Rethinking Importance Weighting for Deep Learning under Distribution Shift

Tongtong Fang*  
KTH & RIKEN  
tongtongf@hotmail.com

Nan Lu*  
Univ. of Tokyo & RIKEN  
lu@ms.k.u-tokyo.ac.jp

Gang Niu  
RIKEN  
gang.niu@riken.jp

Masashi Sugiyama  
RIKEN & Univ. of Tokyo  
sugi@k.u-tokyo.ac.jp

Abstract

Under distribution shift (DS) where the training data distribution differs from the test one, a powerful technique is importance weighting (IW) which handles DS in two separate steps: weight estimation (WE) estimates the test-over-training density ratio and weighted classification (WC) trains the classifier from weighted training data. However, IW cannot work well on complex data, since WE is incompatible with deep learning. In this paper, we rethink IW and theoretically show it suffers from a circular dependency: we need not only WE for WC, but also WC for WE where a trained deep classifier is used as the feature extractor (FE). To cut off the dependency, we try to pretrain FE from unweighted training data, which leads to biased FE. To overcome the bias, we propose an end-to-end solution dynamic IW that iterates between WE and WC and combines them in a seamless manner, and hence our WE can also enjoy deep networks and stochastic optimizers indirectly. Experiments with two representative DSs on Fashion-MNIST and CIFAR-10/100 demonstrate that dynamic IW compares favorably with state-of-the-art methods.

1 Introduction

Supervised deep learning is extremely successful (Goodfellow et al., 2016), while its success relies highly on the fact that training and test data come from the same distribution. A big challenge in this deep learning age is distribution shift or dataset shift where training and test data come from

*Equal contribution.
two different distributions (Quionero-Candela et al., 2009; Sugiyama and Kawanabe, 2012; Pan and Yang, 2009): the training data are drawn from \( p_{\text{tr}}(x, y) \), the test data are drawn from \( p_{\text{te}}(x, y) \), and \( p_{\text{tr}}(x, y) \neq p_{\text{te}}(x, y) \). Under distribution shift, standard supervised deep learning will lead to deep classifiers which are biased to training data and whose performance will drop on test data.

It is usually assumed under distribution shift that \( p_{\text{te}}(x, y) \) is absolutely continuous w.r.t. \( p_{\text{tr}}(x, y) \), i.e., \( p_{\text{tr}}(x, y) = 0 \) implies \( p_{\text{te}}(x, y) = 0 \). Then, there is a function \( w^*(x, y) = p_{\text{te}}(x, y) / p_{\text{tr}}(x, y) \), such that for any function \( f \) of \( x \) and \( y \),

\[
\mathbb{E}_{p_{\text{te}}(x, y)}[f(x, y)] = \mathbb{E}_{p_{\text{tr}}(x, y)}[w^*(x, y)f(x, y)].
\]

Eq. (1) means after taking proper weights into account, the weighted expectation of \( f \) over \( p_{\text{tr}}(x, y) \) becomes unbiased, no matter if \( f \) is a loss to be minimized or a reward to be maximized. Thanks to this property, we can use importance weighting (IW) (Shimodaira, 2000; Sugiyama et al., 2007a; Huang et al., 2007; Sugiyama et al., 2007b, 2008; Kanamori et al., 2009) to handle distribution shift in two separate steps: (i) weight estimation (WE) estimates \( w^* \) from the training data and a tiny set of validation data drawn from \( p_{\text{te}}(x, y) \) or \( p_{\text{te}}(x) \); (ii) weighted classification (WC) approximates \( \mathbb{E}_{p_{\text{tr}}(x,y)}[w^*(x, y)f(x, y)] \) from the training data and then trains our favorite classifiers as if there is no distribution shift (Shimodaira, 2000). IW works very well if the form of data is simple, and has been the common practice of non-deep learning under distribution shift (Sugiyama et al., 2012).

 Nonetheless, IW cannot work well if the form of data is complex. Consider a \( k \)-class classification problem with an input domain \( \mathcal{X} \subset \mathbb{R}^d \) and an output domain \( \mathcal{Y} = \{1, \ldots, k\} \) where \( d \) is the input dimension, and let \( f : \mathcal{X} \rightarrow \mathbb{R}^k \) be the classifier to be trained for this problem. Here, \( w^* \) processes \( (d + 1) \)-dimensional input and \( f \) processes \( d \)-dimensional input, and consequently the WE step is not necessarily easier than the WC step. Thus, more expressive power is definitely needed in WE.

In this paper, we focus on improving IW to make it work for deep learning under distribution shift. We argue that it is difficult to boost the expressive power of WE for three reasons. Firstly, some WE methods are model-free such that they assign weights to data without a model of \( w^* \). Secondly, other...
WE methods are model-based and model-independent, but the optimizations are constrained and incompatible with stochastic optimization solvers, because \( \mathbb{E}_{p_{tr}(x,y)}[w^*(x,y)] = \mathbb{E}_{p_{te}(x,y)}[1] = 1 \). Finally, even if we ignore the constraint or satisfy it in each mini-batch, most powerful deep models nowadays are designed for classification and hard to train with WE objectives. Therefore, it is better to boost the expressive power by an external feature extractor (FE) inside \( f \), a deep classifier (DC) chosen for the classification problem to be solved. Going along this way, we encounter the circular dependency in Figure 1: originally we need \( w^* \) to train \( f \); now we need a trained \( f \) to estimate \( w^* \). This causality dilemma pushes us to ask which should come first: the chicken or the egg?

We think of two possible ways to solve the circular dependency, one pipelined and one end-to-end. The pipelined solution has two steps: (i) pretrain a DC as FE from unweighted training data and perform WE on the data transformed by FE; (ii) perform WC. Since the weights cannot change, we call this method static importance weighting (SIW), as illustrated in the top diagram of Figure 2. The DC as FE is trained without considering distribution shift, and we empirically confirm it is biased to training data. As a result, this naive solution is only a bit better than no DC/FE unfortunately.

On the other hand, the end-to-end solution, called dynamic importance weighting (DIW) and illustrated in the bottom diagram of Figure 2, has a single step: train a DC as FE from weighted training data (i.e., perform WC) and at the same time perform WE on the data transformed by FE in a seamless manner. More specifically, let \( \mathcal{W} \) be the set of importance weights initialized to be all ones and let \( f \) be initialized randomly. Subsequently, we update \( f \) for several epochs to pretrain it a little, and then we update both \( \mathcal{W} \) and \( f \) for the remaining epochs: in each mini-batch, \( \mathcal{W} \) is updated by an objective of WE where \( f \) is fixed and then \( f \) is updated by the objective of WC where \( \mathcal{W} \) is fixed in backpropagation. As a consequence, this more advanced solution gradually reduces the biases of \( \mathcal{W} \) and \( f \), which suggests that IW for deep learning nowadays can perform as well as IW for non-deep learning in the old days hopefully.

The rest of the paper is organized as follows. DIW is proposed in Sec. 2 with its applications given in Sec. 3. We discuss the related works in Sec. 4, and present the experiments in Sec. 5. Some more theoretical and experimental results can be found in the appendices.

## 2 Dynamic importance weighting

As mentioned earlier, under distribution shift, training and test data come from two different distributions \( p_{tr}(x,y) \) and \( p_{te}(x,y) \) (Quionero-Candela et al., 2009; Sugiyama and Kawanabe, 2012). Let \( \{(x_i^{tr},y_i^{tr})\}_{i=1}^{n_{tr}} \) be a set of i.i.d. training data sampled from \( p_{tr}(x,y) \) where \( n_{tr} \) is the training sample size, and \( \{(x_i^{v},y_i^{v})\}_{i=1}^{n_{v}} \) be a set of i.i.d. validation data sampled from \( p_{te}(x,y) \) where \( n_{v} \) is the validation sample size. We assume validation data are much less than training data, namely \( n_{v} \ll n_{tr} \), otherwise we can use validation data for training.

*We can update a weight by convexly combining its old value from the last epoch and its new value from DIW. This can stabilize the weight across epochs, in case that DIW is unstable when the batch size is small.*
Weighted classification From now on, we assume our classifier $f$ to be trained is a deep network parameterized by $\theta$ that is denoted by $f_\theta$. Let $\ell : \mathbb{R}^k \times \mathcal{Y} \to \mathbb{R}_+$ be the surrogate loss function for $k$-class classification, e.g., softmax cross-entropy loss. The classification risk of $f_\theta$ is defined as

$$R(f_\theta) = \mathbb{E}_{p_{\text{te}}(x,y)}[\ell(f_\theta(x), y)],$$

which is the performance measure we would like to optimize. According to Eq. (1), if $w^*(x, y)$ is given or $\mathcal{W}^* = \{w_i^* \mid w_i^* = w^*(x_i^{\text{tr}}, y_i^{\text{tr}})\}_{i=1}^{n_{\text{tr}}}$ is given, $R(f_\theta)$ can be approximated by

$$\hat{R}(f_\theta) = \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} w_i^* \ell(f_\theta(x_i^{\text{tr}}), y_i^{\text{tr}}),$$

which is the objective of weighted classification. The weighted empirical risk $\hat{R}(f_\theta)$ in Eq. (3) is an unbiased estimator of the risk $R(f_\theta)$ in Eq. (2), and hence the trained classifier as the minimizer of $\hat{R}(f_\theta)$ converges to the minimizer of $R(f_\theta)$ as $n_{\text{tr}}$ goes to infinity (Shimodaira, 2000; Sugiyama et al., 2007a; Huang et al., 2007; Sugiyama et al., 2007b, 2008; Kanamori et al., 2009).

Non-linear transformation of data Now, the issue is how to estimate the function $w^*$ or the set $\mathcal{W}^*$. As discussed earlier, we should boost the expressive power externally but not internally. This means we should apply a non-linear transformation of data rather than directly model $w^*(x, y)$ or $p_{\text{tr}}(x, y)$ and $p_{\text{te}}(x, y)$ by deep networks. Let $\pi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_{\pi}}$ or $\pi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_{\pi} - 1} \times \mathcal{Y}$ be a transformation where $d_\pi$ is the reduced dimension and $d_\pi \ll d$; let $z = \pi(x, y)$ be the transformed random variable whose source of randomness is $(x, y)$ exclusively. We expect that weight estimation on $z$ is much easier than on $(x, y)$. The feasibility of applying $\pi$ to transform data is justified below.

**Theorem 1.** For a fixed, deterministic and invertible transformation $\pi : (x, y) \mapsto z$, let $p_{\text{tr}}(z)$ and $p_{\text{te}}(z)$ be the probability density functions (PDFs) induced by $p_{\text{tr}}(x, y)$, $p_{\text{te}}(x, y)$, and $\pi$. Then,

$$w^*(x, y) = \frac{p_{\text{te}}(x, y)}{p_{\text{tr}}(x, y)} = \frac{p_{\text{te}}(z)}{p_{\text{tr}}(z)} = w^*(z).$$

**Proof.** Let $F_{\text{tr}}(x, y)$, $F_{\text{te}}(x, y)$, $F_{\text{tr}}(z)$ as well as $F_{\text{te}}(z)$ be the corresponding cumulative distribution functions (CDFs). By the definition of CDFs, the fundamental theorem of calculus, and three properties of $\pi$ namely $\pi$ is fixed, deterministic and invertible, it holds that

$$p_{\text{tr}}(x, y)dx = dF_{\text{tr}}(x, y) = dF_{\text{tr}}(z) = p_{\text{tr}}(z)dz,$$

$$p_{\text{te}}(x, y)dx = dF_{\text{te}}(x, y) = dF_{\text{te}}(z) = p_{\text{te}}(z)dz,$$

---

1 Here, it is implicitly assumed that PDFs $p_{\text{te}}(x)$ are Riemann-integrable and CDFs $F_{\text{te}}(x)$ are differentiable, and the proof is invalid if $p_{\text{te}}(x)$ are only Lebesgue-integrable and $F_{\text{te}}(x)$ are only absolutely continuous. The more formal proof is given as follows. Since $p_{\text{te}}(x, y)$ are Lebesgue-Stieltjes-integrable, we can use probability measures: for example, let $N_{x,y} \ni (x, y)$ be an arbitrary neighborhood around $(x, y)$, then as $N_{x,y} \rightarrow (x, y)$ where the convergence is w.r.t. the distance metric on $\mathcal{X} \times \mathcal{Y}$, it holds that

$$p_{\text{tr}}(x,y)d|N_{x,y}| = d\mu_{x,y,\text{tr}}(N_{x,y}) = d\mu_{x,y,\text{te}}(\pi(N_{x,y})) = p_{\text{tr}}(z)d|\pi(N_{x,y})|,$$

where $\mu_{x,y,\text{tr}}$ and $\mu_{x,y,\text{te}}$ are the corresponding probability measures, $\pi(N_{x,y}) = \{\pi(x', y') \mid (x', y') \in N_{x,y}\}$, and $\cdot$ denotes the Lebesgue measure of a set. This more formal proof may be more than needed, since $w^*$ is estimable only if $p_{\text{te}}(x)$ are continuous and $F_{\text{te}}(x)$ are continuously differentiable.
where $d$ denotes the differential operator, and
\[
    dF_s(x, y) = \frac{\partial}{\partial x} \left( \sum_{y' \leq y} \int_{x' \leq x} p_s(x', y') \, dx' - \sum_{y' < y} \int_{x' \leq x} p_s(x', y') \, dx' \right) \cdot dx.
\]

For simplicity, the continuous random variable $x$ and the discrete random variable $y$ are considered separately. Dividing Eq. (6) by Eq. (5) proves Eq. (4).

Theorem 1 requires that $\pi$ satisfies three properties: we cannot guarantee $dF_{tr}(z) = p_{te}(z) \, dz$ if $\pi$ is not fixed or $dF_{tr}(x, y) = dF_{tr}(z)$ if $\pi$ is not deterministic or invertible. As a result, when $\mathcal{W}$ is updated in DIW, $f_\theta$ is regarded as fixed, and it should be switched to the evaluation mode from the training mode to avoid the randomness due to dropout (Srivastava et al., 2014) or similar randomized algorithms. The invertibility of $\pi$ is non-trivial: it assumes that $\mathcal{X} \times \mathcal{Y}$ is generated by a manifold $\mathcal{M} \subset \mathbb{R}^d$ with an intrinsic dimension $d_m \leq d_t$, and $\pi^{-1}$ recovers the generating function from $\mathcal{M}$ to $\mathcal{X} \times \mathcal{Y}$. If $\pi$ is from parts of $f_\theta$, $f_\theta$ must be a reasonably good classifier so that $\pi$ compresses $\mathcal{X} \times \mathcal{Y}$ back to $\mathcal{M}$. This finding is the circular dependency in Figure 1 which is the major theoretical contribution.

**Practical choices of the transformation of data**  Let us take a closer look at practical choices of $\pi$. It seems obvious that $\pi$ can be $f_\theta$ as a whole or $f_\theta$ without the topmost layer. However, the latter drops $y$ and corresponds to assuming
\[
    p_{te}(y \mid x) \implies p_{te}(y) \cdot p_{te}(x \mid y) = p_{te}(x) = p_{te}(z) / p_{ti}(x),
\]

which is only possible under covariate shift (Pan and Yang, 2009; Shimodaira, 2000; Sugiyama et al., 2007b, 2008). It is conceptually a bad idea to attach $y$ after the latent representation of $x$, since the distance metric on $\mathcal{Y}$ is completely different. A better idea to take the information of $y$ into account would consist of three steps: first partition $\{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ and $\{(x_i^v, y_i^v)\}_{i=1}^{n_v}$ according to $y$, second estimate $p_{te}(y) / p_{ti}(y)$, and third invoke weight estimation $k$ times on $k$ partitions separately based on the following identity: let $w^*_y = p_{te}(y) / p_{ti}(y)$, then
\[
    \frac{p_{te}(x, y)}{p_{ti}(x, y)} = \frac{p_{te}(y) \cdot p_{te}(x \mid y)}{p_{ti}(y) \cdot p_{ti}(x \mid y)} = \frac{p_{te}(y \mid x)}{p_{ti}(x \mid y)} = w^*_y \cdot \frac{p_{te}(z \mid y)}{p_{ti}(z \mid y)}. \quad (8)
\]

That being said, in a mini-batch, invoking weight estimation $k$ times on $k$ partitions may be remarkably unreliable than invoking it once on the whole mini-batch.

To this end, we propose an alternative choice $\pi : (x, y) \mapsto \ell(f_{\theta}(x), y)$ that is motivated as follows. In practice, we are not sure about the existence of $\mathcal{M}$, we cannot check whether $d_m \leq d_t$ when $\mathcal{M}$ indeed exists, or it is computationally hard to confirm that $\pi$ is invertible. Consequently, Eqs. (7-8) may not hold or only hold approximately. As a matter of fact, Eq. (1) also only hold approximately after replacing the expectations with empirical averages, and it seems alright to go one step further. According to Eq. (1), there exists $w(x, y)$ such that for all possible $f(x, y)$,
\[
    \frac{1}{n_v} \sum_{i=1}^{n_v} f(x_i^v, y_i^v) \approx E_{p_{te}(x, y)}[f(x, y)] \approx E_{p_{te}(x, y)}[w(x, y) f(x, y)] \approx \frac{1}{n_t} \sum_{i=1}^{n_t} w_i f(x_i^t, y_i^t),
\]
with the general goal of IW, the goal of DIW is special and easy to achieve, and then there may be
where the left- and right-hand sides are conditioned on $w$
where $w = w(x_i^t, y_i^t)$ for $i = 1, \ldots, n_{tr}$. This goal, IW for everything, is too general and then its
only solution is $w = w^*_{\pi}$; however, it is more than needed—IW for classification should be enough.

Specifically, the goal of DIW is to find a set of weights $\mathcal{W} = \{w_i\}_{i=1}^{n_{tr}}$ such that for $\ell(f_{\theta}(x), y)$,
\[
\frac{1}{n_v} \sum_{i=1}^{n_v} \ell(f_{\theta}(x_i^v), y_i^v)\big|_{\theta=\theta_t} \approx \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} w_i \ell(f_{\theta}(x_i^t), y_i^t)\big|_{\theta=\theta_t}
\] (9)
where the left- and right-hand sides are conditioned on $\theta = \theta_t$, and $\theta_t$ holds model parameters at a
certain time point of training. After $\mathcal{W}$ is found, $\theta_t$ will be updated to $\theta_{t+1}$, and the current $f_{\theta}$
will move to the next $f_{\theta}$; then, we need to find a new set of weights satisfying Eq. (9) again. Compared
with the general goal of IW, the goal of DIW is special and easy to achieve, and then there may be
many different solutions, any of which can be used to replace $\mathcal{W}^* = \{w_i^*\}_{i=1}^{n_{tr}}$ in $\hat{R}(f_{\theta})$ in Eq. (3).
The above argument elaborates the motivation of $\pi : (x, y) \mapsto \ell(f_{\theta}(x), y)$. This is possible thanks
to the dynamic nature of weights in DIW which is the major methodological contribution.

**Distribution matching** Finally, we perform distribution matching between the set of transformed
training data $\{x_i^t\}_{i=1}^{n_{tr}}$ and the set of transformed validation data $\{z_i^v\}_{i=1}^{n_{tr}}$. Let $\mathcal{H}$ be a Hilbert space
of real-valued functions on $\mathbb{R}^d$, with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, or $\mathcal{H}$ be a reproducing kernel Hilbert space,
where $k : (z, z') \mapsto \langle \phi(z), \phi(z') \rangle_{\mathcal{H}}$ is the reproducing kernel of $\mathcal{H}$ and $\phi : \mathbb{R}^d \to \mathcal{H}$ is the
kernel-induced feature map (Schölkopf and Smola, 2001). We perform distribution matching by
**kernel mean matching** (Huang et al., 2007).

Let $\mu_{tr} = \mathbb{E}_{p_{tr}(x,y) \cdot w(z)}[\phi(z)]$ and $\mu_{te} = \mathbb{E}_{p_{te}(x,y) \cdot w(z)}[\phi(z)]$ be the kernel embeddings of $p_{tr} \cdot w$ and
$p_{te}$ in $\mathcal{H}$, then the **maximum mean discrepancy** (MMD) (Borgwardt et al., 2006; Gretton et al., 2012)
is defined as
\[
\sup_{\|f\|_{\mathcal{H}} \leq 1} \mathbb{E}_{p_{tr}(x,y) \cdot w(z)}[f(z)] - \mathbb{E}_{p_{te}(x,y) \cdot w(z)}[f(z)] = \|\mu_{tr} - \mu_{te}\|_{\mathcal{H}},
\]
and the squared MMD can be approximated by
\[
\left\| \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} w_i \phi(z_i^t) - \frac{1}{n_v} \sum_{i=1}^{n_v} \phi(z_i^v) \right\|_{\mathcal{H}}^2 = \frac{1}{n_{tr}} w^T K w - \frac{2}{n_{tr}} k^T w + \text{Const.},
\] (10)
where \( w \in \mathbb{R}^{n_{tr}} \) is the weight vector, \( K \in \mathbb{R}^{n_{tr} \times n_{tr}} \) is a kernel matrix such that \( K_{ij} = k(z_i^{tr}, z_j^{tr}) \), and \( k \in \mathbb{R}^{n_{tr}} \) is a vector such that \( k_i = \frac{1}{n_{tr}} \sum_{j=1}^{n_{tr}} k(z_i^{tr}, z_j^{tr}) \). In practice, Eq. (10) is minimized subject to \( 0 \leq w_i \leq B \) and \( \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} w_i - 1 \leq \epsilon \) where \( B > 0 \) and \( \epsilon > 0 \) are hyperparameters as the upper bound of weights and the slack variable of \( \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} w_i = 1 \). Eq. (10) is the objective of distribution matching, and the proposed DIW algorithm is presented in Algorithm 1\(^4\), which is our major algorithmic contribution.

3 Applications

We have proposed DIW for deep learning under distribution shift which is almost everywhere in the wild. Here, we introduce some examples: covariate shift, class imbalance, and label noise.

Covariate shift may be the most popular shift whose definition was given in Eq. (7) (Pan and Yang, 2009; Shimodaira, 2000; Sugiyama et al., 2007b, 2008). It is harmful though not change, since the expressive power of \( f_\theta \) is limited so that it will focus more on the regions where \( p_{te}(x) \) is higher but not where \( p_{te}(x) \) is higher.

Class imbalance may be the simplest shift which is defined by plugging \( p_{tr}(x \mid y) = p_{te}(x \mid y) \) in Eq. (8) (Japkowicz and Stephen, 2002; He and Garcia, 2009; Zhang et al., 2013; Huang et al., 2016; Buda et al., 2018; Lipton et al., 2018). The optimal solution is simply \( w^*(x, y) = p_{te}(y) / p_{tr}(y) \), involving counting instead of density ratio estimation (Sugiyama et al., 2012). It is however quite important—otherwise \( f_\theta \) will emphasize over-represented classes and neglect under-represented classes, which is unacceptable in terms of the transferability or fairness (Cao et al., 2019). It can also serve as a unit test to see if an IW method can successfully recover \( w^*(x, y) \) without being told that the shift is indeed class imbalance.

Label noise may be the hardest or already adversarial shift where \( p_{tr}(x) = p_{te}(x) \) and \( p_{tr}(y \mid x) \neq p_{te}(y \mid x) \) which is opposite to covariate shift. There is a label corruption process \( p(\tilde{y} \mid y, x) \) where \( \tilde{y} \) denotes the corrupted label so that \( p_{tr}(\tilde{y} \mid x) = \sum_y p(\tilde{y} \mid y, x) \cdot p_{te}(y \mid x) \), i.e., a label \( y \) may flip to every corrupted label \( \tilde{y} \neq y \) with a probability \( p(\tilde{y} \mid y, x) \). It is extremely detrimental to training, since an over-parameterized \( f_\theta \) is able to fit any training data even if the training labels are random (Zhang et al., 2017). As a result, label noise could significantly mislead \( f_\theta \) to fit \( p_{tr}(\tilde{y} \mid x) \) that is an improper map from \( x \) to \( y \), and this is much more serious than misleading the attention of \( f_\theta \). Note that DIW can estimate \( p(\tilde{y} \mid y, x) \), since our validation data carry the information about \( p_{te}(y \mid x) \); without those validation data, \( p(\tilde{y} \mid y, x) \) is unidentifiable, and then it is usually assumed to be independent of \( x \) and simplified as \( p(\tilde{y} \mid y) \), i.e., the class-conditional noise (CCN) (Natarajan et al., 2013; Patrini et al., 2017; Liu and Tao, 2016; Han et al., 2018b,a; Yu et al., 2019; Xia et al., 2019). DIW can also be applied to the shift where \( p_{tr}(x \mid \tilde{y}) = \sum_y p(\tilde{y} \mid y) \cdot p_{te}(x \mid y) \) (Scott et al., 2013; du Plessis et al., 2013; Menon et al., 2015; Lu et al., 2019, 2020).

\(^4\)For space reasons, we defer convergence analysis of the proposed algorithm to Appendix A.
4 Discussions

Since distribution shift is ubiquitous in the wild, there are many philosophies different from IW for mitigating its negative effects. In what follows, we discuss some very related philosophies: learning to reweight, distributionally robust supervised learning, and domain adaptation.

Learning to reweight iterates between weighted classification on training data for updating \( f_\theta \), and unweighted classification on validation data for updating \( \mathcal{W} \) (Ren et al., 2018). Although this looks like DIW, its philosophy is fairly different from IW: IW has a specific target \( \mathcal{W}^* \) to estimate, while reweighting has a goal to optimize but no target to estimate; its goal is still empirical risk minimization on very limited validation data, and thus it may overfit the validation data. Technically, \( \mathcal{W} \) is hidden in \( \theta_W \) in the objective of unweighted classification, so that (Ren et al., 2018) had to use a series of approximations just to differentiate the objective w.r.t. \( \theta_W \), which is remarkably more difficult than Eq. (10). This reweighting philosophy can also be used to train another deep network for providing \( \mathcal{W} \) (Jiang et al., 2018).

Distributionally robust supervised learning (DRSL) assumes that there is no validation data drawn from \( p_{te}(x,y) \) or \( p_{te}(x) \), and consequently its philosophy is to consider the worst-case distribution shift within a prespecified uncertainty set (Ben-Tal et al., 2013; Wen et al., 2014; Namkoong and Duchi, 2016, 2017). We can clearly see its sufficient difference from IW: IW regards \( p_{te}(x,y) \) as fixed and \( p_{tr}(x,y) \) as shifted from \( p_{te}(x,y) \) while DRSL regards \( p_{tr}(x,y) \) as fixed and \( p_{te}(x,y) \) as shifted from \( p_{tr}(x,y) \). This worst-case philosophy makes DRSL more sensitive to bad training data (e.g., outliers or noisy labels) which leads to less robust \( f_\theta \) (Hu et al., 2018).

Domain adaptation (DA) is also closely related where \( p_{te}(x,y) \) and \( p_{te}(x) \) are called in-domain and out-of-domain distributions (Daume III and Marcu, 2006) or called target and source domain distributions (Ben-David et al., 2007). Although supervised DA is more similar to DIW, this area focuses more on unsupervised DA (UDA), i.e., the validation data come from \( p_{te}(x) \) rather than \( p_{te}(x,y) \). UDA has at least three major philosophies: transfer knowledge from \( p_{tr}(x) \) to \( p_{te}(x) \) by bounding the domain discrepancy (Ghifary et al., 2017) or finding some domain-invariant representations (Ganin et al., 2016), transfer from \( p_{tr}(x \mid y) \) to \( p_{te}(x \mid y) \) by conditional domain-invariant representations (Gong et al., 2016), and transfer from \( p_{tr}(y \mid x) \) to \( p_{te}(y \mid x) \) by pseudo-labeling target domain data (Saito et al., 2017). They all have their own assumptions such as \( p(y \mid x) \) or \( p(x \mid y) \) cannot change too much, and hence none of them can deal with the label-noise application of IW. Technically, the key difference of UDA from IW is that UDA methods do not weight/reweight source domain data.

5 Experiments

In this section, we verify the effectiveness of the proposed DIW for deep learning. First, we compare DIW (the loss-value transformation version in Algorithm 1) with baselines under two representative distribution shift settings: label noise and class imbalance. Second, we discuss which experimental choices (e.g., SIW/DIW, with/without FE, FE fixed/updated, with/without pretraining) contribute the most to the success of DIW in an extensive ablation study.
Figure 3: Experimental results of training deep neural networks on Fashion-MNIST, CIFAR-10 and CIFAR-100 with label noise. Shaded regions present standard deviation over five repeated trials.

Setup  We perform experiments on Fashion-MNIST (Han Xiao, 2017), CIFAR-10 (Krizhevsky and Hinton, 2009) and CIFAR-100 (Krizhevsky and Hinton, 2009). For the tiny validation set, we use 1,000 clean data for label noise experiments and 10 data per class for class imbalance experiments. Note that the validation set is included in the training set for all baseline methods. For Fashion-MNIST and CIFAR, the models are LeNet-5 (LeCun et al., 1998) and ResNet-32 (He et al., 2016) respectively; the optimizer is stochastic gradient descent (Robbins and Monro, 1951). For a fair comparison, we normalize the weights of all examples in a training batch so that they sum up to one, and we do not employ any form of data augmentation for all the methods. More details about the setups and supplementary experimental results can be found in Appendix B and C.

Baselines  (i) Clean, use only the tiny validation dataset for training; (ii) Uniform, assign the same weights to all the training data; (iii) Random, assign random weights according to a rectified Gaussian distribution \( w_i = \max(s_i, 0) \), where \( s_i \sim \mathcal{N}(0, 1) \); (iv) Reweight, proposed by (Ren et al., 2018); (v) IW, apply IW directly on the original data for assigning weights (Huang et al., 2007).
Figure 4: Weight distribution for correctly and wrongly labeled data on CIFAR-10 0.4 symmetric flip.

5.1 Experimental results

Results on label noise  We start with the most challenging distribution shift setting, label noise. Two representative noise settings are considered here: symmetric flip (Van Rooyen et al., 2015) where each label is independently flipped to another class with a certain probability; pair flip (Han et al., 2018b) where the flipping occurs only within similar classes, and the noise rates are \{0.3, 0.4, 0.5\}. The experimental results are reported in Figure 3. We can see that the proposed method performs better than baselines, especially in the symmetric flip case. Moreover, as the noise level increases, the proposed method is still reasonably robust while others tend to overfit to the noisy labels.

To better understand how DIW contributes to learn more robust models, we take a closer look at the learned weights in the final training epoch. As shown in Figure 4, our method can successfully detect the wrongly labeled data and push them to nearly zero weights, and detect the correctly labeled data and upweight them, while others fail to do so. The results corroborate our analysis that DIW can gradually reduce the bias of DC and \(W\) together and learn robustly under the distribution shift.

Results on class imbalance  Then, we test another common distribution shift setting, class imbalance, where we create a multi-class imbalanced dataset from Fashion-MNIST by step imbalance (Buda et al., 2018): the sample sizes within the majority classes, and within the minority classes are the same. Let \(\mu\) be the fraction of minority classes and \(\rho\) be the ratio between sample sizes of the majority classes and minority classes. We test the following two settings: (i) \(\mu = 0.2, \rho = 100\); (ii) \(\mu = 0.2, \rho = 200\). From the results in Table 1, we can see that the proposed method performs favorably than the baselines.

| Method  | \(\rho = 100\)   | \(\rho = 200\)   |
|---------|------------------|------------------|
| Clean   | 67.77 (0.94)     | 67.77 (0.94)     |
| Uniform | 82.39 (0.94)     | 76.87 (1.14)     |
| Random  | 82.85 (0.76)     | 78.48 (0.79)     |
| IW      | 81.58 (0.79)     | 77.01 (1.95)     |
| Reweight| 81.82 (0.95)     | 76.59 (1.11)     |
| Proposed| **83.69 (1.21)** | **81.38 (1.24)** |

Table 1: Mean accuracy (standard deviation) in percentage on imbalanced Fashion-MNIST (5 trials). Best and comparable methods (paired t-test at significance level 5%) are highlighted in bold.
5.2 Ablation study

Since our proposed DIW comprises different options in algorithmic design as illustrated in Figure 2, here we perform an extensive ablation study to better understand the mechanism and provide a guidance for practical use. The options considered are whether to: (i) introduce FE; (ii) update $\mathcal{W}$; (iii) update FE; (iv) pretrain FE. Starting from the original IW: adding (i) to IW yields SIW; adding (ii) to SIW yields DIW1; adding (iii) to DIW1 yields DIW3; adding (iv) to DIW3 yields DIW2.

We conduct thorough experiments to compare the above methods in the label noise setting. The results are reported in Table 2, where method with "-F"/-"L" suffix means using hidden-layer-output/loss-value transformation in Algorithm 1. Our observations in general are: (i) SIWs outperform IW due to the advantages of introducing FE; (ii) DIWs outperform SIWs since they benefit from updating $\mathcal{W}$ on-the-fly in an end-to-end fashion; (iii) for DIWs with pretrained FE (i.e. DIW1 and DIW2), updating FE during training is usually better than fixing it; (iv) for DIWs with updating FE (i.e. DIW2 and DIW3), "-F" methods perform better when FE is pretrained than randomly initialized, while "-L" methods do not necessarily need a pretrained FE to perform well and thus are more recommended.

We further visualize the last layer representations $h(x) \in \mathbb{R}^{64}$ of learned models on CIFAR-10 with 0.4 symmetric label noise by t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008) in Figure 5. The learned representations of DIWs are in general more concentrated within clusters and therefore easier to be separated for different classes, which shows their superiority over static methods.
Table 2: Mean accuracy (standard deviation) in percentage on Fashion-MNIST (F-MNIST for short), CIFAR-10 and CIFAR-100 with label noise (5 repeated trials). Best and comparable methods (paired t-test at significance level 5%) are highlighted in bold. p/s is short for pair/symmetric flip.

| Data          | Noise | IW       | SIW-F | SIW-L | DIW1-F | DIW2-F | DIW3-F | DIW1-L | DIW2-L | DIW3-L |
|---------------|-------|----------|-------|-------|--------|--------|--------|--------|--------|--------|
|               | 0.3 p | 81.90    | 81.89 | 83.30 | 81.31  | 83.06  | 83.18  | 83.20  | 78.58  | 88.14  |
|               |       | (0.60)   | (0.51)| (0.68)| (1.14) | (0.62) | (0.59) | (1.20) | (3.76) | (0.50) |
| F-MNIST       | 0.4 s | 80.57    | 80.81 | 88.93 | 82.99  | 82.85  | 82.08  | 89.06  | 88.78  | 89.09  |
|               |       | (0.58)   | (0.62)| (0.08)| (0.58) | (0.79) | (0.41) | (0.12) | (0.27) | (0.07) |
|               | 0.5 s | 79.53    | 79.02 | 88.39 | 83.73  | 82.18  | 81.69  | 88.19  | 87.65  | 88.31  |
|               |       | (0.61)   | (0.74)| (0.18)| (0.65) | (0.81) | (0.40) | (0.10) | (0.47) | (0.21) |
|               | 0.3 p | 43.54    | 54.38 | 76.28 | 83.90  | 84.14  | 74.33  | 77.82  | 81.05  | 82.50  |
|               |       | (0.84)   | (0.55)| (0.53)| (0.29) | (0.45) | (1.20) | (0.93) | (0.49) | (0.26) |
| CIFAR-10      | 0.4 s | 43.53    | 43.38 | 71.93 | 81.08  | 72.19  | 73.52  | 78.92  | 79.76  |        |
|               |       | (0.46)   | (0.44)| (0.45)| (0.61) | (0.35) | (1.33) | (0.66) | (0.52) | (0.40) |
|               | 0.5 s | 41.36    | 34.07 | 64.03 | 77.31  | 70.23  | 67.50  | 71.72  | 73.83  |        |
|               |       | (0.84)   | (0.56)| (0.45)| (0.66) | (0.30) | (0.75) | (0.93) | (0.88) | (0.53) |
|               | 0.3 p | 9.24     | 47.49 | 51.27 |        |        |        | 52.96  | 55.94  | 54.03  |
|               |       | (0.26)   | (0.29)| (0.34)|        |        |        | (0.32) | (0.42) | (0.4)  |
| CIFAR-100§    | 0.4 s | 9.07     | 35.40 | 46.74 |        |        |        | 49.27  | 50.70  | 50.53  |
|               |       | (0.19)   | (0.77)| (0.37)|        |        |        | (0.49) | (0.55) | (0.34) |
|               | 0.5 s | 8.97     | 28.19 | 41.10 |        |        |        | 44.54  | 46.63  | 46.62  |
|               |       | (0.30)   | (0.65)| (0.38)|        |        |        | (0.61) | (0.49) | (0.39) |

6 Conclusion

We rethought importance weighting for deep learning under distribution shift and explained that it suffers from a circular dependency conceptually and theoretically. To avoid the issue, we proposed dynamic importance weighting that iterates between deep classifier training and weight estimation, where features for weight estimation can be extracted as either hidden-layer outputs or loss values. Experiments on typical distribution shifts demonstrated the effectiveness of the proposed method.

§Note that “-F” methods for DIW is not applicable on CIFAR-100, since after partitioning the mini-batch data by 100 classes, the data in each partition is too few to conduct weight estimation.
Acknowledgments
We thank Prof. Magnus Boman and Prof. Henrik Boström for constructive suggestions in developing the work, and thank Xuyang Zhao, Tianyi Zhang, Yifan Zhang and Ikko Yamane for helpful discussions. NL was supported by MEXT scholarship No. 171536 and the MSRA D-CORE Program. GN and MS were supported by JST AIP Acceleration Research Grant Number JPMJCR20U3, Japan.

References

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In NeurIPS, 2007.

A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. Management Science, 59(2):341–357, 2013.

K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics, 22(14):e49–e57, 2006.

M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 106:249–259, 2018.

K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In NeurIPS, 2019.

H. Daume III and D. Marcu. Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research, 26:101–126, 2006.

M. C. du Plessis, G. Niu, and M. Sugiyama. Clustering unclustered data: Unsupervised binary labeling of two datasets having different class balances. In TAAI, 2013.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky. Domain-adversarial training of neural networks. Journal of Machine Learning Research, 17(59):1–35, 2016.

M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(7):1414–1430, 2017.

M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf. Domain adaptation with conditional transferable components. In ICML, 2016.

I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. The MIT press, 2016.
A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

B. Han, J. Yao, G. Niu, M. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, 2018a.

B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018b.

R. V. Han Xiao, Kashif Rasul. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747v2*, 2017.

H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *ICML*, 2018.

C. Huang, Y. Li, C. Change Loy, and X. Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016.

J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola. Correcting sample selection bias by unlabeled data. In *NeurIPS*, 2007.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(7):1391–1445, 2009.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

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

Z. C. Lipton, Y.-X. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In *ICML*, 2018.
T. Liu and D. Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2016.

N. Lu, G. Niu, A. K. Menon, and M. Sugiyama. On the minimal supervision for training any binary classifier from only unlabeled data. In *ICLR*, 2019.

N. Lu, T. Zhang, G. Niu, and M. Sugiyama. Mitigating overfitting in supervised classification from two unlabeled datasets: A consistent risk correction approach. In *AISTATS*, 2020.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

A. Menon, B. Van Rooyen, C. S. Ong, and B. Williamson. Learning from corrupted binary labels via class-probability estimation. In *ICML*, 2015.

H. Namkoong and J. C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *NeurIPS*, 2016.

H. Namkoong and J. C. Duchi. Variance-based regularization with convex objectives. In *NeurIPS*, 2017.

N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *NeurIPS*, 2013.

S. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of mathematical statistics*, pages 400–407, 1951.

K. Saito, Y. Ushikku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*, 2017.

B. Schölkopf and A. Smola. *Learning with Kernels*. The MIT Press, 2001.

C. Scott, G. Blanchard, and G. Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *COLT*, 2013.
H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

M. Sugiyama and M. Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. The MIT press, 2012.

M. Sugiyama, M. Krauledat, and K. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5):985–1005, 2007a.

M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NeurIPS*, 2007b.

M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

M. Sugiyama, T. Suzuki, and T. Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

B. Van Rooyen, A. Menon, and R. C. Williamson. Learning with symmetric label noise: The importance of being unhinged. In *NeurIPS*, 2015.

J. Wen, C.-N. Yu, and R. Greiner. Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In *ICML*, 2014.

X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, 2019.

X. Yu, B. Han, J. Yao, G. Niu, I. W. Tsang, and M. Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, 2019.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *ICML*, 2013.
A Convergence analysis

Let $R_{te}(\theta) = \mathbb{E}_{p_{te}(x,y)}[\ell(f_{\theta}(x), y)]$ be the classification risk which is the objective we would like to optimize, and $R_{tr}(\theta, w) = \mathbb{E}_{p_{tr}(x,y)}[w(z)\ell(f_{\theta}(x), y)]$ be the objective of our weighted classification, where $w(z) = \frac{p_{te}(z)}{p_{tr}(z)}$. In what follows, we theoretically show that our method converges to a critical point of $R_{te}(\theta)$ under mild conditions, and we also give its convergence rate. Before presenting the analysis, we list required assumptions.

**Assumption 1** (Lipschitz continuous gradient). *The learning objective $R_{tr}(\theta, w)$ is twice differentiable and has an $L$-Lipschitz continuous gradient for all $w$, i.e.,

$$-LI \preceq \nabla^2_{\theta} R_{tr}(\theta, w) \preceq LI,$$

where $I$ is the identity matrix.

**Assumption 2** (Bounded variance of noise). *We consider the general stochastic gradient descent scenario. Denote the stochastic gradient by $\tilde{\nabla}_{\theta} R_{tr}(\theta, w)$, and we assume it satisfies:

$$\mathbb{E}[\tilde{\nabla}_{\theta} R_{tr}(\theta, w)] = \mathbb{E}[\nabla_{\theta} R_{tr}(\theta, w)], \quad \mathbb{E}[\mathbb{E}_{\theta} \left( \| \tilde{\nabla}_{\theta} R_{tr}(\theta, w) - \nabla_{\theta} R_{tr}(\theta, w) \|^{2} \right)] \leq \sigma^{2}$$

for some constant $\sigma^{2}$ and all $w$.

**Assumption 3** (Sensitivity). *For any $\theta$ and $\theta'$, we assume the sensitivity of weights with respect to the model parameters satisfy

$$\mathbb{E}_{p_{tr}(x,y)}|w' - w| \leq B\|\theta' - \theta\|^{2},$$

where $w = \frac{p_{te}(\pi_{\theta}(x,y))}{p_{tr}(\pi_{\theta}(x,y))}$ and $w' = \frac{p_{te}(\pi_{\theta'}(x,y))}{p_{tr}(\pi_{\theta'}(x,y))}$ in Theorem 1.

Next, we show the main convergence result.

**Theorem 2.** *Suppose the learning objective $R_{tr}(\theta, w)$, learned weights $w$ and model parameter $\theta$ satisfy the aforementioned assumptions. Let $\ell$ be a bounded loss such that $\ell < M$ and $M > 0$, $T$ be the number of training epochs, and the learning rate $\alpha_t$ satisfies $\alpha_t = \frac{c}{\sqrt{t}}$, where $c$ is a constant and $t \in [T]$. Then, our proposed method given by Algorithm 1 achieves $\mathbb{E}[\| \nabla R_{tr}(\theta_T, w) \|^{2}] \leq \epsilon$ in $O(1/\epsilon^2)$ steps. More specifically, the uniformly randomized output satisfies

$$\mathbb{E}[\| \nabla R_{tr}(\theta_T, w_T) \|^{2}] \leq \frac{\Delta}{\sqrt{T}},$$

(11)

where $\Delta = \frac{2M}{c} + 2cL\sigma^{2} + 4cMB\sigma^{2}$ is a constant independent of the convergence process.
Proof. The update rule for our proposed method given by Algorithm 1 in the population version is as follows:

$$\theta_{t+1} = \theta_t - \alpha_t \nabla_{\theta} R_{te}(\theta_t, w_t).$$

(12)

Given Eq. (1) and Theorem 1, we have

$$R_{te}(\theta)\big|_{\theta=\theta_t} = \mathbb{E}_{p_t(x,y)}[\ell(f_{\theta_t}(x), y)] = \mathbb{E}_{p_t(x,y)} \left[ w(x) \ell(f_{\theta_t}(x), y) \right] = R_{tr}(\theta, w)\big|_{\theta=\theta_t, w=w_t}.\quad (13)$$

Then, the objective at next time step will be

$$R_{te}(\theta_{t+1}) = R_{tr}(\theta_{t+1}, w_{t+1})$$

$$= R_{tr}(\theta_{t+1}, w_t) + R_{tr}(\theta_{t+1}, w_{t+1}) - R_{tr}(\theta_{t+1}, w_t).\quad (14)$$

By Taylor’s theorem and Assumption 1, there exists \( \theta' \) such that

$$R_{tr}(\theta_{t+1}, w_t) = R_{tr} \left( \theta_t - \alpha_t \nabla_{\theta} w_t \right)$$

$$= R_{tr}(\theta_t, w_t) - \alpha_t \nabla_{\theta} R_{tr}(\theta_t, w_t) + \frac{\alpha_t^2}{2} \nabla_{\theta}^2 R_{tr}(\theta', w_t) \nabla_{\theta}$$

$$\leq R_{tr}(\theta_t, w_t) - \alpha_t \nabla_{\theta} R_{tr}(\theta_t, w_t) + \frac{\alpha_t^2 L}{2} \| \nabla_{\theta} \|^2.\quad (15)$$

Given the bounded loss function \( \ell \) and Assumption 3, we have

$$R_{tr}(\theta_{t+1}, w_{t+1}) - R_{tr}(\theta_{t+1}, w_t) = \mathbb{E}_{p_t(x,y)} \left[ w_{t+1} \ell(f_{\theta_{t+1}}(x), y) - w_t \ell(f_{\theta_{t+1}}(x), y) \right]$$

$$= \mathbb{E}_{p_t(x,y)} \left[ (w_{t+1} - w_t) \ell(f_{\theta_{t+1}}(x), y) \right]$$

$$\leq M \mathbb{E}_{p_t(x,y)} |w_{t+1} - w_t|$$

$$\leq MB \| \theta_{t+1} - \theta_t \|^2$$

$$= MB \alpha_t^2 \| \nabla_{\theta} \|^2.\quad (16)$$

Then, we obtain

$$R_{te}(\theta_{t+1}) \leq R_{tr}(\theta_t, w_t) - \alpha_t \nabla_{\theta} R_{tr}(\theta_t, w_t) + \frac{\alpha_t^2 L}{2} + MB \alpha_t^2 \| \nabla_{\theta} \|^2,\quad (17)$$

where \( \nabla_{\theta} \) denotes \( \nabla_{\theta} R_{tr}(\theta_t, w_t) \). Taking the expected value gives us

$$\mathbb{E}[R_{te}(\theta_{t+1})|\theta_t] \leq R_{tr}(\theta_t, w_t) - \alpha_t \mathbb{E} \left[ \nabla_{\theta} R_{tr}(\theta_t, w_t) \right] + \left( \frac{\alpha_t^2 L}{2} + MB \alpha_t^2 \right) \mathbb{E} \left[ \| \nabla_{\theta} \|^2 | \theta_t \right]$$

$$\leq R_{tr}(\theta_t, w_t) - \alpha_t \| \nabla_{\theta} \|^2 + \left( \frac{\alpha_t^2 L}{2} + MB \alpha_t^2 \right) \mathbb{E} \left[ \| \nabla_{\theta} \|^2 | \theta_t \right].\quad (18)$$

18
Since
\[
\mathbb{E} \left[ \left\| \tilde{\nabla}_t \right\|^2 \mid \theta_t \right] = \mathbb{E} \left[ \left\| \tilde{\nabla}_t - \nabla_t + \nabla_t \right\|^2 \mid \theta_t \right] \\
= \mathbb{E} \left[ \left\| \tilde{\nabla}_t - \nabla_t \right\|^2 + \left\| \nabla_t \right\|^2 + 2 \left( \tilde{\nabla}_t - \nabla_t \right)^T \nabla_t \mid \theta_t \right] \\
\leq \sigma^2 + \left\| \nabla_t \right\|^2,
\]
where the last inequality Eq. (19) comes from Assumption 2, then we have
\[
\mathbb{E}[R_{te}(\theta_{t+1}) \mid \theta_t] \leq R_{tr}(\theta_t, w_t) - \left( \alpha_t - \frac{\alpha_t^2 L}{2} - MB\alpha_t^2 \right) \left\| \nabla_t \right\|^2 + \left( \frac{\alpha_t^2 L}{2} + MB\alpha_t^2 \right) \sigma^2.
\]
If we set \( \alpha_t \) small enough such that \( \alpha_t < \frac{1}{L + 2MB} \) for all \( t \), then
\[
\mathbb{E}[R_{te}(\theta_{t+1}) \mid \theta_t] \leq R_{tr}(\theta_t, w_t) - \frac{\alpha_t}{2} \left\| \nabla_t \right\|^2 + \left( \frac{\alpha_t^2 L}{2} + MB\alpha_t^2 \right) \sigma^2.
\]
Now taking the full expectation,
\[
\mathbb{E}[R_{te}(\theta_{t+1})] \leq \mathbb{E}[R_{tr}(\theta_t, w_t)] - \frac{\alpha_t}{2} \mathbb{E} \left[ \left\| \nabla_t \right\|^2 \right] + \left( \frac{\alpha_t^2 L}{2} + MB\alpha_t^2 \right) \sigma^2,
\]
and then rearranging the terms,
\[
\frac{1}{2} \mathbb{E} \left[ \left\| \nabla_t \right\|^2 \right] \leq \frac{\mathbb{E}[R_{tr}(\theta_t, w_t)] - \mathbb{E}[R_{te}(\theta_{t+1})]}{\alpha_t} + \left( \frac{\alpha_t L}{2} + MB\alpha_t \right) \sigma^2.
\]
Next summing up Eq. (23) from \( t = 1 \) to \( T \),
\[
\frac{1}{2} \sum_{t=1}^{T} \mathbb{E} \left[ \left\| \nabla_t \right\|^2 \right] \leq \sum_{t=1}^{T} \frac{\mathbb{E}[R_{tr}(\theta_t, w_t)]}{\alpha_t} - \sum_{t=1}^{T} \frac{\mathbb{E}[R_{te}(\theta_{t+1})]}{\alpha_t} + \left( \frac{L}{2} + MB \right) \sigma^2 \sum_{t=1}^{T} \alpha_t \\
= \sum_{t=1}^{T} \frac{\mathbb{E}[R_{te}(\theta_t)]}{\alpha_t} - \sum_{t=2}^{T+1} \frac{\mathbb{E}[R_{te}(\theta_t)]}{\alpha_{t-1}} + \left( \frac{L}{2} + MB \right) \sigma^2 \sum_{t=1}^{T} \alpha_t \\
= \sum_{t=2}^{T} \left( \frac{1}{\alpha_t} - \frac{1}{\alpha_{t-1}} \right) \mathbb{E}[R_{te}(\theta_t)] + \frac{\mathbb{E}[R_{te}(\theta_1)]}{\alpha_1} - \frac{\mathbb{E}[R_{te}(\theta_{T+1})]}{\alpha_T} + \left( \frac{L}{2} + MB \right) \sigma^2 \sum_{t=1}^{T} \alpha_t \\
\leq \sum_{t=2}^{T} \left( \frac{1}{\alpha_t} - \frac{1}{\alpha_{t-1}} \right) M + \frac{M}{\alpha_1} + \left( \frac{L}{2} + MB \right) \sigma^2 \sum_{t=1}^{T} \alpha_t \\
= \frac{M}{\alpha_T} + \left( \frac{L}{2} + MB \right) \sigma^2 \sum_{t=1}^{T} \alpha_t,
\]
and then setting $\alpha_t = \frac{c}{\sqrt{t}}$ (with $\frac{1}{\alpha_0} \triangleq 0$),
\[
\frac{1}{2} \sum_{t=1}^{T} \mathbb{E} [\|\nabla_t\|^2] \leq \frac{M\sqrt{T}}{c} + \left( \frac{L}{2} \sigma^2 + MB\sigma^2 \right) 2c\sqrt{T} \\
= \left( \frac{M}{c} + cL\sigma^2 + 2cMB\sigma^2 \right) \sqrt{T},
\]
(25)
where the last inequality follows since $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\sqrt{T}$.

Therefore, let $m_T = \theta_t$ with probability $\frac{1}{T}$ for all $t \in \{0, \ldots, T-1\}$, and the expected value of gradient at this point is
\[
\mathbb{E}[\|\nabla R_t(m_T, w_t)\|^2] = \sum_{t=1}^{T} \mathbb{E} [\|\nabla_t\|^2] \cdot \mathbb{P} (m_T = \theta_t) \\
= \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} [\|\nabla_t\|^2].
\]
(26)
Substituting this back into Eq. (25) gives us
\[
\mathbb{E}[\|\nabla R_t(m_T, w_t)\|^2] \leq \frac{2}{\sqrt{T}} \left( \frac{M}{c} + cL\sigma^2 + 2cMB\sigma^2 \right),
\]
(27)
which concludes the proof. \hfill \Box

\section*{B Supplementary information on experimental setup}

\subsection*{B.1 Datasets and base models}

**Fashion-MNIST**  Fashion-MNIST Han Xiao (2017) is a 28*28 grayscale image dataset of fashion items in 10 classes. It contains 60,000 training images and 10,000 test images. See https://github.com/zalandoresearch/fashion-mnist for details.

The model for Fashion-MNIST is a LeNet-5 LeCun et al. (1998):

- 0th (input) layer: (32*32)-
- 1st to 2nd layer: C(5*5,6)-S(2*2)-
- 3rd to 4th layer: C(5*5,16)-S(2*2)-
- 5th layer: FC(120)-
- 6th layer: FC(84)-10

where C(5*5,6) means 6 channels of 5*5 convolutions followed by ReLU, S(2*2) means max-pooling layer with a filter size 2*2 and a stride of 2, FC(120) means a fully connected layer with 120 outputs, etc.
CIFAR-10 and CIFAR-100  CIFAR-10 Krizhevsky and Hinton (2009) dataset is a collection of 60,000 real-world object images in 10 classes, 50,000 images for training and 10,000 for testing. Each class has 6,000 32x32 RGB images. CIFAR-100 Krizhevsky and Hinton (2009) is the same dataset as CIFAR-10 but has a total number of 100 classes with 600 images in each class. See https://www.cs.toronto.edu/~kriz/cifar.html for details.

ResNet-32 He et al. (2016) is used as the base model for CIFAR-10 and CIFAR-100:
0th (input) layer: (32*32*3)-
1st to 11th layers: C(3*3, 16)-[C(3*3, 16), C(3*3, 16)]*5-
12th to 21st layers: [C(3*3, 32), C(3*3, 32)]*5-
22nd to 31st layers: [C(3*3, 64), C(3*3, 64)]*5-
32nd layer: Global Average Pooling-10/100
where the input is a 32*32 RGB image, [·, ·] means a building block (He et al., 2016) and [·]*2 means 2 such layers, etc. Batch normalization (Ioffe and Szegedy, 2015) is applied after the 1st layer.

B.2 Label noise experiments
In this work, the noisy labels are generated according to a predefined label noise transition matrix $T$, where $T_{ij} = P(\bar{y} = j | y = i)$. Two types of label noise transition matrices are defined in Figure 6, where $\eta$ is the label noise rate and $n$ is the total number of classes. In pair flip case, the labels in every class only flip to one neighbor class with a probability $\eta$. In symmetric flip label noise, the labels can randomly flip to other $n-1$ classes with equal probability $\eta/n-1$. Note that the label noise transition matrix and label noise rate are unknown to the model.

Figure 6: Label noise transition matrix. Left: Pair flip label noise; Right: Symmetric flip label noise.

B.3 Class imbalance experiments
To create a class imbalance version from Fashion-MNIST, we randomly select 10 data per class for validation set, 4,000 data (including the 10 validation data) per majority class for training set. The number of data per minority class (including the 10 validation data) in training set are computed according to $\rho$ as described in Sec. 5.1. We also randomly select 1,000 data from test set for the test set used in class imbalance experiments.
Table 3: Mean accuracy (standard deviation) in percentage on Fashion-MNIST (F-MNIST for short), CIFAR-10 and CIFAR-100 with label noise corresponding to Figure 3 (5 repeated trials). Best and comparable methods (paired t-test at significance level 5%) are highlighted in bold. p/s is short for pair/symmetric flip.

| Dataset   | Noise | Clean       | Uniform   | Random     | IW         | Reweight   | Proposed   |
|-----------|-------|-------------|-----------|------------|------------|------------|------------|
| F-MNIST   | 0.3 p | 79.11 (0.82)| 71.91 (1.04)| 77.98 (0.99)| 81.90 (0.60)| 86.92 (0.55)| **88.14 (0.50)** |
|           | 0.4 s | 79.11 (0.82)| 80.19 (1.21)| 84.82 (0.90)| 80.57 (0.58)| 80.70 (0.97)| **89.09 (0.07)** |
|           | 0.5 s | 79.11 (0.82)| 77.90 (1.05)| 83.30 (0.83)| 79.53 (0.61)| 77.81 (0.50)| **88.31 (0.21)** |
| CIFAR-10  | 0.3 p | 41.77 (0.78)| 68.40 (0.77)| 79.65 (0.66)| 43.54 (0.84)| 80.26 (0.28)| **82.50 (0.26)** |
|           | 0.4 s | 41.77 (0.78)| 60.70 (0.66)| 68.68 (1.27)| 43.53 (0.46)| 68.06 (0.78)| **79.76 (0.40)** |
|           | 0.5 s | 41.77 (0.78)| 52.27 (1.12)| 61.94 (1.14)| 41.36 (0.84)| 62.85 (0.39)| **73.83 (0.53)** |
| CIFAR-100 | 0.3 p | 10.32 (0.19)| 52.02 (0.54)| 53.00 (0.36)| 9.24 (0.26) | 48.20 (0.52)| **54.03 (0.40)** |
|           | 0.4 s | 10.32 (0.19)| 40.76 (0.51)| 41.82 (0.59)| 9.07 (0.19) | 37.35 (0.98)| **50.53 (0.34)** |
|           | 0.5 s | 10.32 (0.19)| 34.11 (0.41)| 33.42 (0.91)| 8.97 (0.30) | 29.67 (0.94)| **46.62 (0.39)** |

C Supplementary experimental results

Summary of classification accuracy Table 3 presents the mean accuracy and standard deviation on Fashion-MNIST, CIFAR-10 and CIFAR-100 with label noise. This table corresponds to Figure 3 in Sec. 5.1.

Importance weight distribution Figure 7 presents weight distribution for correctly and wrongly labeled data on CIFAR-10 under two label noise settings: 0.3 pair flip and 0.5 symmetric flip. We can see the results here are consistent with that in Figure 4.

Experimental results on ablation study Here we provide supplementary results on ablation study. Figure 8 shows experimental results of methods discussed in in Sec. 5.2, which corresponds to Table 2. Figure 9 is given to present the t-SNE visualization of embeddings, where colors denote noisy labels actually used in training rather than ground-truth labels. By comparing Figure 9 with Figure 5, we can see how the label noise effect is mitigated in classification.
Figure 7: Weight distribution for correctly and wrongly labeled data on CIFAR-10.
Figure 8: Experimental results of training deep neural networks on Fashion-MNIST, CIFAR-10 and CIFAR-100 with label noise.
Figure 9: t-SNE visualization of embeddings for CIFAR-10. Colors denote noisy labels.