update the parameters of the discriminator [33]. The overall logic of training ADIM is summarized in Algorithm 1. The three players are first initialized with random values and then trained alternatively via minibatch stochastic gradient descent.

4.4 Correlating Adversarial and Distillation Losses

During the training of ADIM, we observe that the gradients of the adversarial and the distillation losses have a larger cosine similarity when the hyper-parameters vary across a certain range (e.g., Fig. 7). This observation motivates us to explore the correlation between the adversarial and the distillation losses. In this section, we show that in theory

[1] Samples from the Gumbel distribution can be obtained by first drawing $u \sim \mathrm{Uniform}(0,1)$ and then computing $g = -\log(-\log u)$.

Algorithm 1: Minibatch stochastic gradient descent training of ADIM using the Gumbel-Softmax trick.

```
Randomly initialize the parameters $\theta_s$, $\theta_t$, and $\theta_d$ of a student $S$, a teacher $T$, and a discriminator $D$, respectively.
for the number of epochs for training ADIM do
    for the number of steps for training the discriminator do
        Sample a batch $\{(\boldsymbol{x}_j^r, y_j^r)\}_{j=1}^{n_r} \sim p(\boldsymbol{x})p_r(y|\boldsymbol{x})$, $\{(\boldsymbol{x}_j^s, y_j^s)\}_{j=1}^{n_s} \sim p(\boldsymbol{x})q_s(y|\boldsymbol{x})$, and $\{(\boldsymbol{x}_j^t, y_j^t)\}_{j=1}^{n_t} \sim p(\boldsymbol{x})q_t(y|\boldsymbol{x})$.
        Update the discriminator by ascending along the gradients of the value function w.r.t. its parameters
            $\nabla_{\theta_d}\mathcal{V}_{\mathrm{ADIM}} = \frac{\omega_r}{n_r}\sum_{j=1}^{n_r}\nabla_{\theta_d}\log D_r(\boldsymbol{x}_j^r, y_j^r) + \frac{\omega_s}{n_s}\sum_{j=1}^{n_s}\nabla_{\theta_d}\log D_s(\boldsymbol{x}_j^s, y_j^s) + \frac{\omega_t}{n_t}\sum_{j=1}^{n_t}\nabla_{\theta_d}\log D_t(\boldsymbol{x}_j^t, y_j^t)\,.$
    for the number of steps for training the teacher do
        Sample a batch of feature-label pairs $\{(\boldsymbol{x}_j^t, y_j^t)\}_{j=1}^{n_t} \sim p(\boldsymbol{x})q_t(y|\boldsymbol{x})$ and $\{(\boldsymbol{x}_j^r, y_j^r)\}_{j=1}^{n_r} \sim p(\boldsymbol{x})p_r(y|\boldsymbol{x})$.
        Update the teacher by descending along the gradients of the value function w.r.t. its parameters
            $\nabla_{\theta_t}\mathcal{V}_{\mathrm{ADIM}} = \frac{\omega_t}{n_t}\sum_{j=1}^{n_t}\nabla_{\theta_t}\log q_t(y_j^t|\boldsymbol{x}_j^t)\log D_t(\boldsymbol{x}_j^t, y_j^t) + \frac{\mu}{n_r}\sum_{j=1}^{n_r}\nabla_{\theta_t}\mathcal{L}_{\mathrm{KD}}(p_s(y_j^r|\boldsymbol{x}_j^r)\,\|\,p_t(y_j^r|\boldsymbol{x}_j^r))\,.$
    for the number of steps for training the student do
        Sample a batch of feature-label pairs $\{(\boldsymbol{x}_j^s, y_j^s)\}_{j=1}^{n_s} \sim p(\boldsymbol{x})q_s(y|\boldsymbol{x})$ and $\{(\boldsymbol{x}_j^r, y_j^r)\}_{j=1}^{n_r} \sim p(\boldsymbol{x})p_r(y|\boldsymbol{x})$.
        Update the student by descending along the gradients of the value function w.r.t. its parameters
            $\nabla_{\theta_s}\mathcal{V}_{\mathrm{ADIM}} = \frac{\omega_s}{n_s}\sum_{j=1}^{n_s}\nabla_{\theta_s}\log q_s(y_j^s|\boldsymbol{x}_j^s)\log D_s(\boldsymbol{x}_j^s, y_j^s) + \frac{\nu}{n_r}\sum_{j=1}^{n_r}\nabla_{\theta_s}\mathcal{L}_{\mathrm{KD}}(p_t(y_j^r|\boldsymbol{x}_j^r)\,\|\,p_s(y_j^r|\boldsymbol{x}_j^r))\,.$
```

the adversarial losses used in ADIM can be reduced to the distillation losses under certain conditions.

We first show that given the optimal discriminator, the adversarial losses used in ADIM can be reduced to the distillation loss used to train the student.

Lemma 4.6. *Let $\upsilon = \frac{\omega_r}{\omega_t}$. As ω_s approaches 1, suppose that the discriminator is optimal, the adversarial losses are reduced to the distillation loss used for training the student*

$$\lim_{\omega_s \to 1}\frac{\mathcal{V}^*_{\mathrm{ADIM}}}{\omega_t} = \mathcal{L}_{\mathrm{KL}}(p_t\,\|\,p_s) + \upsilon\mathcal{L}_{\mathrm{KL}}(p_r\,\|\,p_s)\,. \tag{12}$$

Proof. As ω_s approaches 1, ω_t approaches 0. Hence, we can write the following Taylor expansion [21]

$$\mathcal{L}_{\mathrm{KL}}(p_s\,\|\,p_s + \Delta_s) = \Delta_s^{\intercal}H_s\Delta_s\,,$$

where H_s is a positive definite Hessian matrix, $\Delta_s = \omega_t\delta_s$ is a vector with infinitesimally small values, and δ_s is given by

$$\delta_s = \upsilon(p_r(y|\boldsymbol{x}) - p_s(y|\boldsymbol{x})) + (p_t(y|\boldsymbol{x}) - p_s(y|\boldsymbol{x}))\,.$$

Based on the Taylor expansion, we can compute the limit as

$$\begin{aligned}
&\lim_{\omega_s \to 1}\frac{\mathcal{V}^*_{\mathrm{ADIM}}}{\omega_t} = \lim_{\omega_s \to 1}\sum_{i \in \mathcal{I}}\frac{\omega_i\mathcal{L}_{\mathrm{KL}}(p_i\,\|\,\omega_r p_r + \omega_s p_s + \omega_t p_t)}{\omega_t}\,,\\
&= \frac{\omega_r}{\omega_t}\mathcal{L}_{\mathrm{KL}}(p_r\,\|\,p_s) + \lim_{\omega_s \to 1}\frac{\omega_s}{\omega_t}\mathcal{L}_{\mathrm{KL}}(p_s\,\|\,p_s + \Delta_s) + \mathcal{L}_{\mathrm{KL}}(p_t\,\|\,p_s)\,,\\
&= \upsilon\mathcal{L}_{\mathrm{KL}}(p_r\,\|\,p_s) + \lim_{\omega_s \to 1}\omega_t\delta_s^{\intercal}H_s\delta_s + \mathcal{L}_{\mathrm{KL}}(p_t\,\|\,p_s)\,,
\end{aligned}$$

which is equal to the distillation loss used to train the student. This is because as ω_s goes to 1, ω_t goes to 0 and hence the second term vanishes. This completes the proof. □

Since the student and the teacher are symmetric up to privileged provisions in ADIM, we show that the same adversarial losses can also be reduced to the distillation loss used to train the teacher. Due to page limit, we omit the proof which can essentially be obtained by switching the student and the teacher in the proof of Lemma 4.6.

Lemma 4.7. *Let $\upsilon = \frac{\omega_r}{\omega_s}$. As ω_t approaches 1, suppose that the discriminator is optimal, the adversarial losses are reduced to the distillation loss used for training the teacher*

$$\lim_{\omega_t \to 1}\frac{\mathcal{V}^*_{\mathrm{ADIM}}}{\omega_s} = \mathcal{L}_{\mathrm{KL}}(p_s\,\|\,p_t) + \upsilon\mathcal{L}_{\mathrm{KL}}(p_r\,\|\,p_t)\,. \tag{13}$$

Lemmas 4.6 and 4.7 show the generality of the adversarial losses: the distillation losses are different limits of the adversarial losses given the optimal discriminator. Note that the conditions, under which Lemmas 4.6 and 4.7 hold, are often hard to satisfy in practice: (1) The optimal discriminator is difficult to obtain due to the dependence on the student and the teacher, both of which keep changing during training. (2) The limits of the adversarial losses cannot be reached because the hyper-parameters are usually set within a reasonable range for good empirical performance. However, finding such a tight correlation between the adversarial and the distillation losses can help us obtain more insights into the behaviors of ADIM. We will use these theoretical results to explain our observation in experiments (see Section 5).

5 Experiments

The proposed three-player frameworks can be applied to a wide range of multi-label learning tasks where privileged provisions are available. We use the task of deep model compression to illustrate the applicability of our frameworks in Section 5.1, where intensive computation resources are privileged provisions. We also apply our frameworks to the task of image tag recommendation in Section 5.2, where extra textual features are privileged provisions.

First, we briefly describe the common experimental setup used across different tasks. We use Tensorflow [1] to implement the proposed ADIB and ADIM frameworks. We use two widely adopted formulations of distillation losses: the L2 loss between logits [7] and the Kullback-Leibler divergence between distributions [18]. The two formulations exhibit comparable results and the results presented in this paper are based on the Kullback-Leibler divergence. To

TABLE 1: Accuracy in deep model compression (n is the number of training images). Our approaches are ADIB and ADIM.

	MNIST				CIFAR			
	$n = 100$	$n = 1,000$	$n = 10,000$	$n = 50,000$	$n = 500$	$n = 5,000$	$n = 10,000$	$n = 50,000$
CODIS	74.02 ± 0.13	95.77 ± 0.10	98.89 ± 0.08	99.31 ± 0.06	54.17 ± 0.20	77.82 ± 0.14	81.60 ± 0.13	85.12 ± 0.11
DISTN	68.34 ± 0.06	93.97 ± 0.08	98.79 ± 0.07	99.26 ± 0.05	50.92 ± 0.18	76.59 ± 0.15	80.03 ± 0.09	83.32 ± 0.08
NOISY	66.53 ± 0.18	93.45 ± 0.11	98.58 ± 0.11	99.05 ± 0.10	50.18 ± 0.28	75.42 ± 0.19	79.89 ± 0.17	82.99 ± 0.12
MIMIC	67.35 ± 0.15	93.78 ± 0.13	98.65 ± 0.05	98.99 ± 0.04	51.74 ± 0.23	75.66 ± 0.17	80.32 ± 0.14	84.33 ± 0.10
NGAN	64.90 ± 0.31	93.60 ± 0.22	98.95 ± 0.19	99.36 ± 0.16	46.29 ± 0.32	76.11 ± 0.24	81.11 ± 0.22	85.34 ± 0.27
ADIB	77.95 ± 0.08	96.42 ± 0.09	99.25 ± 0.06	99.54 ± 0.05	57.56 ± 0.18	79.36 ± 0.16	83.02 ± 0.11	86.50 ± 0.10
ADIM	**78.70** ± 0.11	**97.20** ± 0.09	**99.38** ± 0.08	**99.61** ± 0.04	**59.10** ± 0.20	**80.53** ± 0.17	**83.93** ± 0.10	**87.03** ± 0.09

TABLE 2: Storage and runtime complexity of the student (S) and the teacher (T) in the task of deep model compression.

Dataset	Model	Implementation	#Parameters	#Flops
MNIST	S	MLP	1.28M	2.55M
	T	LeNet	4.62M	9.23M
CIFAR	S	LeNet	1.03M	2.06M
	T	DenseNet	15.58M	77.66M

TABLE 3: Storage and training complexity in deep model compression. The training time of ADIB is regarded as 1×.

Dataset	Approach	#Parameters	#Flops	Training (s)
MNIST	ADIB	10.51M	21.01M	1×
	ADIM	15.13M	30.24M	1.23×
CIFAR	ADIB	32.30M	156.09M	1×
	ADIM	47.78M	233.10M	1.17×

compute the categorical distribution defined by the student, we apply a softmax function to a scoring function $f(\boldsymbol{x}, y; \theta_s)$ that does not use privileged provisions

$$p_s(y|\boldsymbol{x}) = \mathrm{softmax}(f(\boldsymbol{x}, y; \theta_s)) \,,\ y \in \mathcal{Y} \,. \tag{14}$$

In contrast, to compute the categorical distribution defined by the teacher, we apply a softmax function to a scoring function $g^{\varrho}(\boldsymbol{x}, y; \theta_t)$ that makes use of privileged provisions

$$p_t(y|\boldsymbol{x}) = \mathrm{softmax}(g^{\varrho}(\boldsymbol{x}, y; \theta_t)) \,,\ y \in \mathcal{Y} \,. \tag{15}$$

Like the teacher, the discriminator can also use privileged provisions. Hence, we implement the binary-class discriminator by applying a sigmoid function to the same scoring function $g^{\varrho}(\boldsymbol{x}, y; \theta_d)$ with a different set of parameters

$$D(\boldsymbol{x}, y) = \mathrm{sigmoid}(g^{\varrho}(\boldsymbol{x}, y; \theta_d)) \,. \tag{16}$$

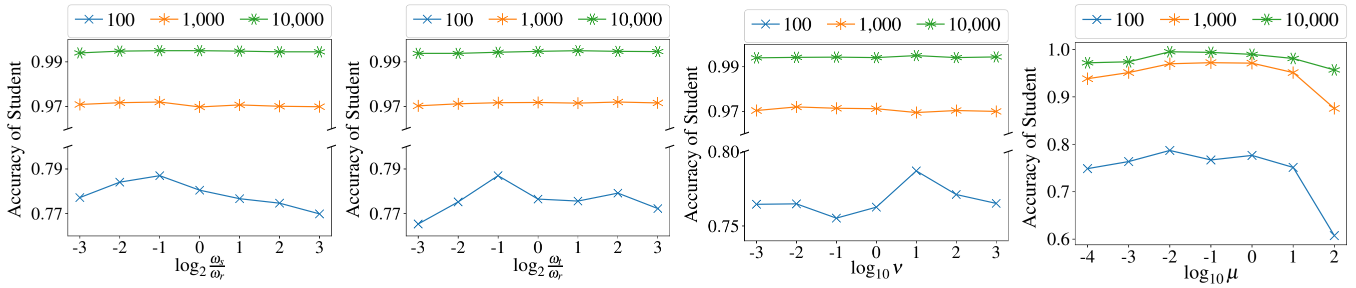
For simplicity, we implement the multi-class discriminator with two binary-class discriminators: one decides whether a label is real or fake, and the other decides whether a fake label is generated by the student or the teacher. The scoring functions are task-specific and are detailed in respective sections. We search the hyper-parameters within $0 < \omega_s < 1$, $0 < \omega_t < 1$, $0.001 < \nu < 1000$, $0.0001 < \mu < 100$ based on validation performance. We apply the GS trick when training ADIB and ADIM in all experiments, unless otherwise stated. We find that a reasonable annealing schedule for the temperature hyper-parameter τ is to start with a large value (10.) and exponentially decay it to a small value (0.1).

5.1 Deep Model Compression

Deep model compression aims at improving the deployability of deep models on portable devices such as smart phones by reducing the storage and the runtime complexity of such models. Since we usually train deep models on powerful servers, we treat extensive computational resources available at training as privileged provisions in this task.

Experimental Setup. We experiment with the widely adopted MNIST [26] and CIFAR [23] datasets. The MNIST dataset has 60,000 grayscale images (50,000 for training and 10,000 for testing) with 10 different label classes. For fair comparison with previous work [35], we do not preprocess the images on MNIST. The CIFAR dataset has 60,000 colored images (50,000 for training and 10,000 for testing) with 10 different label classes. We preprocess the images by subtracting per-pixel mean and augment the training dataset by mirrored images. On MNIST, we implement the scoring functions $f(\boldsymbol{x}, y)$ and $g^{\varrho}(\boldsymbol{x}, y)$ based on an MLP and a LeNet. On CIFAR, we implement the scoring functions $f(\boldsymbol{x}, y)$ and $g^{\varrho}(\boldsymbol{x}, y)$ based on a LeNet and a DenseNet. We detail the architectures of the MLP, the LeNets, and the DenseNet in the appendix (see Section A). We summarize the storage and the runtime complexity of the student and the teacher by the number of parameters and floating point operations (flops) in Table 2. We evaluate the approaches over 10 different runs with random initializations and report the mean accuracy and the standard deviation. Since the focus of this paper is to achieve a better accuracy for a given architecture suitable for inference, we defer to the appendix comparing the student and the teacher in terms of ratio of compression and drop of accuracy (see Section C).

Overall Results. First, we compare **ADIB** and **ADIM** with **NGAN** and KD-based approaches including **MIMIC** [7], **DISTN** [18], **NOISY** [35], and **CODIS** [2] in terms of accuracy. We vary the number of training images from 100 to 50,000. We show the results on MNIST and CIFAR in Table 1. The proposed ADIM performs the best on both datasets, e.g., ADIM (78.70%) outperforms CODIS (74.02%) by 6.32% on MNIST. Although the architectures of the student are the same in ADIB and ADIM, we can see that ADIM is more accurate than ADIB at the cost of increased storage and training complexity, as shown in Table 3. We also find that ADIM requires a smaller number of training instances than NGAN does to reach the same level of accuracy. For example, ADIM using fewer training samples (10,000) has a higher accuracy (99.38% vs. 99.36%) than NGAN using more training samples (50,000) on MNIST. This can be explained by that the teacher of ADIM provides soft targets for training the



(a) Effects of varying $\frac{\omega_s}{\omega_r}$ (b) Effects of varying $\frac{\omega_t}{\omega_r}$ (c) Effects of varying ν (d) Effects of varying μ

Fig. 3: Effects of the hyper-parameters in ADIM using 100, 1,000, and 10,000 training images on the MNIST dataset.

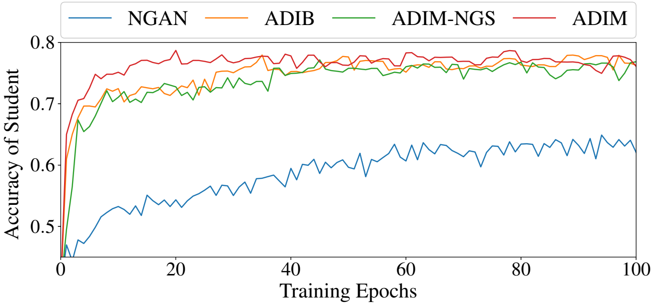
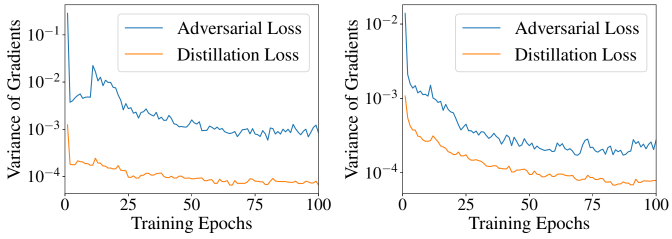
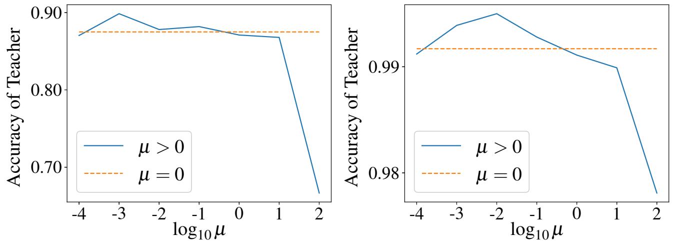


Fig. 4: Training curves of the student on MNIST.



(a) Training ADIM-NGS (b) Training ADIM

Fig. 5: Variance of the gradients w.r.t. the student on MNIST.



(a) 100 training images (b) 10,000 training images

Fig. 6: Accuracy of the teacher on MNIST with varying μ.

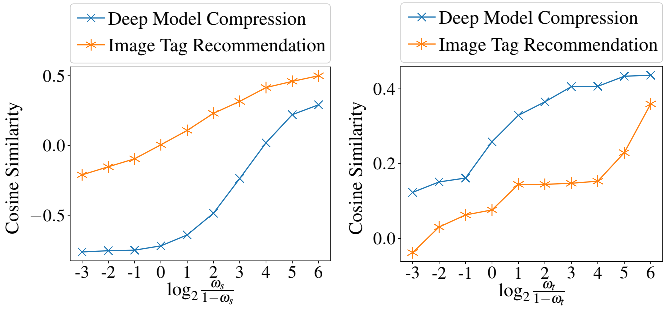
student. The soft targets have high entropy and reveal much useful information about training instances. Hence, the soft targets impose much more constraints on the training than the real labels do, which reduces the number of instances required to train the student. We further compare NGAN with the KD-based approaches. We observe that NGAN performs better when a large number of training instances are available (e.g., 50,000 training images on CIFAR), while KD-based approaches perform better when a small number of training images are available (e.g., 500 training images on CIFAR). This is consistent with our analyses in Section 3 that NGAN can learn the real data distribution better, which however requires a large amount of training data.

Training Speed. Next, we investigate the training speed of various approaches in terms of the number of training epochs required for convergence. To examine the benefits of using the GS trick for training, we refer to ADIM that does not use the GS trick as **ADIM-NGS**. We present typical training curves (i.e., accuracy against training epochs) of the student using 100 training images on MNIST in Fig. 4 (the training curves of using more training images are similar). We can see that NGAN converges to a worse accuracy even with a larger number of training epochs than the other approaches (involving a teacher) do. We also find that the training curve of NGAN is less stable than those of the other approaches after convergence. This is largely because the gradients that the student obtains from the discriminator (via the adversarial losses) have higher variance than those from the teacher (via the distillation losses), as shown in Fig. 5. By comparing ADIM with ADIM-NGS, we can see that the GS trick speeds up the training by around 50% and improves the accuracy by around 2% (from 77.20% to 78.70%). One possible reason is that the GS trick can effectively reduce the variance of the gradients from the discriminator as discussed in Section 4.3. This is also observed in our experiments by comparing the gradient variance of the adversarial loss in ADIM-NGS (see Fig. 5a) with that in ADIM (see Fig. 5b).

Ablation Study. Next, we study how the proposed ADIM achieves the highest accuracy. We present the accuracy of ADIM against the hyper-parameters using 100, 1,000, and 10,000 training images on MNIST in Fig. 3 (note the logarithmic scale of the x-axis). We find that ω_r, ω_s, and ω_t have a relatively small effect on the accuracy, which suggests that ADIM is a robust framework. We can see that the teacher is important in improving the accuracy of the student especially when the number of training images is small. For example, if we set ν to a very small value (0.001), we get more than 2% drop in accuracy (from 78.70% to 76.46%) when using 100 training images. We also find that a large value of μ causes the accuracy of the student to deteriorate rapidly. Such accuracy deterioration occurs because the soft targets provided by the student are usually noisy. Emphasizing on training the teacher to predict the noisy targets decreases the accuracy of the teacher, as shown in Fig. 6. The decrease in the accuracy of the teacher in turn decreases the accuracy of the student via the distillation process. However, the teacher achieves a higher accuracy

TABLE 4: Accuracy in image tag recommendation on YFCC. The proposed approaches are ADIB and ADIM.

	Most Popular Tags						Randomly Sampled Tags					
	MAP	MRR	P@3	R@3	F@3	NDCG@3	MAP	MRR	P@3	R@3	F@3	NDCG@3
KVOTE	0.5755	0.5852	0.2320	0.4400	0.2339	0.5592	0.3970	0.4092	0.1623	0.2790	0.1575	0.3607
TPROP	0.6177	0.6270	0.2420	0.5281	0.2811	0.6103	0.4512	0.4636	0.1883	0.3244	0.1810	0.4225
TFEAT	0.6417	0.6503	0.2560	0.5420	0.2871	0.6371	0.5149	0.5309	0.2002	0.4132	0.2195	0.4990
REXMP	0.7015	0.7122	0.2720	0.6285	0.3324	0.6999	0.5205	0.5331	0.2228	0.4450	0.2427	0.5377
NGAN	0.7432	0.7555	0.2892	0.6676	0.3516	0.7465	0.5791	0.5911	0.2415	0.4904	0.2693	0.5834
ADIB	0.7787	0.7905	0.3047	0.6971	0.3678	0.7846	0.6302	0.6452	0.2572	0.5403	0.2946	0.6255
ADIM	**0.7862**	**0.7980**	**0.3060**	**0.7032**	**0.3687**	**0.7914**	**0.6480**	**0.6623**	**0.2620**	**0.5462**	**0.2950**	**0.6416**



(a) Gradients w.r.t. the student (b) Gradients w.r.t. the teacher

Fig. 7: Cosine similarity between the gradients of adversarial and distillation losses w.r.t. the student (a) or the teacher (b).

when learning from the student, e.g., the accuracy of setting $\mu = 0.001$ is higher than that of setting $\mu = 0$, as shown in Fig. 6. To study the correlation between the adversarial and the distillation losses, we compute the cosine similarity between their gradients during training. We average the cosine similarity over training epochs when training ADIM using 100 training images on MNIST and present the results in Fig. 7. Fig. 7a shows that the gradients of the adversarial and the distillation losses w.r.t. the student become more similar as ω_s goes to 1. Fig. 7b shows that the gradients of the adversarial and the distillation losses w.r.t. the teacher become more similar as ω_t goes to 1. We do not observe such trend if we set ω_s (ω_t) to a very small value ($< 2^{-3}$) or to a very large value ($> 2^6$). This is because when we set the hyper-parameters outside a reasonable range, ADIM does not perform well and the discriminator can be far from the optimality. These results are consistent with our theoretical results in Lemmas 4.6 and 4.7. We obtain similar results about the effects of the hyper-parameters on CIFAR.

5.2 Image Tag Recommendation

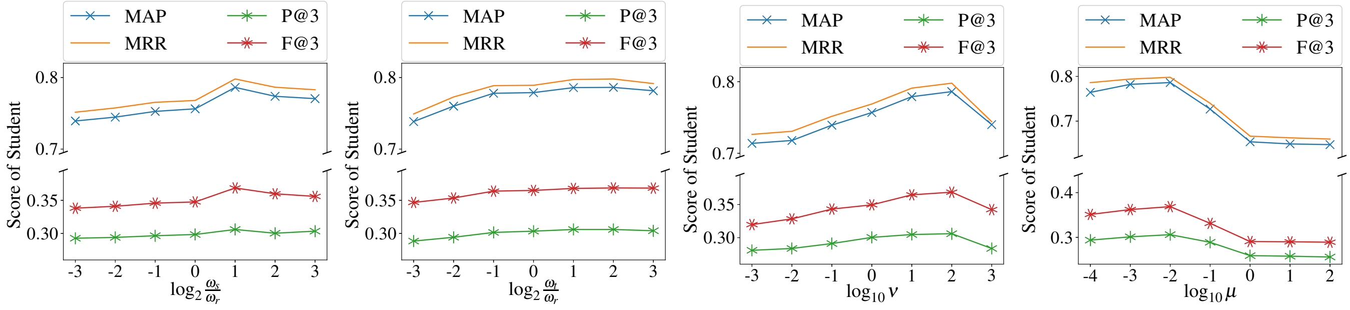
Next, we experiment with the task of image tag recommendation, which aims to recommend relevant tags for users to label their images uploaded to image-hosting websites such as Flickr. Specifically, the goal is to recommend relevant tags right after a user uploads an image. This way, the user can just select from the recommended tags instead of inputting tags, which is inconvenient. Users may continue to add extra texts, e.g., titles and descriptions about the uploaded image. We only use such extra texts for training as privileged provisions. The student, once trained, only takes an uploaded image as input to recommend tags at inference.

Experimental Setup. We use the Yahoo Flickr Creative Commons 100 Million (YFCC) dataset [37]. To simulate the case where extra texts about images are available at training, we randomly sample 20,000 images with titles or descriptions for training and sample another 2,000 images for testing. We create a dataset of images labeled with the 200 most popular tags and create another one labeled with 200 randomly sampled tags. Following an earlier study [3], we use a VGGNet [36] pretrained on ImageNet [13] to extract visual features and use a LSTM [19] with pretrained word embeddings [31] to learn textual features. We implement the scoring function $f(\boldsymbol{x}, y)$ as an MLP taking visual features as input and $g^{\varrho}(\boldsymbol{x}, y)$ as an MLP taking element-wise product of the visual and the textual features as input. The architectures of the MLPs are detailed in the appendix (see Section B). We use precision (P@3), recall (R@3), F-measure (F@3), normalized discounted cumulative gain (NDCG@3), mean average precision (MAP), and mean reciprocal ranking (MRR) to evaluate the accuracy of the student and the teacher.

Overall Results. First, we compare the students of ADIB and ADIM with **KNN** [30], **TPROP** [17], **TFEAT** [10], and **REXMP** [27] in terms of accuracy. The results using the most popular and using the randomly sampled tags are presented in Table 4. We can see that ADIM achieves consistent improvements over the other approaches under all the evaluation metrics. Although ADIM does not explicitly model the semantic similarity between two tags like what REXMP does, it still makes better recommendations than REXMP does. The reason is that in ADIM, the teacher provides the student with soft targets at training. The soft targets contain a rich similarity structure over tags which cannot be modeled well by any pairwise similarities between tags in REXMP. For example, an image labeled with a tag `volleyball` is provided with a soft target assigning probabilities of 10^{-2} to `basketball`, 10^{-4} to `baseball`, and 10^{-8} to `dragonfly`. The reason why the teacher generalizes is reflected in the relative probabilities over tags, which can be used for guiding the student to generalize better.

Training Speed. We also examine the training speed of the approaches. We use P@3 to evaluate the accuracy of the student and present the training curves using the most popular tags in Fig. 9. We can see that: (1) ADIM learns a more accurate student with a smaller number of training epochs than the other approaches; (2) NGAN is unstable largely due to high variance of the gradients from the adversarial losses, as shown in Fig. 10. We observe a similar trend when using the other metrics to evaluate the accuracy.

Ablation Study. Last, we explore how the accuracy of



(a) Effects of varying $\frac{\omega_s}{\omega_r}$ (b) Effects of varying $\frac{\omega_t}{\omega_r}$ (c) Effects of varying ν (d) Effects of varying μ

Fig. 8: Effects of the hyper-parameters in ADIM using the most popular tags on the YFCC dataset.

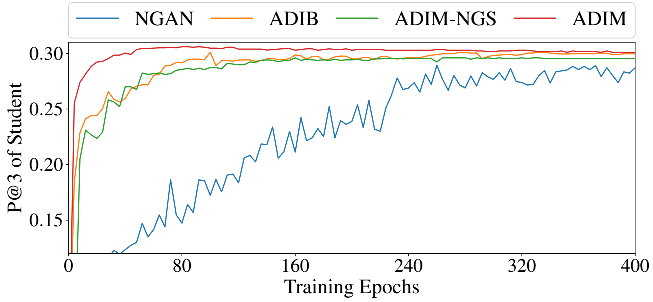
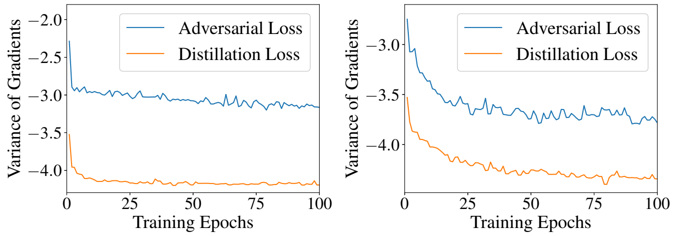


Fig. 9: Training curves of the student on YFCC.

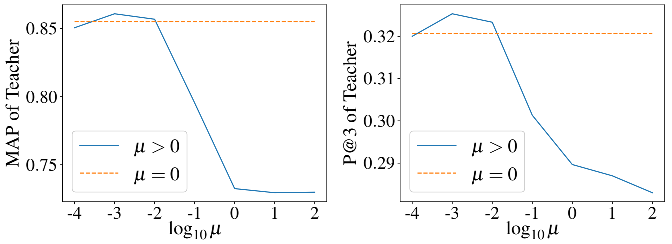


(a) Training ADIM-NGS (b) Training ADIM

Fig. 10: Variance of the gradients w.r.t. the student on YFCC.

ADIM varies against the hyper-parameters. The results using the most popular tags are presented in Fig. 8. We can see that the effects of the hyper-parameters in image tag recommendation are consistent with those in deep model compression. We also study the benefits of the teacher learning from the student by varying μ. We use MAP and P@3 to evaluate the accuracy of the teacher and present the results using the most popular tags in Fig. 11. We can see that the teacher achieves the highest accuracy when learning from the student with $\mu = 0.001$. We further compute the cosine similarity, averaged over training epochs, between the gradients of the adversarial and the distillation losses. Fig. 7 shows the results using the most popular tags, which are similar to those in deep model compression.

6 Conclusion



(a) MAP (b) P@3

Fig. 11: Accuracy of the teacher on YFCC with varying μ.

We proposed a novel training framework, adversarial distillation, for learning with privileged provisions. We formulated adversarial distillation as a minimax game where a student and a teacher compete against a discriminator via adversarial losses while learning from each other via distillation losses. We proposed two variants of adversarial distillation: one uses a binary-class discriminator for more efficient training and the other uses a multi-class discriminator for more accurate inference. We proved that both of the variants guarantee the equilibrium where the student and the teacher fit the real data distribution perfectly. We proposed a Gumbel-Softmax trick to control the variance of gradients and hence obtain low-variance gradient updates during training. We proved that the adversarial losses can be reduced to the distillation losses used to train the student and the teacher. We showed that adversarial distillation significantly outperforms the state-of-the-art in the tasks of image tag recommendation and deep model compression. We also showed that the Gumbel-Softmax trick speeds up the training by reducing the variance of the gradients from the discriminator.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, 2016.

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

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

[4] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

[5] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[6] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*, 2017.

[7] J. Ba and R. Caruana. Do deep nets really need to be deep? In *NeurIPS*, 2014.

[8] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

[9] C. Bucilu, R. Caruana, and A. Niculescu-Mizil. Model compression. In *SIGKDD*, 2006.

[10] L. Chen, D. Xu, I. W. Tsang, and J. Luo. Tag-based image retrieval improved by augmented features and group-based refinement. *IEEE Transactions on Multimedia*, 2012.

[11] X. Chen, Y. Zhang, H. Xu, Z. Qin, and H. Zha. Adversarial distillation for efficient recommendation with external knowledge. *TOIS*, 2018.

[12] L. Chongxuan, T. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. In *NeurIPS*, 2017.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] S. Feizi, C. Suh, F. Xia, and D. Tse. Understanding gans: the lqg setting. *arXiv preprint arXiv:1710.10793*, 2017.

[15] Z. Gan, L. Chen, W. Wang, Y. Pu, Y. Zhang, H. Liu, C. Li, and L. Carin. Triangle generative adversarial networks. In *NeurIPS*, 2017.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[17] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[18] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NeurIPS workshop*, 2014.

[19] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997.

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

[21] F. Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015.

[22] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

[23] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[24] X. Lan, X. Zhu, and S. Gong. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 2018.

[25] M. Lapin, M. Hein, and B. Schiele. Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. *TPAMI*, 2018.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[27] X. Li and C. G. Snoek. Classifying tag relevance with relevant positive and negative examples. In *ACMMM*, 2013.

[28] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. Unifying distillation and privileged information. In *ICLR*, 2016.

[29] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.

[30] A. Makadia, V. Pavlovic, and S. Kumar. Baselines for image annotation. *IJCV*, 2010.

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013.

[32] Y. Pu, S. Dai, Z. Gan, W. Wang, G. Wang, Y. Zhang, R. Henao, and L. C. Duke. Jointgan: Multi-domain joint distribution learning with generative adversarial nets. In *ICML*, 2018.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[34] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

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

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[37] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: the new data in multimedia research. *Communications of the ACM*, 2016.

[38] G. Tucker, A. Mnih, C. J. Maddison, J. Lawson, and J. Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *NeurIPS*, 2017.

[39] V. Vapnik and R. Izmailov. Learning using privileged information: similarity control and knowledge transfer. *JMLR*, 2015.

[40] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *SIGIR*, 2017.

[41] R. J. Wang, X. Li, and C. X. Ling. Pelee: a real-time object detection system on mobile devices. In *NeurIPS*, 2018.

[42] X. Wang, R. Zhang, Y. Sun, and J. Qi. Kdgan: Knowledge distillation with generative adversarial networks. In *NeurIPS*, 2018.

[43] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. Hcp: A flexible cnn framework for multi-label image classification. *TPAMI*, 2016.

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

[45] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.

[46] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *TKDE*, 2014.

[47] Y. Zhang, Z. Gan, and L. Carin. Generating text via adversarial training. In *NeurIPS workshop on Adversarial Training*, 2016.

[48] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin. Adversarial feature matching for text generation. In *ICML*, 2017.

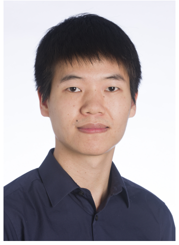
[49] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu. Deep mutual learning. *arXiv preprint arXiv:1706.00384*, 2017.

[50] X. Zhu, S. Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 2018.

Xiaojie Wang is a Ph.D. candidate in the School of Computing and Information Systems at University of Melbourne. He received his B.S degrees in Applied Mathematics and Computer Science from Renmin University of China in 2016. His research interests include information retrieval, recommender systems, and machine learning.

Rui Zhang is a Professor in the School of Computing and Information Systems at University of Melbourne. His research interests include big data, data mining, and machine learning. Professor Zhang has won several awards including Future Fellowship by the Australian Research Council in 2012, Chris Wallace Award for Outstanding Research by the Computing Research and Education Association of Australasia in 2015, and Google Faculty Research Award in 2017.

Yu Sun is a software engineer on machine learning at Twitter. He obtained his Ph.D. from University of Melbourne in 2017. He worked for seven months at Nanyang Technological University in 2012. He was a software engineering intern at Google Research in 2017. He visited the Social Computing Group of Microsoft Research Asia in 2015. His research interests include context-aware recommendation, personalization, and spatial/temporal data mining.



Jianzhong Qi is a lecturer in the School of Computing and Information Systems at University of Melbourne. He received his Ph.D degree from University of Melbourne in 2014. He has been an intern at Microsoft Redmond in 2013 and an Eshbach Visiting Scholar at Northwestern University in 2017, respectively. His research interests include spatio-temporal databases and natural language processing.