Weighted Distillation with Unlabeled Examples

Fotis Iliopoulos, Vasilis Kontonis, Cenk Baykal, Khoa Trinh, Gaurav Menghani, Erik Vee
Google Research
[fotisi, kontonis, baykalc, khoatrinh, gmenghani, erikvee]@google.com

Abstract

Distillation with unlabeled examples is a popular and powerful method for training deep neural networks in settings where the amount of labeled data is limited: A large "teacher" neural network is trained on the labeled data available, and then it is used to generate labels on an unlabeled dataset (typically much larger in size). These labels are then utilized to train the smaller "student" model which will actually be deployed. Naturally, the success of the approach depends on the quality of the teacher's labels, since the student could be confused if trained on inaccurate data. This paper proposes a principled approach for addressing this issue based on a "debiasing" reweighting of the student's loss function tailored to the distillation training paradigm. Our method is hyper-parameter free, data-agnostic, and simple to implement. We demonstrate significant improvements on popular academic datasets and we accompany our results with a theoretical analysis which rigorously justifies the performance of our method in certain settings.

1 Introduction

In many modern applications of deep neural networks, where the amount of labeled examples for training is limited, *distillation with unlabeled examples* [8, 20] has been enormously successful. In this two-stage training paradigm a larger and more sophisticated "teacher model" — typically non-deployable for the purposes of the application — is trained to learn from the limited amount of available training data in the first stage. In the second stage, the teacher model is used to generate labels on an unlabeled dataset, which is usually much larger in size than the original dataset the teacher itself was trained on. These labels are then utilized to train the "student model", namely the model which will actually be deployed. Notably, distillation with unlabeled examples is the most commonly used training-paradigm in applications where one finetunes and distills from very large-scale foundational models such as BERT [13] and GPT-3 [5] and, additionally, it can be used to significantly improve distillation on supervised examples only (see e.g. [47]).

While this has proven to be a very powerful approach in practice, its success depends on the quality of labels provided by the teacher model. Indeed, often times the teacher model generates inaccurate labels on a non-negligible portion of the unlabeled dataset, confusing the student. As deep neural networks are susceptible to overfitting to corrupted labels [50], training the student on the teacher's noisy labels can lead to degradation in its generalization performance. As an example, Figure 1 depicts an instance based on CIFAR-10 where filtering out the teacher's *noisy* labels is quite beneficial for the student's performance.

We address this shortcoming by introducing a natural "noise model" for the teacher which allows us to modify the student's loss function in a principled way in order to produce an unbiased estimate of the student's clean objective. From a practical standpoint, this produces a fully "plug-and-play" method which adds minimal implementation overhead, and which is composable with essentially every other known distillation technique.

The idea of compressing a teacher model into a smaller student model by matching the predictions of the teacher was initially introduced by Buciluǎ, Caruana and Niculescu-Mizil [6] and, since

then, variations of this method [20, 26, 31, 35, 39, 42, 49] have been applied in a wide variety of contexts [37, 48, 51]. (Notably, some of these applications go beyond compression — see e.g. [8, 47, 48, 52] for reference.) In the simplest form of the method [26], and using classification as a canonical example, the labels produced by the teacher are one-hot vectors that represent the class which has the maximum predicted probability — this method is often referred to as "hard"-distillation. More generally, Hinton et. al. [8, 20] have shown that it is often beneficial to train the student so that it minimizes the cross-entropy (or KL-divergence) with the probability distribution produced by the teacher while also potentially using a temperature higher than 1 in the softmax of both models ("soft"-distillation). (Temperatures higher than 1 are used in order to emphasize the difference between the probabilities of the classes with lower likelihood of corresponding to the correct label according to the teacher model.)

The main contribution of this work is a principled method for improving distillation with unlabeled examples by reweighting the loss function of the student. That is, we assign importance weights to the examples labeled by the teacher so that each weight reflects (i) how likely it is that the teacher has made an inaccurate prediction regarding the label of the example and (ii) how "distorted" the unweighted loss function we use to train the student is (measured with respect to using the ground-truth label for the example instead of the teacher's label). More concretely, our reweighting strategy is based on introducing and analyzing a certain noise model designed to capture the behavior of the teacher in distillation. In this setting, we are able to come up with a closed-form solution for weights that "de-noise" the objective in the sense that (on expectation) they simulate having access to clean labels. Crucially, we empirically observe that the key characteristics of our noise model for the teacher can be effectively estimated through a small validation dataset, since in practice the teacher's noise is neither random nor adversarial, and it typically correlates well with its "confidence" — see e.g. Figure 1. In particular, we use the validation dataset to learn a map that takes as input the teacher's and student's confidence for the label of a certain example and outputs estimates for the quantities mentioned in items (i) and (ii) above. Finally, we plug in these estimates to our closed-form solution for the weights so that, overall, we obtain an automated way of computing the student's reweighted objective in practice. A detailed description of our method can be found in Section 2.3.

Our main findings and contributions can be summarized as follows:

- We propose a *principled and hyperparameter-free* reweighting method for knowledge distillation with unlabeled examples. The method is efficient, data-agnostic and simple to implement.
- We present extensive experimental results which show that our method provides significant improvements when evaluated in standard benchmarks.
- Our reweighting technique comes with provable guarantees: (i) it is information-theoretically optimal; and (ii) under standard assumptions SGD optimization of the reweighted objective learns a solution with nearly optimal generalization.

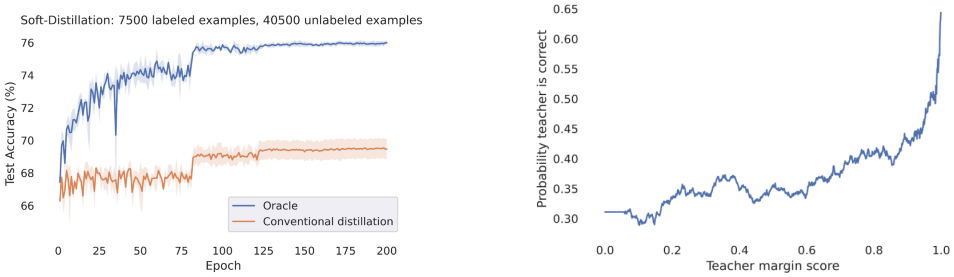


Figure 1: **Left:** Performance comparison between a student trained using conventional distillation with unlabeled examples (orange) and a student trained only on the examples which are labeled *correctly* by the teacher (blue). Here we assume access to 7500 labeled examples and 40500 unlabeled examples of CIFAR-10. The teacher is a MobileNet of depth multiplier 2, while the student is a MobileNet of depth multiplier 1. **Right:** Plot of teacher's accuracy as a function of the margin score.

1.1 Related work

Fidelity vs Generalization in knowledge distillation. Conceptually, our work is related to the paper of Stanton et al. [43] where the main message is that "good student accuracy does not imply good distillation fidelity", i.e., that more closely matching the teacher does not necessarily lead to better student generalization. In particular, [43] demonstrates that when it comes to enlarging the distillation dataset beyond the teacher training data, there is a trade-off between optimization complexity and distillation data quality. Our work can be seen as a principled way of improving this trade-off.

Advanced distillation techniques. Since the original paper of Hinton et. al. [20], there have been several follow-up works [2, 7, 31, 45] which develop advanced distillation techniques which aim to enforce greater consistency between the teacher and the student (typically in the context of distillation on labeled examples). These methods enforce consistency not only between the teacher's predictions and student's predictions, but also between the representations learned by the two models. However, in the context of distillation with unlabeled examples, forcing the student to match the teacher's inaccurate predictions is still harmful, and therefore weighting the corresponding loss functions via our method is still applicable and beneficial. We demonstrate this fact by considering instances of the Variational Information Distillation for Knowledge Transfer (VID) [2] framework, and showing how it is indeed beneficial to combine it with our method in Section 3.1.3. We also show that our approach provides benefits on top of any improvement obtained via temperature-scaling in Appendix B.2.

Learning with noisy data techniques. As we have already discussed, the main conceptual contribution of our work is viewing the teacher model in distillation with unlabeled examples as the source of stochastic noise with certain characteristics. Naturally, one may wonder what is the relationship between our method and works from the vast literature of learning with noisy data (e.g. [4, 14, 16, 23, 25, 27, 29, 33, 36, 38]). The answer is that our method *does not attempt to solve the generic "learning with noisy data"* problem, which is a highly non-trivial task in the absence of assumptions (both in theory and in practice). Instead, our contribution is to observe and exploit the fact that the noise introduced by the teacher in distillation has structure, as it correlates with several metrics of confidence such as the margin-score, the entropy of the predictions etc. We use and quantify this empirical observation to our advantage in order to formulate a principled method for developing a debiasing reweighting scheme (an approach inspired by the "learning with noisy data"-literature) which comes with theoretical guarantees. An additional difference is that works from the learning with noisy data literature typically assume that the training dataset consists of corrupted *hard* labels, and often times their proposed method is not compatible with *soft*-distillation (in the sense that they degrade its performance, making it much less effective) — see e.g. [32] for a study of this phenomenon in the case of label smoothing [28, 44].

Uncertainty-based weighting schemes. Related to our method are approaches for semi-supervised learning where examples are downweighted or filtered out when the teacher is "uncertain." (A plethora of ways for defining and measuring "uncertainty" have been proposed in the literature, e.g. entropy, margin-score, dropout variance etc.) These methods are independent of the student model (they only depend on the teacher model), and so they cannot hope to remove the bias from the student's loss function. In fact, these methods can be viewed as preprocessing steps that can be combined with the student-training process we propose. To demonstrate this we both combine and compare our method to the fidelity-based weighting scheme of [12] in Section 3.3.

1.2 Organization of the paper.

In Section 2, we present our method in detail, while in Section 3 we present our key experimental results. In Section 4, we discuss the theoretical aspects of our work. In Section 5, we summarize our results, we discuss the benefits and limitations of our method, and future work. Finally, we present extended experimental and theoretical results in the Appendix.

2 Weighted distillation with unlabeled examples

In this section we present our method. In Section 2.1, we review multiclass classification. In Section 2.2, we introduce the noise model for the teacher in distillation, which motivates our approach. And finally, in Section 2.3, we describe our method in detail.

2.1 Multiclass classification

In multiclass classification, we are given a training sample $S = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$ drawn from $\mathbb{P}^n$, where $\mathbb{P}$ is an unknown distribution over instances $\mathcal{X}$ and labels $\mathcal{Y} = [L] = \{1, 2, \dots, L\}$. Our goal is to learn a predictor $f : \mathcal{X} \to \mathbb{R}^L$, namely to minimize the *risk* of f. The latter is defined as the expected loss of f:

$$R(f) = \mathbb{E}[\ell(y, f(x))] \tag{1}$$

where (x, y) is drawn from $\mathbb{P}$, and $\ell : [L] \times \mathbb{R}^L \to \mathbb{R}_+$ is a loss function such that, for a label $y \in [L]$ and prediction vector $f(x) \in \mathbb{R}^L$, $\ell(y, f(x))$ is the loss incurred for predicting $f(x)$ when the true label is y. The most common way to approximate the risk of a predictor f is via the so-called *empirical risk*:

$$R_S(f) = \frac{1}{|S|} \sum_{(x,y)\in S} \ell(y, f(x)). \tag{2}$$

That is, given a hypothesis class of predictors $\mathcal{F}$, our goal is typically to find $\min_{f\in\mathcal{F}} R_S(f)$ as a way to estimate $\min_{f\in\mathcal{F}} R(f)$.

2.2 Debiasing weights

As we have already discussed, the main drawback of distillation with unlabeled examples is essentially that the empirical risk minimizer corresponding to the dataset labeled by the teacher cannot be trusted, since the teacher may generate inaccurate labels. To quantify this phenomenon and guide our algorithmic design, we consider the following simple but natural noise model for the teacher.

Let $\mathbb{X}$ be an unknown distribution over instances $\mathcal{X}$. We assume the existence of a *ground-truth classifier* so that each $x \in \mathcal{X}$ is associated with a ground-truth label $f_{\text{true}}(x) \in [L] = \{1, 2, \dots, L\}$. In other words, a clean labeled example is of the form $(x, f_{\text{true}}(x)) \sim \mathbb{P}$ (and $x \sim \mathbb{X}$). Additionally, we consider a stochastic adversary that given an instance $x \in \mathcal{X}$, outputs a "corrupted" label $y_{\text{adv}}(x)$ with probability $p(x)$, and the ground-truth label $f_{\text{true}}(x)$ with probability $1 - p(x)$. Let $\mathbb{D}$ denote the induced adversarial distribution over instances and labels.

It is not hard to see that the empirical risk with respect to a predictor f and sample from $\mathbb{D}$ is *not* an unbiased estimator of the risk

$$R(f) = \mathbb{E}_{x\sim\mathbb{X}}[\ell(f_{\text{true}}(x), f(x))] \tag{3}$$

— see Proposition 2.1. On the other hand, the following *weighted* empirical risk (5) is indeed an unbiased estimator of $R(f)$. For each $x \in \mathcal{X}$ let

$$w_f(x) = \frac{1}{1 + p(x)\,(\text{distortion}_f(x) - 1)}, \quad \text{where} \quad \text{distortion}_f(x) = \frac{\ell(y_{\text{adv}}(x), f(x))}{\ell(f_{\text{true}}(x), f(x))}, \tag{4}$$

and define

$$R^w_S(f) = \frac{1}{|S|} \sum_{(x,y)\in S} w_f(x)\,\ell(y, f(x)), \tag{5}$$

where $S = \{(x_i, y_i)\}_{i=1}^n \sim \mathbb{D}^n$. Observe that the weight for each instance x depends on (i) how likely it is the adversary corrupts its label; and on (ii) how this corrupted label "distorts" the loss we observe at instance x. In the following proposition we establish that the standard (unweighted) empirical risk with respect to distribution $\mathbb{D}$ and a predictor f is a biased estimator of the risk of f under the clean distribution $\mathbb{P}$, while the weighted empirical risk (5) is an unbiased one.

Proposition 2.1 (Debiasing Weights)**.** *Let $S = \{(x_i, y_i)\}_{i=1}^n \sim \mathbb{D}^n$ be a sample from the adversarial distribution. Defining* $\text{Bias}(f) = \mathbb{E}_{x\sim\mathbb{X}}\left[p(x) \cdot (\text{distortion}_f(x) - 1) \cdot \ell(f_{\text{true}}(x), f(x))\right]$ *we have:*

$$\text{“reweighted”} \quad \mathbb{E}[R^w_S(f)] = R(f) \qquad \text{vs} \qquad \text{“standard”} \quad \mathbb{E}[R_S(f)] = R(f) + \text{Bias}(f)\,.$$

Notice that, as expected, the bias of the unweighted predictor is a function of the "power" of the adversary, i.e., a function of how often they can corrupt the label of an instance, and the "distortion" this corruption causes to the loss we observe. The proof of Proposition 2.1 follows from simple, direct calculations and it can be found in Appendix D.3.

Intuitively, given a sufficiently large sample $S \sim \mathbb{D}^n$, optimizing an unbiased estimator for the risk should provide a better approximation for $\min_{f\in F} R(f)$ compared to optimizing an estimator with constant bias. We formalize this intuition in Section 4 and in Appendix D.

2.3 Our method

We consider the standard setting for distillation with unlabeled examples where we are given a dataset $S_\ell = \{(x_i, y_i)\}_{i=1}^m$ of m labeled examples from an unknown distribution $\mathbb{P}$, and a dataset $S_u = \{x_i\}_{i=m+1}^{m+n}$ of n unlabeled examples — typically, $n \geq m$. We also assume the existence of a (small) clean validation dataset $S_v = \{(x_i, y_i)\}_{i=m+n+1}^{i=m+n+q}$ of size q. Finally, let $\ell : \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}_+$ be a loss function that takes as input two vectors over the set of labels $[L]$. We describe our method below and more formally in Algorithm 1 in Appendix A.

Remark 2.1. *The only assumption we need to make about the validation set S_v is that it is not in the train set of the teacher model. That is, set S_v can be used in the train set of the student model if needed — we chose to present S_v as a completely independent hold out set to make our presentation as conceptually clear as possible.*

Training the teacher. The teacher model is trained on dataset S_ℓ, and then it is used to generate labels for the instances in S_u. The labels can be one-hot vectors or probability distributions on $[L]$, depending on whether we apply "hard" or "soft" distillation, respectively.

Training the student. We start by pretraining the student model on dataset S_ℓ. Then, the idea is to think of the teacher model as the source of noise in the setting of Section 2.2, compute a weight $w_f(x)$ for each example x based on (4), and finally train the student on the union of labeled and teacher-labeled examples by minimizing the weighted empirical risk (examples from S_ℓ are assigned unit weight).

We point out two remarks. First, in order to apply (4) to compute the weight of an example x, we need to have estimates of $p(x)$ and $\mathrm{distortion}_f(x)$. To obtain these estimates we use the validation dataset S_v as we describe in the next paragraph. Second, observe that, according to (4), the weight of an example is a function of the predictor, namely the parameters of the model in the case of neural networks. This means that ideally we should be updating our weights assignment every time the parameters of the student model get updated during training. However, we empirically observe that even estimating the weight assignment *only once* during the whole training process (right after training the student model on S_ℓ) suffices for our purposes, and so our method adds minimal overhead to the standard training process. More generally, the fact that our process of computing the weights is simple and inexpensive allows us to recompute them during training (say every few epochs) to improve our approximation of the theoretically optimal weights. We explore the effect of updating the weights during training in Section 3.2.

Estimating the weights. We estimate the weights using the Nearest Neighbors method on S_v to learn a map that takes as input the teacher's and student's "confidence" for the label of a certain example x, and outputs estimates for $p(x)$ and $\mathrm{distortion}_f(x)$ so we can apply (4). In our experiments we measure the confidence of a model either via the *margin-score*, i.e., the difference between the largest two predicted classes for the label of a given example (see e.g. the so-called "margin" uncertainty sampling variant [40]), or via the *entropy* of its prediction. However, one could use *any* metric (not necessarily confidence) that correlates well with the accuracy of the corresponding models.

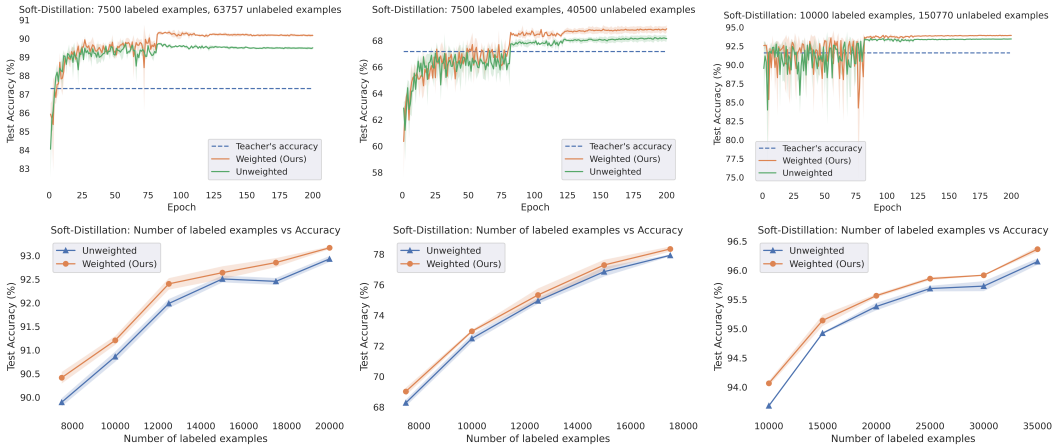
More concretely, we reduce the task of estimating the weights to a two-dimensional (i.e., two inputs and two outputs) regression task over the validation set which we solve using the Nearest Neighbors method. In particular, our Nearest Neighbor data structure is constructed as follows. Each example x of the validation set is assigned the two following pairs of points: (i) (teacher confidence at x, student confidence at x) — this is the covariate of the regression task; (ii) (1, distortion at (x)) if the teacher correctly predicts the label of x, or (0, distortion at x), if the teacher does not correctly predict the label of x — this is the response of the regression task. The query corresponding to an unlabeled example x' is of the form (teacher confidence at x', student confidence at x'). The Nearest Neighbors data structure returns the average response over the k closest in euclidean distance pairs (teacher confidence at x, student confidence at x) in the validation set. The value of k is specified in the next paragraph. The pseudocode for our method can be found in Algorithm 2 in Appendix A.

We point out two remarks. First, the number k of neighbors we use for our weights-estimation is always $k = \frac{\sqrt{|S_v|}}{2}$. This is because choosing $k = \Theta(q^{2/(2+\dim)})$, where q is the size of the validation dataset and $\dim$ is the dimension of the underlying metric space ($\dim = 2$ in our case), is asymptotically optimal, and $1/2$ is a popular choice for the hidden constant used in practice, see e.g. [9, 18]. Second, notice that (4) implies that the weight of an example could be larger than 1 if

(and only if) the corresponding distortion value (4) at that example is less than 1. This could happen for example if both the student and teacher have the same (or very similar) inaccurate prediction for a certain example. In such a case, the value of the weight in (4) informs us that the loss at this example should be larger than the (low) value the unweighted loss function suggests. However, since we do not have the ground-truth label for a point during training — but only the inaccurate prediction of the teacher — having a weight larger than 1 in this case would most likely guide our student model to fit an inaccurate label. To avoid this phenomenon, we always project our weights onto the $[0, 1]$ interval (Line 16 of Algorithm 2). In Appendix D.2 we discuss an additional reason why it is beneficial to project the weights of examples of low distortion onto $[0, 1]$ based on a MSE analysis.

3 Experimental results

In this section we present our experimental results. In Section 3.1 we consider an experimental setup according to which the weights are estimated only once during the whole training process. In Section 3.2 we study the effect of updating the weights during training. Finally, in Section 3.3 we demonstrate that our method can be combined with uncertainty-based weighting techniques by both combining and comparing our method to the fidelity-based weighting scheme of [12].



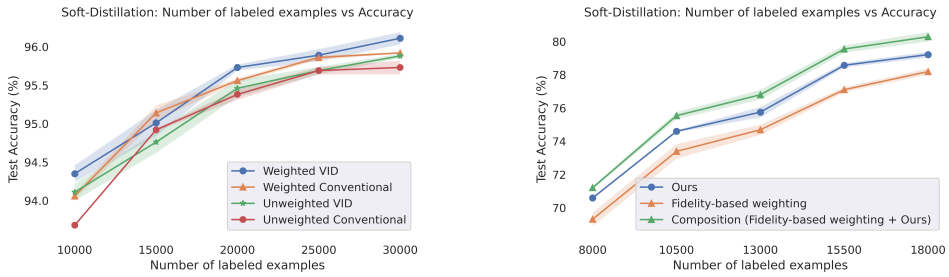
SVHN CIFAR-10 CelebA

Figure 2: The student's test accuracy over the training trajectory (first row) and student's best test-accuracy achieved over all (second row) when applying **one-shot estimation of the weights**. The teacher model is a MobileNet with depth multiplier 2. In the cases of SVHN and CIFAR-10 the student model is a MobileNet of depth multiplier 1, and in the case of CelebA is a ResNet-11. Our approach leads to consistently better models in terms of test-accuracy and convergence speed. Shaded regions correspond to values within one standard deviation of the mean.

3.1 Improvements via one-shot estimation of the weights

Here we show how applying our reweighting scheme provides consistent improvements on several popular benchmarks even when the weights are estimated only once during the whole training process. We compare our method against conventional distillation with unlabeled examples, but we also show that our method can provide benefits when combined with more advanced distillation techniques such as the framework of [2] (see Figure 9).

More concretely, we evaluate our method on benchmark vision datasets. We compare our method to conventional distillation with unlabeled examples both in terms of the best test accuracy achieved by the student, and in terms of convergence speed (see Figure 2). We also evaluate the comparative advantage of our method as a function of the number of labeled examples available (size of dataset S_ℓ). We always choose the temperature in the softmax of the models to be 1 for simplicity and consistency, and our metric for confidence is always the margin-score. Implementation details for our experiments and additional results can be found in Appendices B and C. We implemented all



CelebA CIFAR-10

Figure 3: **Left:** We compare conventional distillation (unweighted conventional), standard Variational Information Distillation (VID) [2] (unweighted VID), and reweighting the conventional loss function (weighted conventional) and the VID loss function with our method (weighted VID) on CelebA. (These results correspond to **one-shot estimation of the weights**.) We see that our method can be combined effectively with more advanced distillation techniques such as VID. **Right:** We combine and compare our method with the fidelity-based reweighting scheme of [12] on CIFAR-10. We see that our method can be combined effectively with weighting schemes based only on the teacher's uncertainty. (These results correspond to **updating our weights at the end of every epoch**.)

algorithms in Python making use of the TensorFlow deep learning library [1]. We use 64 Cloud TPU v4s each with two cores.

3.1.1 Experimental setup

Our experiments are of the following form. The academic dataset we use each time is first split into two parts A and B. Part A, which is typically smaller, is used as the labeled dataset S_ℓ where the teacher model is trained on (recall the setting we described in Section 2.3). Part B is randomly split again into two parts which represent the unlabeled dataset S_u and validation dataset S_v, respectively. Then, (i) the teacher and student models are trained once on the labeled dataset S_ℓ; (ii) the teacher model is used to generate soft-labels for the unlabeled dataset S_u; (iii) we train the student model on the union of S_ℓ and S_u using our method and conventional distillation with unlabeled examples. We repeat step (iii) a number of times: in each trial we partition part B randomly and independently, and then the student model is trained using the (student-)weights reached after completing the training on dataset S_ℓ in step (i) as initialization, both for our method and the competing approaches.

3.1.2 CIFAR-{10, 100} and SVHN experiments

SVHN [34] is an image classification dataset where the task is to classify street view numbers (10 classes). The train set of SVHN contains 73257 labeled images and its test set contains 26032 images. We use a MobileNet [21] with depth multiplier 2 as the teacher, and a MobileNet with depth multiplier 1 as the student[1]. The tables in Figure 4 contain the results of our experiments (averages over 3 trials). In each experiment we use the first $N \in \{7500, 10000, 12500, 15000, 17500, 20000\}$ examples as the labeled dataset S_ℓ, and then the rest $73257 - N$ images are randomly split to a labeled validation dataset S_v of size 2000, and an unlabeled dataset S_u of size $71257 - N$.

CIFAR-10 and CIFAR-100 [24] are image classification datasets with 10 and 100 classes respectively. Each of them consists of 60000 labeled images, which we split to a training set of 50000 images, and a test set of 10000 images. For CIFAR-10, we use a Mobilenet with depth multiplier 2 as the teacher, and a Mobilenet with depth multiplier 1 as the student. For CIFAR-100, we use a ResNet-110 as a teacher, and a ResNet-56 as the student. We use a validation set of 2000 examples. The results of our experiments (averages over 3 trials) can be found in the tables of Figures 5, 6.

3.1.3 CelebA experiments: considering more advanced distillation techniques

As we have already discussed in Section 1.1, our method can be combined with more advanced distillation techniques which aim to enforce greater consistency between the teacher and the student.

[1]Note that we see the student outperform the teacher here and in other experiments, as can often happen with distillation with unlabeled examples, particularly when the teacher is trained on limited data.

Labeled Examples	7500	10000	12500	15000	17500
Teacher	87.31%	88.5%	88.45%	91.38%	91.32%
Weighted (Ours)	**90.41 ± 0.12%**	**91.21 ± 0.09%**	**92.4 ± 0.12%**	**92.64 ± 0.14%**	**92.85 ± 0.1%**
Unweighted	89.89 ± 0.09%	90.86 ± 0.11%	91.99 ± 0.1%	92.51 ± 0.07%	92.45 ± 0.07%

Figure 4: Experiments on SVHN. See Section 3.1.2 for details.

Labeled Examples	7500	10000	12500	15000	17500
Teacher	67.17%	71.39%	74.69%	77%	78.46%
Weighted (Ours)	**69.01 ± 0.25%**	**72.96 ± 0.1%**	**75.33 ± 0.43%**	**77.69 ± 0.35%**	**78.34 ± 0.16%**
Unweighted	68.27 ± 0.22%	72.48 ± 0.26%	74.95 ± 0.2%	76.85 ± 0.32%	77.92 ± 0.06%

Figure 5: Experiments on CIFAR-10. See Section 3.1.2 for details.

Labeled Examples	7500	10000	12500	15000	17500
Teacher	44.1%	51.31%	55.54%	59.05%	62.17%
Weighted (Ours)	**46.81 ± 0.34%**	**53.31 ± 0.66%**	**57.5 ± 0.3%**	**60.94 ± 0.32%**	**62.86 ± 0.41%**
Unweighted	46.29 ± 0.04%	52.83 ± 0.53%	56.89 ± 0.65%	60.73 ± 0.03%	62.79 ± 0.09%

Figure 6: Experiments on CIFAR-100. See Section 3.1.2 for details.

We demonstrate this fact by implementing the method of Variational Information Distillation for Knowledge Transfer (VID) [2] and showing how it is indeed beneficial to combine it with our method. We chose the gender binary classification task of CelebA [15] as our benchmark (see details in the next paragraph), because it is known (see e.g. [31]) that the more advanced distillation techniques tend to be more effective when applied to classification tasks with few classes. In the table below, "Unweighted VID" corresponds to the implementation of loss described in equations (2), (4) and (6) of [2], and "Weighted VID" corresponds to the reweighting of the latter loss using our method.

CelebA [15] is a large-scale face attributes dataset with more than 200000 celebrity images, each with forty attribute annotations. Here we consider the binary male/female classification task. The train set of CelebA contains 162770 images and its test set contains 19962 images. We use a MobileNet with depth multiplier 2 as the teacher, and a ResNet-11 [19] as the student. The tables in Figure 7 contain the results of our experiments (averages over 3 trials). In each experiment we use the first $N \in \{10000, 15000, 20000, 25000, 30000, 35000\}$ examples as the labeled dataset S_ℓ, and then the rest $162770 - N$ images are randomly split to a labeled validation dataset S_v of size 2000, and an unlabeled dataset S_u of size $160770 - N$.

Labeled Examples	10000	15000	20000	25000	30000
Teacher	91.59%	93.76%	94.41%	94.86%	94.92%
Weighted VID	**94.35 ± 0.11%**	95.01 ± 0.17%	**95.73 ± 0.04%**	**95.89 ± 0.08%**	**96.11 ± 0.08%**
Weighted Conventional	94.06 ± 0.04%	**95.14 ± 0.1%**	95.56 ± 0.03%	**95.86 ± 0.03%**	95.92 ± 0.01%
Unweighted VID [2]	94.11 ± 0.11%	94.76 ± 0.14%	95.46 ± 0.11%	95.69 ± 0.05%	95.88 ± 0.03%
Unweighted Conventional	93.68 ± 0.01%	94.92 ± 0.02%	95.38 ± 0.07%	95.69 ± 0.05%	95.73 ± 0.09%

Figure 7: Experiments on CelebA. See Section 3.1.3 for details and also Figure 3.

3.1.4 ImageNet experiments

ImageNet [41] is a large-scale image classification dataset with 1000 classes consisting of approximately $I \approx 1.3$M images. For our experiments, we use a ResNet-50 as the teacher, and a ResNet-18 as the student. In each experiment we use the first $N \in \{64058, 128116\}$ labeled examples (5% and 10% of I, respectively) as the labeled dataset S_ℓ, and the rest $I - N$ examples are randomly split to a labeled validation dataset S_v of size 10000, and an unlabeled dataset S_u of size $I - N - 10000$. The results of our experiments (averages over 10 trials) can be found in Figure 8.

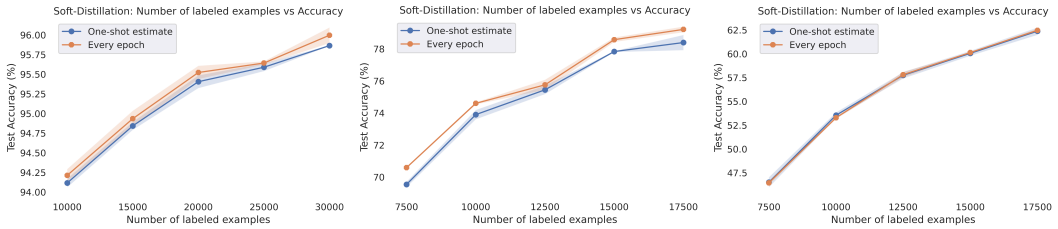
Labeled Examples	5% of ImageNet	10% of ImageNet
Teacher (soft)	36.74%	51.88%
Weighted (Ours)	**38.60 ± 0.07%**	**53.59 ± 0.09%**
Unweighted	38.44 ± 0.06%	53.43 ± 0.08%

Labeled Examples	5% of ImageNet	10% of ImageNet
Teacher (hard)	36.74%	51.88%
Weighted (Ours)	**38.56 ± 0.07%**	**53.34 ± 0.11%**
Unweighted	38.42 ± 0.06%	53.18 ± 0.07%

Figure 8: Experiments with soft-distillation (left) and hard-distillation (right) on ImageNet.

3.2 Updating the weights during training

In this section we consider the effect of updating our estimation of the optimal weights during training. For each dataset we consider, the experimental setup is identical to the corresponding setting in Section 3.1, except for that we always use a validation set of size 500 and the entropy of a model's prediction as the metric for its confidence. We note that the time required for computing the weight for each example is insignificant compared to total training time (less than 1% of the total training time in all of our experiments), which allows us to conduct experiments in which we update our estimation at the end of every epoch. We see that doing this typically significantly improves the resulting student's performance (however, in CIFAR-100 we do not observe substantial benefits).



CelebA CIFAR-10 CIFAR-100

Figure 9: The effect of updating the weights during training. We compare our method when (i) weights are estimated only once and; (ii) when we update our estimation at the end of every epoch.

3.3 Combining and comparing with uncertainty-based weighting schemes

As we have already discussed in Section 1.1, our method can be combined with uncertainty-based weighting schemes as these are independent of the student model and, therefore, they can be seen as a preprocessing step (modifying the loss function) before applying our method. We demonstrate this by combining and comparing our method with the filelity-based weighting scheme of [12] on CIFAR-10 and CIFAR-100. Details about our experimental setup can be found in Appendix C.3.

Labeled Examples	7500	10000	12500	15000	17500
Teacher (soft)	67.55%	72.85%	74.85%	77.63%	78.40%
Our Method	$70.59 \pm 0.05\%$	$74.59 \pm 0.07\%$	$75.75 \pm 0.32\%$	$78.56 \pm 0.14\%$	$79.21 \pm 0.17\%$
Fidelity-based weighting [12]	$69.31 \pm 0.41\%$	$73.39 \pm 0.44\%$	$74.69 \pm 0.31\%$	$77.10 \pm 0.16\%$	$78.19 \pm 0.19\%$
Composition	$\mathbf{71.20 \pm 0.097\%}$	$\mathbf{75.54 \pm 0.21\%}$	$\mathbf{76.80 \pm 0.29\%}$	$\mathbf{79.55 \pm 0.22\%}$	$\mathbf{80.28 \pm 0.28\%}$

Labeled Examples	7500	10000	12500	15000	17500
Teacher (soft)	44.70%	51.45%	56.00%	58.70%	61.82%
Our Method	$46.18 \pm 0.25\%$	$53.15 \pm 0.16\%$	$\mathbf{57.51 \pm 0.4\%}$	$\mathbf{59.65 \pm 0.4\%}$	$62.02 \pm 0.25\%$
Fidelity-based weighting [12]	$46.32 \pm 0.23\%$	$52.92 \pm 0.20\%$	$57.40 \pm 0.47\%$	$59.48 \pm 0.17\%$	$61.39 \pm 0.09\%$
Composition	$\mathbf{46.62 \pm 0.44\%}$	$\mathbf{53.39 \pm 0.16\%}$	$57.11 \pm 0.41\%$	$\mathbf{59.63 \pm 0.15\%}$	$\mathbf{62.30 \pm 0.26\%}$

Figure 10: Combining and comparing our method with the weighting scheme of [12] on CIFAR-10 (top table) and CIFAR-100 (bottom table). See also Figure 3.

4 Theoretical motivation

In this section we show that our debiasing reweighting method described in Section 2.2 comes with provable **statistical** and **optimization** guarantees. We first show that the method is statistically consistent in the sense that, with a sufficiently large dataset, the reweighted risk converges to the true or "clean" risk. We show (see Appendix D) the following convergence guarantee.

Theorem 4.1 (Uniform Convergence of Reweighted Risk)**.** *Assume that S is a dataset of i.i.d. "noisy" samples from $\mathbb{D}$. Under standard capacity assumptions for the class of models $\mathcal{F}$ and regularity assumptions for the loss $\ell(\cdot)$, for every $f \in \mathcal{F}$ it holds that*

$$\lim_{|S|\to\infty} R_S^w(f) = R(f) \quad \textit{and} \quad \lim_{|S|\to\infty} R_S(f) = R(f) + \mathrm{Bias}(f)$$

To prove our optimization guarantees we analyze the reweighted objective in the fundamental case where the model $f(x;\Theta)$ is linear, i.e., $f(x;\Theta) = \Theta x \in \mathbb{R}^L$, and the loss $\ell(y,z)$ is convex in z for every y. In this case, the composition of the loss and the model $f(x;\Theta)$ is convex as a function of the parameter $\Theta \in \mathbb{R}^{L\times d}$. Recall that we denote by $f_{\mathrm{true}}(x) : \mathbb{R}^d \mapsto \mathbb{R}^L$ the ground truth classifier and by $\mathbb{P}$ the "clean" distribution, i.e., a sample from $\mathbb{P}$ has the form $(x, f_{\mathrm{true}}(x))$ where x is drawn from a distribution $\mathbb{X}$ supported on (a subset of) $\mathbb{R}^d$. Finally, we denote by $\mathbb{D}$ the "noisy" labeled distribution on $\mathbb{R}^d \times \mathbb{R}^L$ and assume that the x-marginal of $\mathbb{D}$ is also $\mathbb{X}$.

We next give a general definition of debiasing weight functions, i.e., weighting mechanisms that make the corresponding objective function an unbiased estimator of the clean objective $R(\Theta)$ for every parameter vector $\Theta \in \mathbb{R}^d$. Recall that the weight function defined in Section 2.2 is debiasing.

Definition 4.2 (Debiasing Weights)**.** *We say that a weight function $w(x, y_{\mathrm{adv}};\Theta)$ is a debiasing weight function if it holds that*

$$R^w(\Theta) \triangleq \mathbb{E}_{(x,y_{\mathrm{adv}})\sim\mathbb{D}}[w(x, y_{\mathrm{adv}};\Theta)\ell(y_{\mathrm{adv}}, f(x;\Theta))] = R(\Theta)\,.$$

Since the loss is convex in Θ, one could try to optimize the naive objective that does not reweight and simply minimizes $\ell(\cdot)$ over the noisy examples, $R^{\mathrm{naive}}(\Theta) \triangleq \mathbb{E}_{(x,y_{\mathrm{adv}})\sim\mathbb{D}}[\ell(y_{\mathrm{adv}}, \Theta x)]$. We show (unsurprisingly) that doing this is a bad idea: there are instances where optimizing the naive objective produces classifiers with bad generalization error over clean examples. For the formal statement and proof we refer the reader to Appendix E.

SGD on the naive objective $R^{\mathrm{naive}}(\cdot)$ learns parameters with arbitrarily bad generalization error over the "clean" data.

Our main theoretical insight is that optimizing linear models with the reweighted loss leads to parameters with almost optimal generalization.

Given a debiasing weight function $w(\cdot)$, SGD on the reweighted objective $R^w(\cdot)$ learns a parameter with almost optimal generalization error over the "clean" data.

The main issue with optimizing the reweighted objective is that, in general, we have no guarantees that the weight function preserves its convexity (recall that it depends on the parameter Θ). However, we know that its population version corresponds to the clean objective $R(\cdot)$ which is a convex objective. We show that we can use the convexity of the underlying clean objective to show results for stochastic gradient descent, by proving the following key structural property.

Proposition 4.3 (Stationary Points of the Reweighted Objective Suffice (Informal))**.** *Let S be a dataset of $n = \mathrm{poly}(dL/\epsilon)$ i.i.d. samples from the noisy distribution $\mathbb{D}$. Let $\widehat{\Theta}$ be any stationary point of the weighted objective $R^w_S(\Theta)$ constrained on the unit ball (with respect to the Frobenious norm $\|\cdot\|_F$). Then, with probability at least 99%, it holds that*

$$R(\widehat{\Theta}) \leq \min_{\|\Theta\|_F \leq 1} R(\Theta) + \epsilon\,.$$

5 Conclusion

We propose a principled reweighting scheme for distillation with unlabeled examples. Our method is hyper-parameter free, adds minimal implementation overhead, and comes with theoretical guarantees. We evaluated our method on standard benchmarks and we showed that it consistently provides significant improvements. We note that investigating improved data-driven ways of estimating the weights (4) could be of interest, since potential inaccurate estimation of the weights is the main limitation of our work. We leave this question open for future work. We envisage that our approach can be used in high-impact, large-scale applications, due to its simplicity and effectiveness.

To conclude, knowledge-distillation is a highly popular method for training Deep Neural Networks. As such, the potential negative societal impact of our work is related to undesirable, malicious usage of the Deep Learning technology. This is an important societal issue, as the latter has widespread applications that range from Natural Language Processing to Robotics and Healthcare.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[3] Martin Anthony and Peter Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 1999.

[4] Noga Bar, Tomer Koren, and Raja Giryes. Multiplicative reweighting for robust neural network optimization. *arXiv preprint arXiv:2102.12192*, 2021.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[7] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein contrastive representation distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16296–16305, 2021.

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[9] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[10] Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. COLT'05, page 249–263, Berlin, Heidelberg, 2005. Springer-Verlag.

[11] Damek Davis and Dmitriy Drusvyatskiy. Stochastic subgradient method converges at the rate $o(k^{-1/4})$ on weakly convex functions. *arXiv preprint arXiv:1802.02988*, 2018.

[12] Mostafa Dehghani, Arash Mehrjou, Stephan Gouws, Jaap Kamps, and Bernhard Schölkopf. Fidelity-weighted learning. *arXiv preprint arXiv:1711.02799*, 2017.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[15] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[16] Dragan Gamberger, Nada Lavrac, and Ciril Groselj. Experiments with noise filtering in a medical domain. In *ICML*, volume 99, pages 143–151, 1999.

[17] Ying Guo, Peter L Bartlett, John Shawe-Taylor, and Robert C Williamson. Covering numbers for support vector machines. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 267–277, 1999.

[18] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[22] Prateek Jain, Dheeraj M. Nagaraj, and Praneeth Netrapalli. Making the last iterate of sgd information theoretically optimal. *SIAM Journal on Optimization*, 31(2):1108–1130, 2021.

[23] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[25] Abhishek Kumar and Ehsan Amid. Constrained instance and class reweighting for robust learning under label noise. *arXiv preprint arXiv:2111.05428*, 2021.

[26] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[27] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[28] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020.

[29] Negin Majidi, Ehsan Amid, Hossein Talebi, and Manfred K Warmuth. Exponentiated gradient reweighting for robust training under label noise and beyond. *arXiv preprint arXiv:2104.01493*, 2021.

[30] Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

[31] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. Subclass distillation. *arXiv preprint arXiv:2002.03936*, 2020.

[32] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[33] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[35] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, 2021.

[36] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

[37] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4119–4128, 2018.

[38] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

[39] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049, 1996.

[40] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *European Conference on Machine Learning*, pages 413–424. Springer, 2006.

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[42] Henry Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[43] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34:6906–6919, 2021.

[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[45] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[46] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[47] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[48] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

[49] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

[50] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[51] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020.

[52] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 3 for an empirical evaluation of our method. See Section 4 for a theoretical justification for the performance of our approach.
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See theoretical results in Section 4 and the Supplementary Material.
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide complete proofs of all theoretical results in the Supplementary Material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3 and the Supplementary Material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All plots in Section 3 contain shaded regions that correspond to the values within one standard deviation of the mean.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Sec. 3
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3 and the Supplementary Material.
   (b) Did you mention the license of the assets? [Yes] See the Supplementary Material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Formal description of our method

In this section we present pseudocode for our method.

Algorithm 1 Weighted distillation with unlabeled examples

Require: model Teacher, model Student, labeled dataset $S_\ell = \{(x_i, y_i)\}_{i=1}^{m}$, unlabeled dataset $S_u = \{x_i\}_{i=m+1}^{m+n}$, validation dataset $S_v = \{(x_i, y_i)\}_{i=m+n+1}^{m+n+q}$, number of weights-estimating iterations r

Train Teacher and Student on S_ℓ
Use Teacher to generate labels for S_u to obtain set $S_u^\ell = \{(x, \text{Teacher}(x)) \mid x \in S_u\}$
for $i = m+1$ to $n+m$ **do**
    $y_i \leftarrow \text{Teacher}(x_i)$
$S \leftarrow \{(x_i, y_i)\}_{i=1}^{m+n}$
for $i = 1$ to m **do**
    $w(x_i) \leftarrow 1$
for $i = 1$ to r **do**
    $\{w(x_{m+1}), \ldots, w(x_{m+n})\} \leftarrow$ ESTIMATEWEIGHTS(Teacher, Student, S_v, S_u^ℓ)
    Train Student on S using the weighted empirical risk:

$$\frac{1}{m+n} \sum_{i=1}^{m+n} w(x_i)\ell(y_i, \text{Student}(x_i))$$

Algorithm 2 Procedure for estimating the weights

procedure ESTIMATEWEIGHTS(Teacher, Student, V, D)
    ▷ V is the validation dataset and D is the teacher-labeled dataset
    $U \leftarrow \emptyset$, $k \leftarrow \lceil \frac{1}{2}\sqrt{|V|} \rceil$
    for every $(x, y) \in V$ **do**
        $X \leftarrow (\text{Confidence}(\text{Teacher}(x)), \text{Confidence}(\text{Student}(x)))$
        if $\arg\max(\text{Teacher}(x)) = \arg\max(y)$ **then**:
            $(p, \text{distortion}) \leftarrow (0, 1)$
        else:
            $(p, \text{distortion}) \leftarrow \left(1, \frac{\ell(\text{Teacher}(x), \text{Student}(x))}{\ell(y, \text{Student}(x))}\right)$
        $Y \leftarrow (p, \text{distortion})$
        $U \leftarrow U \cup \{(X, Y)\}$
    Weights $= \varnothing$ ▷ Initialize and empty list for the weights
    for every $(x, y) \in D$ **do**
        Query $\leftarrow (\text{Confidence}(\text{Teacher}(x)), \text{Confidence}(\text{Student}(x)))$
        $(\hat{p}, \hat{d}) \leftarrow k\text{-NN}(U, \text{Query})$ ▷ Predict $p(x)$ and $\text{distortion}_f(x)$ from the k nearest neighbors of Query in U
        $w(x) \leftarrow \min\left\{1, \frac{1}{1+\hat{p}(\hat{d}-1)}\right\}$
        Append $w(x)$ to Weights
    return Weights

B Extended experiments

B.1 The student's test-accuracy-trajectory

In this section we provide extended experimental results that show the student's test accuracy over the training trajectory corresponding to experiments we mentioned in Section 3.1. Notice that in the vast majority of cases our method significantly outperforms the conventional approach almost throughout the training process.

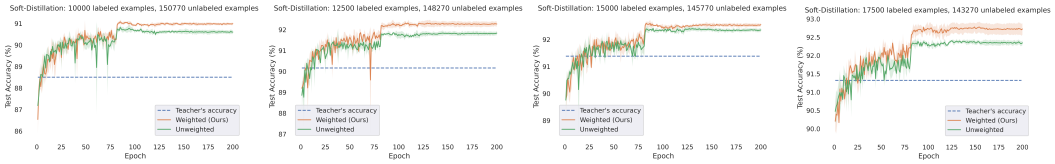


Figure 11: **SVHN** experiments. The student's test accuracy over the training trajectory using hard-distillation corresponding to the experiments of Figure 4. See Section 3.1.2 for more details.

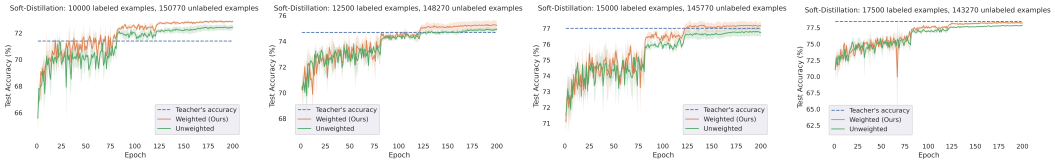


Figure 12: **CIFAR-10** experiments. The student's test accuracy over the training trajectory corresponding to the experiments of Figure 5. See Section 3.1.2 for more details.

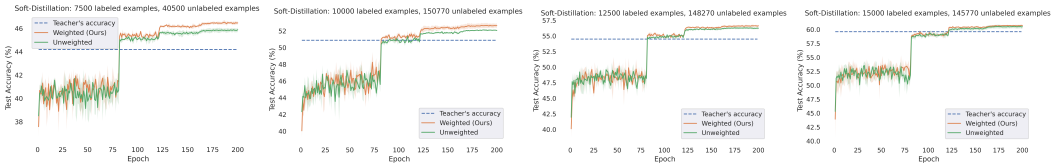


Figure 13: **CIFAR-100** experiments. The student's test accuracy over the training trajectory corresponding to the experiments of Figure 6. See Section 3.1.2 for more details.

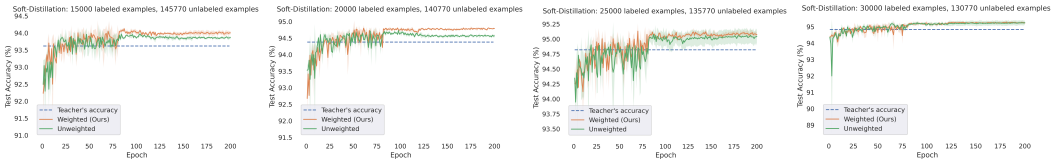


Figure 14: **CelebA** experiments. The student's test accuracy over the training trajectory corresponding to the experiments of Figure 7. See Section 3.1.3 for more details.

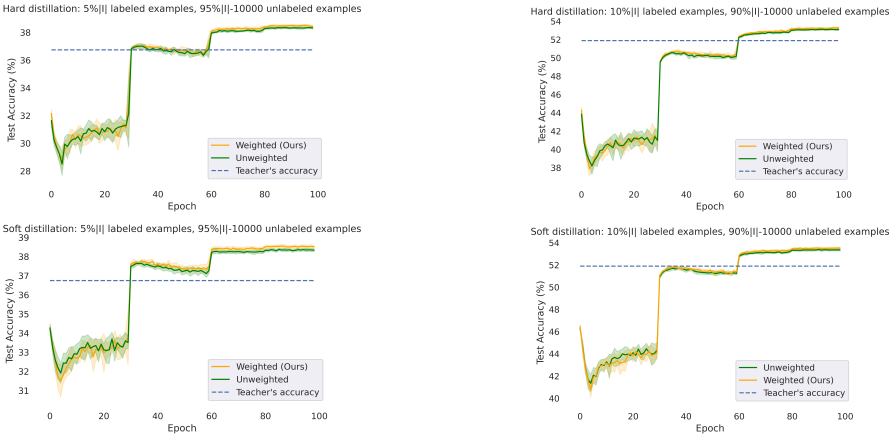


Figure 15: **ImageNet** experiments. The student's test accuracy over the training trajectory using hard-distillation (first row) and soft-distillation (second row) corresponding to the experiments of Figure 8. See Section 3.1.4 for more details.

B.2 Considering the effect of temperature

Temperature-scaling, a technique introduced in the original paper of Hinton et. al. [20], is one the most common ways for improving student's performance in distillation. Indeed, it is known (see e.g. [43]) that choosing the right value for the temperature can be quite beneficial, to the point it can outperform other more advanced techniques for improving distillation. Here we demonstrate that our approach provides benefits on top of any improvement on can get via temperature-scaling by conducting an ablation study on the effect of temperature on CIFAR-100. In our experiment, the teacher model is a Resnet-110 achieving accuracy 56.0%, the student model is a Resnet-56, the number of labeled examples is 12500, the validation set consists of 500 examples, and we use the entropy of a prediction as a metric of confidence. We apply our method using one-shot estimation of the weights. We compare training the student model using conventional distillation to using our method for different values of temperature. The results can be found in the table below. We see that in almost all cases the student-model trained using our method outperforms the student-model trained using conventional distillation and, in particular, the best student overall is the result of choosing 2.0 for the value of temperature and applying our method.

Temperature	Unweighted	Weighted (ours)
0.01	$52.84 \pm 0.08\%$	$53.73 \pm 0.11\%$
0.10	$54.63 \pm 0.09\%$	$54.84 \pm 0.12\%$
0.50	$56.45 \pm 0.12\%$	$57.01 \pm 0.1\%$
0.80	$56.67 \pm 0.12\%$	$57.60 \pm 0.15\%$
1.00	$57.17 \pm 0.15\%$	$57.56 \pm 0.09\%$
2.00	$57.54 \pm 0.11\%$	$\mathbf{57.8 \pm 0.21}\%$
3.00	$57.20 \pm 0.18\%$	$57.09 \pm 0.25\%$
5.00	$56.92 \pm 0.11\%$	$57.01 \pm 0.2\%$

Figure 16: Ablation study on the effect of temperature on CIFAR-100. See Appendix B.2 for details

C Implementation details

In this section we describe the implementation details of our experiments. Recall the description of our method in Section 2.3.

C.1 Experiments on CelebA, CIFAR-10, CIFAR-100, SVHN

All of our experiments are performed according to the following recipe. In all cases, the loss function $\ell : \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}_+$ we use is the cross-entropy loss. We train the teacher model for 200 epochs on dataset S_ℓ. We pretrain the student model for 200 epochs on dataset S_ℓ and save its parameters. Then, using the latter saved parameters for initialization each time, we train the student model for 200 epochs optimizing either the weighted or conventional (unweighted) empirical risk, and report its average performance over three trials.

We use the Adam optimizer. The initial learning rate is $\mathrm{lr} = 0.001$. We proceed according to the following learning rate schedule (see e.g., [19]):

$$\mathrm{lr} \leftarrow \begin{cases} \mathrm{lr} \cdot 0.5 \cdot 10^{-3}, & \text{if } \#\text{epochs} > 180 \\ \mathrm{lr} \cdot 10^{-3}, & \text{if } \#\text{epochs} > 160 \\ \mathrm{lr} \cdot 10^{-2}, & \text{if } \#\text{epochs} > 120 \\ \mathrm{lr} \cdot 10^{-1}, & \text{if } \#\text{epochs} > 80 \end{cases}$$

Finally, we use data-augmentation. In particular, we use random horizontal flipping and random width and height translations with width and height factor, respectively, equal to 0.1.

C.2 Experiments on ImageNet

For the ImageNet experiments we follow a similar although not identical recipe to the one described in Appendix C.1. In each training stage above, we train the model (teacher or student) for 100 epochs instead of 200. We also use SGD with momentum 0.9 instead of Adam as the optimizer. For data-augmentation we use only random horizontal flipping. Finally, the learning rate schedule is as

follows. For the first 5 epochs the learning rate lr is increased from 0.0 to 0.1 linearly. After that, the learning rate changes as follows:

$$\mathrm{lr} = \begin{cases} 0.01, & \text{if } \#\text{epochs} > 30 \\ 0.001, & \text{if } \#\text{epochs} > 60 \\ 0.0001, & \text{if } \#\text{epochs} > 80 \end{cases}$$

C.3 Details on the experimental setup of Section 3.3

In the CIFAR-10 experiments of Section C.3 the teacher model is a MobileNet with depth multiplier 2, and the student model is a MobileNet with depth multiplier 1. In the CIFAR-100 experiments of the same section, the teacher model is a ResNet-110, and the student model is a ResNet-56. We use a validation set consisting of 500 examples (randomly chosen as always). The student of each method has access to the same number of labeled examples, i.e., the validation set is used for training the student model as we describe in Remark 2.1. We compare the following three methods:

- **Fidelity weighting scheme [12].** For every example x we use the entropy of the teacher's prediction as an uncertainty/confidence measure, which we denote by $\mathrm{entropy}(x)$. We then compute the exponential weights described in [12] as $w(x) = \exp(-\mathrm{entropy}(x)/\overline{\mathrm{entropy}})$, where $\overline{\mathrm{entropy}}$ is the average entropy of the teacher's predictions over all training examples.
- **Our method.** We use the entropy as the metric for confidence. In the case of CIFAR-10 we re-estimate the weights at the end of every epoch. In the case of CIFAR-100 the weights are estimated only once in the beginning of the process.
- **Composition.** We reweight each example in the loss function by multiplying the weights resulting from the two methods above.

D Extended theoretical motivation: statistical aspects

In this section we study the statistical aspects of our approach. In Section D.1 we revisit and formally state Theorem 4.1 — see Corollary D.2 and Remark D.1. In Section D.2 we perform a Mean-Squared-Error analysis that provides additional justification of our choice to always project the weights on the $[0, 1]$ interval (recall Line 16 of Algorithm 2). Finally, in Section D.3 we provide the proof of Proposition 2.1 which was omitted from the main body of the paper.

D.1 Statistical motivation

Recall the background on multiclass classification in Section 2.1. In this section we study hypothesis classes $\mathcal{F}$ and loss functions $\ell : \mathbb{R}^L \times \mathbb{R}^L :\to \mathbb{R}$ that are "well-behaved" with respect to a certain (standard in the machine learning literature) complexity measure we describe below.

For $\epsilon > 0$, a class $\mathcal{H}$ of functions $h : \mathcal{X} \to [0, 1]$ and an integer n, the "growth function" $\mathcal{N}_\infty(\epsilon, \mathcal{H}, n)$ is defined as

$$\mathcal{N}_\infty(\epsilon, \mathcal{H}, n) = \sup_{\mathbf{x} \in \mathcal{X}^n} \mathcal{N}(\epsilon, \mathcal{H}(\mathbf{x}), \|\cdot\|_\infty), \tag{6}$$

where $\mathcal{H}(\mathbf{x}) = \{(h(x_1), \ldots, h(x_n)) : h \in \mathcal{H}\} \subseteq \mathbb{R}^n$ and for $A \subseteq \mathbb{R}^n$ the number $\mathcal{N}(\epsilon, A, \|\cdot\|_\infty)$ is the smallest cardinality A_0 of a set $A_0 \subseteq A$ such that A is contained in the union of ϵ-balls centered at points in A_0, in the metric induced by $\|\cdot\|_\infty$ The growth number is a complexity measure of function classes commonly used in the machine learning literature [3, 17].

The following theorem from [30] provides large deviation bounds for function classes of polynomial growth.

Theorem D.1 (Theorem 6, [30]). *Let Z be a random variable taking values in $\mathcal{Z}$ distributed according to distribution μ, and let $\mathcal{H} : \mathcal{Z} \to [0, 1]$ be a class of functions. Fix $\delta \in (0, 1), n \geq 16$ and set*

$$\mathcal{M}(n) = 10\mathcal{N}_\infty(1/n, \mathcal{H}, 2n).$$

Then with probability at least $1-\delta$ *in the random vector* $Z=(Z_1,\ldots,Z_n)\sim\mu^n$*, for* every $h\in\mathcal{H}$ *we have:*

$$\left|\mathbb{E}[h(Z)]-\frac{1}{n}\sum_{i=1}^{n}h(Z_i)\right|\leq\sqrt{\frac{18\mathbb{V}_n(h,Z)\ln(2\mathcal{M}(n)/\delta)}{n}}+\frac{15\ln\left(2\mathcal{M}(n)/\delta\right)}{n-1},$$

where $\mathbb{V}_n(h,Z)$ *is the sample variance of the sequence* $\{h(Z_i)\}_{i=1}^n$.

A straightforward corollary of Theorem D.1 and Proposition 2.1, and the main motivation for our method, is the following corollary.

Corollary D.2. *Let* $\ell:\mathbb{R}^L\times\mathbb{R}^L\to[0,1]$ *be a loss function and fix* $\delta>0$*. Consider any hypothesis class* $\mathcal{F}$ *of predictors* $f:\mathcal{X}\to\mathbb{R}^L$*, and the two induced classes* $\mathcal{H}\subseteq[0,1]^{\mathbb{R}^L\times\mathbb{R}^L}$*,* $\mathcal{H}^w\subseteq[0,1]^{\mathbb{R}^L\times\mathbb{R}^L}$ *of functions* $h_f(x,y):=\ell(y,f(x))$ *and* $h_f^w(x,y):=w_f(x)\ell(y,f(x))$*, respectively. Fix* $\delta>0$*,* $n\geq16$*, and set* $\mathcal{M}(n)=10\mathcal{N}_\infty(1/n,\mathcal{H},2n)$ *and* $\mathcal{M}^w(n)=10\mathcal{N}_\infty(1/n,\mathcal{H}^w,2n)$*. Then, with probability at least* $1-\delta$ *over* $S=\{x_i,y_i\}_{i=1}^n\sim\mathbb{D}^n$*,*

$$|R(f)+\mathrm{Bias}(f)-R_S(f)| = O\left(\sqrt{\mathbb{V}_S(f)\cdot\frac{\ln\frac{\mathcal{M}(n)}{\delta}}{n}}+\frac{\ln\frac{\mathcal{M}(n)}{\delta}}{n}\right) \tag{7}$$

$$|R(f)-R_S^w(f)| = O\left(\sqrt{\mathbb{V}_S^w(f)\cdot\frac{\ln\frac{\mathcal{M}^w(n)}{\delta}}{n}}+\frac{\ln\frac{\mathcal{M}^w(n)}{\delta}}{n}\right) \tag{8}$$

where $\mathbb{V}_S(f),\mathbb{V}_S^w(f)$ *are the sample variances of the loss values* $\{h_f(x_i,y_i)\}_{i=1}^n$*,* $\{h_f^w(x_i,y_i)\}_{i=1}^n$*, respectively.*

The following remark formally captures Theorem 4.1.

Remark D.1. *Under the assumptions of Corollary D.2, if we additionally have that* $\mathcal{M}(n)$ *and* $\mathcal{M}^w(n)$ *are polynomially bounded in* n*, then, for every* $f\in\mathcal{F}$ *it holds that*

$$\lim_{|S|\to\infty}R_S^w(f)=R(f)\quad\textit{and}\quad\lim_{|S|\to\infty}R_S(f)=R(f)+\mathrm{Bias}(f).$$

D.2 Studying the MSE of a fixed prediction

In this section we study the Mean-Squared-Error (MSE) of a fixed prediction $f(x)$ for an arbitrary instance $x\in\mathcal{X}$, predictor f, and loss function $\ell:\mathbb{R}^L\times\mathbb{R}^L\to\mathbb{R}_+$, in order to gain some understanding on when the importance weighting scheme could potentially underperform the standard unweighted approach from a bias-variance perspective (i.e., when the training sample is "small enough" so that asymptotic considerations are ill-suited). These considerations lead us to an additional justification for always projecting the weights to the $[0,1]$ interval (recall Line 16 of Algorithm 2).

Formally, we study the behavior of the quantities:

$$\begin{aligned}\mathrm{MSE}(x) &= \mathbb{E}_{y|x}\left[(\ell(f_{\mathrm{true}}(x),f(x))-\ell(y,f(x)))^2\right],\\ \mathrm{MSE}^w(x) &= \mathbb{E}_{y|x}[(\ell(f_{\mathrm{true}}(x),f(x))-w_f(x)\ell(y(x),f(x)))^2].\end{aligned}$$

Recalling the definition of distortion (4) we have the following proposition.

Proposition D.3. *Let* $\ell:\mathbb{R}^L\times\mathbb{R}^L\to\mathbb{R}_+$ *be a bounded loss function. Fix* $x\in\mathcal{X}$ *and a predictor* $f:\mathcal{X}\to\mathbb{R}^L$*. We have* $\mathrm{MSE}(x)<\mathrm{MSE}^w(x)$ *if and only if:*

1. $\mathrm{distortion}_f(x)<1/2$*; and*
2. $p(x)\in\left(0,\frac{1-2\cdot\mathrm{distortion}_f(x)}{(1-\mathrm{distortion}_f(x))^2}\right)$.

Proof sketch. Via direct calculations we obtain:

$$\begin{aligned}\mathrm{MSE}(x) &= \mathbb{E}_{y|x}\left[(\ell(f_{\mathrm{true}}(x),f(x))-\ell(y,f(x)))^2\right]\\ &= p(x)(\ell(f_{\mathrm{true}}(x),f(x))-\ell(y_{\mathrm{adv}}(x),f(x)))^2\\ &= p(x)\ell(f_{\mathrm{true}}(x),f(x))^2(1-\mathrm{distortion}_f(x))^2\end{aligned} \tag{9}$$

and

$$\begin{aligned}
\mathrm{MSE}^w(x) &= \mathbb{E}_{y|x}[(\ell(f_{\text{true}}(x), f(x)) - w_f(x)\ell(y, f(x)))^2] \\
&= (1-p(x))\ell(f_{\text{true}}(x), f(x))^2(1-w_f(x))^2 \\
&\quad + p(x)(\ell(f_{\text{true}}(x), f(x)) - w_f(x)\ell(y_{\text{adv}}(x), f(x)))^2 \\
&= (1-p(x))\ell(f_{\text{true}}(x), f(x))^2(1-w_f(x))^2 \\
&\quad + p(x)\ell(f_{\text{true}}(x), f(x))^2(1-w_f(x)\text{distortion}_f(x))^2 \qquad (10)
\end{aligned}$$

Recalling the definition of weights (4), and combining it with (9) and (10) implies the claim.

□

In words, Proposition D.3 implies that when the adversary does not have the power to corrupt the label of an instance x with high enough probability, i.e., $p(x)$ is sufficiently small, and the prediction of the student is "close enough" to the adversarial label (i.e., when $\text{distortion}_f(x)$ is small enough), then it potentially makes sense to use the unweighted estimator instead of the weighted one from a bias-variance trade-off perspective, as the former has smaller MSE in this case. Notice that this observation aligns well with our method as we always project $w_f(x)$ to $[0, 1]$ (observe that $w_f(x) > 1$ iff $\text{distortion}_f(x) < 1$ and $p(x) > 0$).

D.3 Proof of Proposition 2.1

Recall the weight, distortion and bias definitions in (4) and Proposition 2.1. We prove the first claim of Proposition 2.1 via direct calculations:

$$\begin{aligned}
\mathbb{E}[R_S(f)] &= \mathbb{E}_{S\sim\mathbb{D}^n}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i))\right] \\
&= \mathbb{E}_{(x,y)\sim\mathbb{D}}\left[\ell(y, f(x))\right] \\
&= \mathbb{E}_{x\sim\mathbb{X}}[\mathbb{E}_{y|x}[\ell(y, f(x))]] \\
&= \mathbb{E}_{x\sim\mathbb{X}}[p(x)\ell(y_{\text{adv}}(x), f(x)) + (1-p(x))\ell(f_{\text{true}}(x), f(x))] \\
&= \mathbb{E}_{x\sim\mathbb{X}}[\ell(f_{\text{true}}(x), f(x))] + \mathbb{E}_{x\sim\mathbb{X}}[p(x)\cdot(\ell(y_{\text{adv}}(x), f(x)) - \ell(f_{\text{true}}, f(x)))] \\
&= \mathbb{E}_{x\sim\mathbb{X}}[\ell(f_{\text{true}}(x), f(x))] + \mathbb{E}_{x\sim\mathbb{X}}\left[p(x)\cdot\left(\frac{\ell(y_{\text{adv}}(x), f(x))}{\ell(f_{\text{true}}(x), f(x))} - 1\right)\cdot\ell(f_{\text{true}}(x), f(x))\right] \\
&= \mathbb{E}_{x\sim\mathbb{X}}[\ell(f_{\text{true}}(x), f(x))] + \mathbb{E}_{x\sim\mathbb{X}}\left[p(x)\cdot(\text{distortion}_f(x) - 1)\cdot\ell(f_{\text{true}}(x), f(x))\right] \\
&= R(f) + \text{Bias}(f).
\end{aligned}$$

Similarly for the second claim:

$$\begin{aligned}
\mathbb{E}[R_S^w(f)] &= \mathbb{E}_{S\sim\mathbb{D}^n}\left[\frac{1}{n}\sum_{i=1}^{n}w_f(x_i)\ell(y_i, f(x_i))\right] \qquad (11) \\
&= \mathbb{E}_{x\sim\mathbb{X}}\left[\mathbb{E}_{y|x}\left[w_f(x)\ell(y, f(x))\right]\right] \\
&= \mathbb{E}_{x\sim\mathbb{X}}[w_f(x)\cdot(p(x)\ell(y_{\text{adv}}(x), f(x)) + (1-p(x))\ell(f_{\text{true}}(x), f(x)))] \\
&= \mathbb{E}_{x\sim\mathbb{X}}\left[\frac{p(x)\ell(y_{\text{adv}}(x), f(x)) + (1-p(x))\ell(f_{\text{true}}(x), f(x))}{1 + p(x)\cdot(\text{distortion}_f(x) - 1)}\right] \\
&= \mathbb{E}_{x\sim\mathbb{X}}\left[\frac{\ell(f_{\text{true}}(x), f(x)) + \ell(f_{\text{true}}(x), f(x))\cdot p(x)\cdot(\text{distortion}_f(x) - 1)}{1 + p(x)\cdot(\text{distortion}_f(x) - 1)}\right] \\
&= R(f),
\end{aligned}$$

concluding the proof.

E Extended theoretical motivation: optimization aspects

To prove our optimization guarantees, we analyze the reweighted objective in the fundamental case where the model $f(x;\Theta)$ is linear, i.e., $f(x;\Theta) = \Theta x \in \mathbb{R}^L$, and the loss $\ell(y, z)$ is convex in z for

every y. In this case, the composition of the loss and the model $f(x;\Theta)$ is convex as a function of the parameter $\Theta \in \mathbb{R}^{L\times d}$. Recall that we denote by $f_{\text{true}}(x) : \mathbb{R}^d \mapsto \mathbb{R}^L$ the ground truth classifier and by $\mathbb{P}$ the "clean" distribution, i.e., a sample from $\mathbb{P}$ has the form $(x, f_{\text{true}}(x))$ where x is drawn from a distribution $\mathbb{X}$ supported on (a subset of) $\mathbb{R}^d$. Finally, we denote by $\mathbb{D}$ the "noisy" labeled distribution on $\mathbb{R}^d \times \mathbb{R}^L$ and assume that the x-marginal of $\mathbb{D}$ is also $\mathbb{X}$.

Notation In what follows, for any elements r, q of the same dimensions we denote by $r \cdot q$ their inner product. For example for two vectors $r, q \in \mathbb{R}^d$ we have $r \cdot q = \sum_{i=1}^d r_i q_i$. Similarly, for two matrices $\Theta, Q \in \mathbb{R}^{L\times d}$ we have $\Theta \cdot Q = \sum_{i=1}^L \sum_{j=1}^d \Theta_{ij} Q_{ij}$. We denote by $\|\cdot\|_2$ the ℓ_2 for vectors and the spectral norm for matrices. We use $\otimes$ to denote the standard tensor (Kronecker) product between two vectors or matrices. For example, for two matrices A, B we have $(A\otimes B)_{ijkl} = A_{ij}B_{kl}$ and for two vectors v, u we have $(v \otimes u)_{ij} = v_i u_j$. We denote by $\|\cdot\|_F$ the Frobenious norm for matrices. We remark that we use standard asymptotic notation $O(\cdot)$, etc. and $\widetilde{O}(\cdot)$ to omit factors that are poly-logarithmic (in the appearing arguments).

For example, training a linear model $f(x;\Theta) = \Theta x$ with the Cross Entropy loss corresponds to using $\ell(t, y) = \sum_{i=1}^L t_i \log(\frac{e^{y_i}}{\sum_{j=1}^L e^{y_j}})$ and minimizing the objective

$$R(\Theta) = \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(y, f(x;\Theta))] = \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(y, \Theta x)] \,.$$

More generally, in what follows we shall refer to the population loss over the clean distribution $\mathbb{P}$ as $R(\cdot)$, i.e.,

$$R(\Theta) \triangleq \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(y, f(x;\Theta))] \,.$$

We next give a general definition of debiasing weight functions, i.e., weighting mechanisms that make the corresponding objective function an unbiased estimator of the clean objective $R(\Theta)$ for every parameter vector $\Theta \in \mathbb{R}^d$.

Definition E.1 (Debiasing Weights)**.** *We say that a weight function $w(x, y_{\text{adv}};\Theta) : \mathbb{R}^d \times \mathbb{R}^L \mapsto \mathbb{R}$ is a debiasing weight function if it holds that*

$$R^w(\Theta) \triangleq \mathbb{E}_{(x,y_{\text{adv}})\sim\mathbb{D}}[w(x, y_{\text{adv}};\Theta)\ell(y_{\text{adv}}, f(x,\Theta))] = R(\Theta) \,.$$

Remark E.1. *We remark that the weight function $w(\cdot)$ depends on the current hypothesis, Θ, and also on the noise advice $p(x)$ that we are given with every example. In order to keep the notation simple, we do not explicitly track these dependencies and simply write $w(x, y_{\text{adv}};\Theta)$. We also remark that, in general, in order to construct the weight function w we may also use "clean" data, which may be available, e.g., as a validation dataset, as we did in Section 2.2.*

Our main result is that, given a convex loss $\ell(\cdot)$ and a debiasing weight function $w(\cdot)$ that satisfy standard regularity assumptions, stochastic gradient descent on the reweighted objective produces a parameter vector with good generalization error. We first present the assumptions on the example distributions, the loss, and the weight function. In what follows, we view the gradient of a function $q(\Theta) : \mathbb{R}^{L\times d} \mapsto \mathbb{R}$ as an $L \times d$-matrix and the hessian $\nabla^2 q(\Theta)$ as an $(L \times d) \times (L \times d)$-tensor (or equivalently as a $dL \times dL$-matrix).

Definition E.2 (Regularity Assumptions)**.** *The x-marginal $\mathbb{X}$ of $\mathbb{D}$ and $\mathbb{P}$ is supported on (a subset of) the ball of radius $R > 0$, $\mathcal{B}_R \triangleq \{x \in \mathbb{R}^d : \|x\|_2 \le R\}$.*

The training model is linear $f(x;\Theta) = \Theta x$ and the parameter space is the unit ball, i.e., $\|\Theta\|_F \le 1$.

For every label $y_{\text{adv}} \in \mathbb{R}^L$ in the support of $\mathbb{D}$, the loss $z \mapsto \ell(y_{\text{adv}}, z)$ is a twice differentiable, convex function in z. Moreover $\ell(y_{\text{adv}}, z)$ is M_ℓ-bounded, L_ℓ-Lipschitz, and B_ℓ-smooth, i.e., $|\ell(y_{\text{adv}}, z)| \le M_\ell$, $\|\nabla_z \ell(y_{\text{adv}}, z)\|_2 \le L_\ell$, and $\|\nabla_z^2 \ell(y_{\text{adv}}, z)\|_2 \le B_\ell$, for all z with $\|z\|_2 \le R$.

For every example $(x, y_{\text{adv}}) \in \mathbb{R}^d \times \mathbb{R}^L$ in the support of $\mathbb{D}$ the weight function $\Theta \mapsto w(x, y_{\text{adv}};\Theta)$ is twice differentiable, M_w-bounded, L_w-Lipschitz, and B_w-smooth, i.e., $|w(x, y_{\text{adv}};\Theta)| \le M_w$, $\|\nabla_\Theta w(x, y_{\text{adv}};\Theta)\|_F \le L_w$, and $\|\nabla_\Theta^2 w(x, y_{\text{adv}};\Theta)\|_2 \le B_w$ for all Θ with $\|\Theta\|_F \le 1$ [2].

Remark E.2. *Observe that if a property in the above definition is satisfied by some parameter-value Q, then it is also satisfied for any other $Q' > Q$. For example, if the loss function is 0.5-Lipschitz it is*

[2] Recall that, formally, $\nabla_\Theta^2 w(x, y_{\text{adv}};\Theta)$ is a $(L \times d) \times (L \times d)$-tensor G. For this tensor G we overload notation and set $\|G\|_2$ to be the standard ℓ_2 operator norm when we view G as an $(Ld) \times (Ld)$-matrix.

also 1*-Lipschitz. Therefore, to simplify the expressions, in what follows we shall assume (without loss of generality) that all the regularity parameters, i.e.,* $R, M_\ell, L_\ell, B_\ell, M_w, L_w, B_w$*, are larger than* 1.

Since the loss is convex, it is straightforward to optimize the naive objective that does not reweight the loss and simply minimizes $\ell(\cdot)$ over the noisy examples, $R^{\text{naive}}(\Theta) \triangleq \mathbb{E}_{(x,y_{\text{adv}})\sim\mathbb{D}}[\ell(y_{\text{adv}}, \Theta x)]$. We first show that (unsurprisingly) it is not hard to construct instances (even in binary classification) where optimizing the naive objective produces classifiers with large generalization error over clean examples. For simplicity, since in the following lemma we consider binary classification, we assume that the labels $y \in \{\pm 1\}$ and the parameter of the linear model is a vector $\theta \in \mathbb{R}^d$.

Proposition E.3 (Naive Objective Fails)**.** *Fix any* $c \in [0, 1]$*. Let* $\ell(\cdot)$ *be the Binary Cross Entropy loss, i.e.,* $\ell(t) = \log(1 + e^{-t})$*. There exists a "clean" distribution* $\mathbb{P}$ *and a noisy distribution* $\mathbb{D}$ *on* $\mathbb{R}^d \times \{\pm 1\}$ *so that the following hold.*

1. *The* x*-marginal of both* $\mathbb{P}$ *and* $\mathbb{D}$ *is uniform on a sphere.*
2. *The clean labels of* $\mathbb{P}$ *are consistent with a linear classifier* $\text{sign}(\theta^* \cdot x)$*.*
3. $\mathbb{D}$ *has (total) label noise* $\Pr_{(x,y_{\text{adv}})\sim\mathbb{D}}[y_{\text{adv}} \neq \text{sign}(\theta^* \cdot x)] = c \in [0, 1]$*.*
4. *The minimizer* $\widehat{\theta}$ *of the (population) naive objective* $R^{\text{naive}}(\theta) = \mathbb{E}_{(x,y)\sim\mathbb{D}}[\ell(y_{\text{adv}}\theta \cdot x)]$*, constrained on the unit has generalization error*

$$R(\widehat{\theta}) \geq \min_{\|\theta\|_2 \leq 1} R(\theta) + c/2\,,$$

where $R(\theta)$ *is the "clean" risk,* $R(\theta) = \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(y\theta \cdot x)]$*.*

Our positive results show that, having a debiasing weight function $w(\cdot)$ that is not very "wild" (see the regularity assumptions of Definition E.2) and optimizing the corresponding weighted objective $R^w(\Theta) = \mathbb{E}_{(x,y_{\text{adv}})\sim\mathbb{D}}[w(x, y_{\text{adv}}; \Theta)\ell(y_{\text{adv}}, \Theta x)]$ with SGD, gives models with almost optimal generalization. The main issue with optimizing the reweighted objective is that, in general, we have no guarantees that the weight function preserves its convexity (recall that it depends on the parameter Θ). However, we know that its population version corresponds to the clean objective $R(\cdot)$ which is a convex objective. We show that we can use the convexity of the underlying clean objective to show results for both single- and multi-pass stochastic gradient descent. We first focus on single-pass stochastic gradient descent where at every iteration a fresh noisy sample (x, y_{adv}) is drawn from $\mathbb{D}$, see Algorithm 3.

Algorithm 3 Single-Pass Stochastic Gradient Descent Algorithm

Input: Number of iterations T, Step size sequence $\eta^{(t)}$
Output: Parameter vector $\Theta^{(T)}$.

Initialize $\Theta^{(1)} \leftarrow 0$.

For $t = 1, \ldots, T$:

Draw sample $(x^{(t)}, y^{(t)}_{\text{adv}}) \sim \mathbb{D}$.

Update using the gradient of the weighted objective:

$$\Theta^{(t+1)} \leftarrow \text{proj}_{\mathcal{B}}\left(\Theta^{(t)} - \eta^{(t)}\nabla_\Theta\left(w(x^{(t)}, y^{(t)}_{\text{adv}}; \Theta^{(t)})\,\ell(y^{(t)}_{\text{adv}}, \Theta^{(t)}x^{(t)})\right)\right)$$

Return $\Theta^{(T)}$.

Theorem E.4 (Generalization of Reweighted Single-Pass SGD)**.** *Assume that the example distributions* $\mathbb{P}, \mathbb{D}$*, the loss* $\ell(\cdot)$*, and the weight function* $w(\cdot)$ *satisfy the assumptions of Definition E.2. Set* $\kappa = L_w M_\ell + R M_w L_\ell$*. After* $T = \Omega(\kappa^2/\epsilon^2)$ *SGD iterations (with a step size sequence that depends on the regularity parameters of Definition E.2, see Algorithm 3), with probability at least* 99%*, it holds*

$$R(\Theta^{(T)}) \leq \min_{\|\Theta\|_F \leq 1} R(\Theta) + \epsilon\,.$$

The main observation in the single-pass setting is that, since the weight function $w(\cdot)$ is debiasing, we can view the gradients of the reweighted objective as stochastic gradients of the true objective over the clean samples. Therefore, as long as we draw a fresh i.i.d. noisy sample $(x, y) \sim \mathbb{D}$ at each round, the corresponding sequence of gradients corresponds to stochastic unbiased estimates of the gradients of the true loss $R(\Theta)$. We next turn our attention to multi-pass SGD (see Algorithm 4), where at each round we pick one of the N samples with replacement and update according to its gradient. The key difference between single- and multi-pass SGD is that the expected loss over the stochastic algorithm for single-pass SGD is the population risk, while the expected loss for multi-pass SGD is the empirical risk. In other words, in the multi-pass setting we have a stochastic gradient oracle to the empirical reweighted objective $\widehat{R}^w(\Theta) = \frac{1}{N}\sum_{i=1}^{N} w(x^{(i)}, y_{\text{adv}}^{(i)}; \Theta)\, \ell(y_{\text{adv}}^{(i)}, \Theta x^{(i)})$, which is not necessarily convex. Our second theorem shows that under the regularity conditions of Definition E.2 multi-pass SGD also achieves good generalization error.

Theorem E.5 (Generalization of Reweighted Multi-Pass SGD)**.** *Assume that the example distributions $\mathbb{P}, \mathbb{D}$, the loss $\ell(\cdot)$, and the weight function $w(\cdot)$ satisfy the assumptions of Definition E.2. Set $\kappa = RM_\ell L_\ell B_\ell M_w L_w B_w$ and define the empirical reweighted objective with $N = (dL)^2/\epsilon^2 \operatorname{poly}(\kappa)$ i.i.d. samples $(x^{(1)}, y_{\text{adv}}^{(1)}), \ldots, (x^{(N)}, y_{\text{adv}}^{(N)})$ from the noisy distribution $\mathbb{D}$ as*

$$\widehat{R}^w(\Theta) = \frac{1}{N}\sum_{i=1}^{N} w(x^{(i)}, y_{\text{adv}}^{(i)}; \Theta)\, \ell(y_{\text{adv}}^{(i)}, \Theta x^{(i)})\,.$$

Then, after $T = \operatorname{poly}(\kappa)/\epsilon^4$ iterations, multi-pass SGD with constant step size sequence $\eta^{(t)} = C/\sqrt{T}$ [3] (see Algorithm 4) on $\widehat{R}^w(\cdot)$ outputs a list $\Theta^{(1)}, \ldots, \Theta^{(T)}$ that, with probability at least 99%, contains a vector $\widehat{\Theta}$ that satisfies

$$R(\widehat{\Theta}) \le \min_{\|\Theta\|_F \le 1} R(\Theta) + \epsilon\,.$$

We remark that our analysis also applies to the multi-pass SGD variant where, at every epoch we pick a random permutation of the N samples and update with their gradients sequentially.

Algorithm 4 Multi-Pass Stochastic Gradient Descent Algorithm

Input: Number of Rounds T, Number of Samples N, Step size sequence $\eta^{(t)}$.
Output: List of weight vectors $\Theta^{(1)}, \ldots, \Theta^{(T)}$.

Draw N i.i.d. samples $(x^{(1)}, y_{\text{adv}}^{(1)}), \ldots, (x^{(N)}, y_{\text{adv}}^{(N)}) \sim \mathbb{D}$.

Initialize $\Theta^{(1)} \leftarrow 0$.

For $t = 1, \ldots, T$:

Pick I uniformly at random from $\{1, \ldots, N\}$ and update using the gradient of the reweighted objective:

$$\Theta^{(t+1)} \leftarrow \operatorname{proj}_{\mathcal{B}}\left(\Theta^{(t)} - \eta^{(t)}\nabla_\Theta\left(w(x^{(I)}, y_{\text{adv}}^{(I)}; \Theta^{(t)})\ell(y_{\text{adv}}^{(I)}, \Theta^{(t)}x^{(I)})\right)\right)\,.$$

Return $\Theta^{(1)}, \ldots, \Theta^{(T)}$.

E.1 The proof of Proposition E.3

In this subsection we restate and prove Proposition E.3.

Proposition E.6 (Naive Objective Fails (Restate of E.3))**.** *Fix any $c \in [0, 1]$. Let $\ell(\cdot)$ be the Binary Cross Entropy loss, i.e., $\ell(t) = \log(1 + e^{-t})$. There exists a "clean" distribution $\mathbb{P}$ and a noisy distribution $\mathbb{D}$ on $\mathbb{R}^d \times \{\pm 1\}$ so that the following hold.*

1. *The x-marginal of both $\mathbb{P}$ and $\mathbb{D}$ is uniform on a sphere.*

[3] C is a constant that depends on the regularity parameters of Definition E.2.

2. *The clean labels of* $\mathbb{P}$ *are consistent with a linear classifier* $\mathrm{sign}(\theta^* \cdot x)$.

3. $\mathbb{D}$ *has (total) label noise* $\mathrm{Pr}_{(x,y_{\mathrm{adv}})\sim\mathbb{D}}[y_{\mathrm{adv}} \neq \mathrm{sign}(\theta^* \cdot x)] = c \in [0,1]$.

4. *The minimizer* $\widehat{\theta}$ *of the (population) naive objective* $R^{\mathrm{naive}}(\theta) = \mathbb{E}_{(x,y)\sim\mathbb{D}}[\ell(y_{\mathrm{adv}}\theta \cdot x)]$, *constrained on the unit has generalization error*

$$R(\widehat{\theta}) \geq \min_{\|\theta\|_2 \leq 1} R(\theta) + c/2\,,$$

where $R(\theta)$ *is the "clean" risk,* $R(\theta) = \mathbb{E}_{(x,z)\sim\mathbb{P}}[\ell(z\theta \cdot x)]$.

Proof. We set the $\mathbb{X}$-marginal to be the uniform distribution on a sphere of radius $R > 0$ to be specified later in the proof. We first observe that the unit vector θ^* minimizes the (clean) Binary Cross Entropy $R(\theta)$. We can now pick a different parameter vector $\widetilde{\theta}$ with angle $\phi \in [0,\pi]$ with θ^*, and construct a noisy instance as follows: we first draw $x \sim \mathbb{X}$ (recall that we want the x-marginal of the noisy distribution to be the same as the "clean") and then set $y_{\mathrm{adv}} = \mathrm{sign}(\widetilde{\theta} \cdot x)$. By the symmetry of the uniform distribution on the sphere we have that

$$\mathbf{Pr}_{(x,y)\sim\mathbb{D}}[y_{\mathrm{adv}} \neq \mathrm{sign}(\theta^* \cdot x)] = \mathbf{Pr}_{x\sim\mathbb{D}_x}[\mathrm{sign}(\widetilde{\theta} \cdot x) \neq \mathrm{sign}(\theta^* \cdot x)] = \frac{\phi}{\pi}\,.$$

Therefore, by picking the angle ϕ to be equal to πc we obtain that $\mathbf{Pr}_{x\sim\mathbb{D}_x}[\mathrm{sign}(\widetilde{\theta}\cdot x) \neq \mathrm{sign}(\theta^*\cdot x)] = c$ as required by Proposition E.3. Moroever, we have that the minimizer of the "naive" BCE objective (constrained on the unit ball) is $\widetilde{\theta}$ and the minimizer of the clean objective is θ^*. We have that

$$\begin{aligned} R(\widetilde{\theta}) &= \mathbb{E}_{x\sim\mathbb{X}}[\log(1 + e^{-\mathrm{sign}(\theta^* \cdot x)\widetilde{\theta}\cdot x})] \geq \mathbb{E}_{x\sim\mathbb{X}}[\mathbb{1}\{\mathrm{sign}(\theta^* \cdot x)\widetilde{\theta} \cdot x < 0\}] \\ &= \mathbf{Pr}_{x\sim\mathbb{X}}[\mathrm{sign}(\widetilde{\theta} \cdot x) \neq \mathrm{sign}(\theta^* \cdot x)] = c\,. \end{aligned}$$

Moreover, we have that

$$\begin{aligned} R(\theta^*) &= \mathbb{E}_{x\sim\mathbb{X}}[\log(1 + e^{-\mathrm{sign}(\theta^* \cdot x)\theta^* \cdot x})] \\ &= \mathbb{E}_{x\sim\mathbb{X}}[\log(1 + e^{-|\theta^* \cdot x|})] \end{aligned}$$

We next need to bound from below the "margin" of the optimal weight vector θ^*, i.e., provide a lower bound on $|\theta^* \cdot x|$ that holds with high probability. We will use the following anti-concentration inequality on the probability of a origin-centered slice under the uniform distribution on the sphere. For a proof see, e.g., Lemma 4 in [10].

Lemma E.7 (Anti-Concentration of Uniform vectors, [10])**.** *Let* $v \in \mathbb{R}^d$ *be any unit vector and let* $\mathbb{X}$ *be the uniform distribution on the sphere. It holds that*

$$\mathbf{Pr}_{x\sim\mathbb{X}}\left[|v \cdot x| \leq \frac{\gamma}{\sqrt{d}}\right] \leq \gamma\,.$$

Using Lemma E.7, we obtain that

$$\mathbb{E}_{x\sim\mathbb{X}}[\log(1 + e^{-|\theta^* \cdot x|})] \leq \log(2)\gamma + \log(1 + e^{-\gamma R/\sqrt{d}})(1-\gamma) \leq 2\gamma + e^{-\gamma R/\sqrt{d}}\,,$$

where, at the last step, we used the elementary inequality $\log(1+x) \leq x$. Assuming that $R/\sqrt{d}$ is much larger than 1, we can pick $\gamma = (\sqrt{d}/R)\log(R/\sqrt{d})$. For this choice of γ we obtain that $\mathbb{E}_{x\sim\mathbb{X}}[\log(1 + e^{-|\theta^* \cdot x|})] = O(\sqrt{d}/R \log(d/R))$. Therefore, for $R = O(\sqrt{d}/c)$ we obtain that

$$\mathbb{E}_{x\sim\mathbb{X}}[\log(1 + e^{-|\theta^* \cdot x|})] \leq c/2\,.$$

Therefore, combining the above bounds we obtain that $R(\widehat{\theta}) - R(\theta^*) \geq c - c/2 \geq c/2$. □

E.2 The Proof of Theorem E.4

In this section we restate and prove our result on the generalization error of single-pass stochastic gradient descent on the weighted objective.

Theorem E.8 (Generalization of Reweighted Single-Pass SGD (Restate of E.4))**.** *Assume that the example distributions $\mathbb{P}, \mathbb{D}$ and the $\ell(\cdot)$ and weight function $w(\cdot)$ satisfy the assumptions of Definition E.2. Set $\kappa = L_w M_\ell + R M_w L_\ell$. After $T = \Omega(\kappa^2/\epsilon^2)$ SGD iterations (see Algorithm 3), with probability at least 99%, it holds*

$$R(\Theta^{(T)}) \leq \min_{\|\Theta\|_F \leq 1} R(\Theta) + \epsilon\,.$$

Proof. We observe that, since w is a debiasing weight function, given a sample $(x^{(t)}, y^{(t)}_{\text{adv}}) \sim \mathbb{D}$ it holds that $\nabla(w(x^{(t)}, y^{(t)}_{\text{adv}}; \Theta)\ell(y_{\text{adv}}, \Theta x^{(t)}))$ is an unbiased gradient estimate of $\nabla_\Theta R(\Theta)$. We will use the following result on the convergence of the last-iterate of SGD for convex objectives. For simplicity, we state the following theorem for the case where the parameter θ is a vector in $\mathbb{R}^d$ (instead of a $L \times d$ matrix).

Lemma E.9 (Last Iterate Stochastic Gradient Descent [22])**.** *Let $\mathcal{W}$ be a closed convex set of diameter R. Moreover, let $F : \mathbb{R}^d \mapsto \mathbb{R}$ be a convex, L-Lipschitz function. Define the stochastic gradient descent iteration as*

$$\theta^{(0)} \leftarrow 0$$
$$\theta^{(t+1)} \leftarrow \text{proj}_{\mathcal{W}}\left(\theta^{(t)} - \eta^{(t)} g^{(t)}(\theta^{(t)})\right)$$

where $g^{(t)}(\theta^{(t)})$ is an unbiased gradient estimate of $\nabla_\theta f_{\text{true}}(\theta^{(t)})$. Assume that for all $t \in [T]$ it holds $\|g^{(t)}(\theta)\|_2 \leq L$ for all $\theta \in \mathcal{W}$. There exists a step size sequence $\eta^{(t)}$ that depends only on T, L, R such that, with probability at least $1 - \delta$, it holds

$$F(\theta^{(T)}) \leq F(\theta^*) + O\left(RL\sqrt{\frac{\log(1/\delta)}{T}}\right)\,.$$

To simplify notation we let $\ell^{(t)}(\Theta) \triangleq \ell(y^{(t)}_{\text{adv}}, \Theta x^{(t)})$ and $w^{(t)}(\Theta) \triangleq w(x^{(t)}, y^{(t)}_{\text{adv}}; \Theta)$. For a sample $(x^{(t)}, y^{(t)}_{\text{adv}})$, the gradient $g^{(t)}$ of the weighted loss is:

$$\begin{aligned} g^{(t)} &= \nabla_\Theta(w^{(t)}(\Theta)\ell^{(t)}(\Theta)) \\ &= \ell^{(t)}(\Theta)\nabla_\Theta w^{(t)}(\Theta) + w^{(t)}(\Theta)\nabla_\Theta \ell^{(t)}(\Theta) \\ &= \ell^{(t)}(\Theta)\nabla_\Theta w^{(t)}(\Theta) + w^{(t)}(\Theta)\nabla_z \ell(x^{(t)} y^{(t)}_{\text{adv}}, z)\big|_{z=\Theta x^{(t)}} (x^{(t)})^T \in \mathbb{R}^{L\times d}\,. \end{aligned}$$

Using the triangle inequality for the Frobenious norm and the assumptions of Definition E.2 on the functions $w(\cdot)$ and $\ell(\cdot)$, we obtain that $\|g^{(t)}\|_F \leq L_w M_\ell + R M_w L_\ell$. Using Lemma E.9 we obtain that with $T = \Omega((L_w M_\ell + R M_w L_\ell)^2/\epsilon^2)$, the last iteration of Algorithm 3 satisfies the claimed guarantee. □

E.3 The proof of Theorem E.5

In this section we prove our result on multi-pass SGD. For convenience, we first restate it.

Theorem E.10 (Generalization of Multi-Pass SGD (Restate of E.5))**.** *Set $\kappa = R M_\ell L_\ell B_\ell M_w L_w B_w$ and define the empirical reweighted objective with $N = d^2/\epsilon^2 \text{ poly}(\kappa)$ i.i.d. samples $(x^{(1)}, y^{(1)}_{\text{adv}}), \ldots, (x^{(N)}, y^{(N)}_{\text{adv}})$ from the noisy distribution $\mathbb{D}$ as*

$$\widehat{R}^w(\Theta) = \frac{1}{N}\sum_{i=1}^{N} w(x^{(i)}, y^{(i)}_{\text{adv}}; \Theta)\ \ell(\theta \cdot x^{(i)} y^{(i)}_{\text{adv}})\,.$$

Then, after $T = \text{poly}(\kappa)/\epsilon^4$ iterations, multi-pass SGD (see Algorithm 4) on $\widehat{R}^w(\cdot)$ outputs a list $\theta^{(1)}, \ldots, \theta^{(T)}$ that, with probability at least 99%, contains a vector $\widehat{\theta}$ that satisfies

$$R(\widehat{\Theta}) \leq \min_{\|\Theta\|_F \leq 1} R(\Theta) + \epsilon\,.$$

Proof. To prove the theorem we shall first show that all stationary points of the empirical objective (which for arbitrary weight functions $w(\cdot)$ may be non-convex) will have good generalization guarantees. Before we proceed we formally define approximate stationary points. To simplify notation we shall assume that the parameter is a vector $\theta \in \mathbb{R}^d$. The definition extends directly to the case where L is a function of a parameter matrix Θ by using the corresponding matrix inner product.

Definition E.11 (ϵ-approximate Stationary Points)**.** *Let $L : \mathbb{R}^d \mapsto \mathbb{R}$ be a differentiable function and C be any convex subset of $\mathbb{R}^d$. A vector $\theta \in \mathbb{R}^d$ is an ϵ-approximate stationary point of $L(\cdot)$ if for every $\theta' \in C$ it holds that*

$$\left| \nabla_\theta L(\theta) \cdot \frac{\theta' - \theta}{\|\theta' - \theta\|_2} \right| \leq \epsilon \,.$$

Proposition E.12. *Set $\kappa = RM_\ell L_\ell B_\ell M_w L_w B_w$ and define the empirical reweighted objective with $N = \widetilde{O}((dL/\epsilon)^2)\,\mathrm{poly}(\kappa)\log(1/\delta)$ i.i.d. samples $(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$ from the noisy distribution $\mathbb{D}$ as*

$$\widehat{R}^w(\theta) = \frac{1}{N} \sum_{i=1}^{N} w(x^{(i)}, y^{(i)}; \Theta)\; \ell(y^{(i)}, \Theta x^{(i)}) \,.$$

Let $\widehat{\Theta}$ be any ϵ-stationary point of $\widehat{R}^w(\Theta)$ constrained on $\mathcal{B}_R$. Then, with probability at least $1 - \delta$, it holds that

$$R(\widehat{\Theta}) \leq \min_{\|\Theta\|_F \leq 1} R(\Theta) + \epsilon \,.$$

Proof. We first show that, as long as the empirical gradients are close to the population gradients, any stationary point of the weighted empirical objective will achieve good generalization error. In what follows we shall denote by Θ^* the parameter that minimizes the clean objective:

$$\Theta^* \triangleq \arg\min_{\|\Theta\|_F \leq 1} R(\Theta) \,.$$

Since the population objective is convex in Θ, we have that for any Θ it holds that

$$\begin{aligned} R(\Theta) - R(\Theta^*) &\leq \nabla_\Theta R(\Theta) \cdot (\Theta - \Theta^*) \\ &= (\nabla_\Theta R(\Theta) - \nabla_\Theta \widetilde{\mathcal{L}}^w(\Theta)) \cdot (\Theta - \Theta^*) + \nabla_\Theta \widehat{R}^w(\theta) \cdot (\Theta - \Theta^*) \\ &\leq 2\|\nabla_\Theta \widehat{R}^w(\Theta) - \nabla_\Theta \widehat{R}^w(\Theta)\|_2 + \nabla_\Theta \widehat{R}^w(\Theta) \cdot (\Theta - \Theta^*) \,. \end{aligned}$$

We have that the contstraint set $\|\Theta\|_F \leq 1$ is convex and therefore for a stationary point $\widehat{\Theta}$ of $\mathcal{L}^w(\Theta)$ we have that $|\nabla_\theta \widehat{R}^w(\Theta) \cdot (\Theta - \Theta^*)| \leq \epsilon \|\Theta - \Theta^*\|_F \leq 2\epsilon$. Therefore, $\widehat{\Theta}$ satisfies

$$R(\widehat{\Theta}) - R(\Theta^*) \leq 2\|\nabla_\Theta R(\widehat{\Theta}) - \nabla_\Theta \widehat{R}^w(\widehat{\Theta})\|_2 + 2\epsilon \,.$$

Since $w(\cdot)$ is a debiasing weighting function, we know that, as the number of samples $N \to \infty$, the empirical gradients of the reweighted objective will converge to the gradients of the population clean objective $R(\cdot)$, i.e., it holds that $\nabla_\theta \widehat{R}^w(\Theta) \to \nabla_\Theta R(\Theta)$. Therefore, to finish the proof, we need to provide a uniform convergence bound for the gradient field of the empirical objective. We first consider estimating the gradient of some fixed parameter matrix Θ. We will use McDiarmid's inequality.

Lemma E.13 (McDiarmid's Inequality)**.** *Let $x_1, \ldots, x_n$ be n i.i.d. random variables taking values in $\mathcal{X}$. Let $\phi : \mathcal{X}^n \to \mathbb{R}$ be such that $|\phi(x) - \phi(x')| \leq b_i$ whenever x and x' differ only on the i-th coordinate. It holds that*

$$\mathbf{Pr}\left[|\phi(x_1, \ldots, x_n) - \mathbb{E}[\phi(x_1, \ldots, x_n)]| \geq \epsilon\right] \leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n b_i^2}\right)$$

We consider the n i.i.d. random variables $(x^{(t)}, y^{(t)}_{\mathrm{adv}})$. We have that the empirical gradient of the weighted loss function is equal to

$$\widehat{g}(\Theta) \triangleq \frac{1}{N} \sum_{t=1}^{N} \left(\ell(y_{\mathrm{adv}}; \Theta x^{(t)})\; \nabla_\Theta w(x^{(t)}, y^{(t)}_{\mathrm{adv}}; \Theta) + w(x^{(t)}, y^{(t)}_{\mathrm{adv}}; \Theta) \nabla \ell(y^{(t)}_{\mathrm{adv}}, \Theta x^{(t)})\; (x^{(t)})^T \right) \,.$$

We have that $w(x^{(t)}, y_{\text{adv}}^{(t)}; \Theta)$ is M_w-bounded and L_w-Lipschitz, ℓ is M_ℓ-bounded and L_ℓ-Lipschitz, and $\|x^{(t)}\|_2 \leq R$. Therefore, the maximum value of each coordinate of each term in the sum of the empirical gradient $\widehat{g}(\theta)$ is bounded by $L_q \triangleq L_w M_\ell + R M_w L_\ell$. Using this fact we obtain that each coordinate of the empirical gradient is a function of the N i.i.d. random variables $(x^{(t)}, y_{\text{adv}}^{(t)})$ that satisfies the bounded differences assumption with constants $b_1, \ldots, b_N$ that satisfy $b_t \leq L_q/N$. From Lemma E.13, we obtain that

$$\begin{aligned}\mathbf{Pr}\left[\|\widehat{g}(\Theta) - \nabla_\Theta R(\Theta)\|_F \geq \epsilon\right] &\leq \sum_{i=1}^{d}\sum_{j=1}^{L} \mathbf{Pr}\left[|(\widehat{g}(\Theta))_{ij} - (\nabla_\Theta R(\Theta))_{ij}| \geq \epsilon/\sqrt{dL}\right] \\ &\leq 2dL \exp\left(-\Omega\left(N\epsilon^2/(dL\, L_q^2)\right)\right)^{4}.\end{aligned}$$

We next need to provide a uniform convergence guarantee over the whole parameter space $\|\Theta\|_F \leq 1$. We will use the following standard lemma bounding the cardinality of an ϵ-net of the unit ball in d-dimensions. For a proof see, e.g., [46].

Lemma E.14 (Cover of the Unit Ball [46])**.** *Let $\mathcal{B}$ be the d-dimensional unit ball around the origin. There exists an ϵ-net of $\mathcal{B}$ with cardinality at most $(1 + 2/\epsilon)^d$.*

Since we plan to construct a net for the gradient of $w(x, y_{\text{adv}}; \Theta)\ell(y_{\text{adv}}; \Theta x)$ we first need to show that the weighted loss $w(x, y_{\text{adv}}; \Theta)\ell(y_{\text{adv}}; \Theta x)$ is a smooth function of its parameter Θ or, in other words, that its gradients do not change very fast with respect to Θ. The following lemma follows directly from the regularity assumptions of Defintion E.2 and the chain and product rules for the derivatives.

Lemma E.15. *For all $(x, y) \in \mathcal{B}_R \times \mathbb{R}^L$, it holds that the function $q(\Theta) = w(x, y; \Theta)\ell(y; \Theta x)$ is B_q-smooth for all Θ with $\|\Theta\|_F \leq 1$, with $B_q = M_\ell B_w + 2L_\ell L_w R + M_w B_\ell R^2$.*

Proof. For simplicy we shall denote $\nabla_z \ell(y; z)$ simply by $\nabla \ell(y; z)$ and similarly $\nabla_z^2 \ell(y; z)$ by $\nabla^2 \ell(y; z)$. Using the chain rule, we have that the gradient of the weighted loss $q(\Theta)$ is equal to

$$\nabla_\Theta q(\Theta) = \nabla_\Theta w(x, y; \Theta)\ \ell(y; \Theta x) + w(x, y; \Theta)\nabla \ell(y; \Theta x) x^T\,.$$

Using again the chain and product rules we find the Hessian of $q(\Theta)$:

$$\begin{aligned}\nabla_\Theta^2 q(\Theta) &= \nabla_\Theta^2 w(x, y; \Theta)\ \ell(y; \Theta x) + (\nabla \ell(y; \Theta x) x^T) \otimes \nabla_\Theta w(x, y; \Theta) \\ &+ \nabla_\Theta w(x, y; \Theta) \otimes (\nabla \ell(y; \Theta x) x^T) + w(x, y; \Theta) H\,,\end{aligned}$$

where H is the $(L \times d) \times (L \times d)$ tensor with element $H_{ijkl} = \nabla^2 \ell(y; \Theta x)_{ik} x_j x_l$. Recall that we view $\nabla_\Theta^2 q(\Theta)$ as an $Ld \times Ld$ and to prove that it is smooth we have to find its operator (spectral) norm. Using the assumptions of Definition E.2 we obtain that $\|\nabla_\Theta^2 w(x, y; \Theta)\|_2 \leq B_w M_\ell$. For the term $(\nabla \ell(y; \Theta x) x^T) \otimes \nabla_\Theta w(x, y; \Theta)$ we consider any $q \in \mathbb{R}^{Ld}$ with $\|q\|_2 = 1$. We assume that q is indexed as q_{ij} for $i = 1, \ldots, L$ and $j = 1, \ldots, d$. We have

$$\begin{aligned}q^T((\nabla \ell(y; \Theta x) x^T) \otimes \nabla_\Theta w(x, y; \Theta))q &= \left(\sum_{ij} q_{ij} (\nabla \ell(y; \Theta x))_i x_j\right)\left(\sum_{kl} q_{kl} (\nabla_\Theta w(x, y; \Theta)_{kl}\right) \\ &\leq R L_\ell L_w\,.\end{aligned}$$

Similarly, we bound the spectral norm of the term $\nabla_\Theta w(x, y; \Theta) \otimes (\nabla \ell(y; \Theta x) x^T)$. Finally for the term H we have

$$q^T H q = \sum_{ijkl} q_{ij} x_j (\nabla \ell(y; \Theta x))_{ik} q_{kl} x_l = \sum_{ik} s_i (\nabla \ell(y; \Theta x))_{ik} s_k,$$

where $s \in \mathbb{R}^L$ has $s_i = \sum_j q_{ij} x_j$. Observe that since $\|x\|_2 \leq R$ and $\|q\|_2 = 1$ we have that $\|s\|_2 \leq R$. Therefore, from the assumption of Definition E.2, we obtain that $\|H\|_2 \leq R^2 B_\ell$.

We conclude that the function $q(\theta)$ is B_q-smooth on the unit ball $\mathcal{B}$ with $B_q = M_\ell B_w + 2L_\ell L_w R + M_w B_\ell R^2$. □

Let $\mathcal{N}_\epsilon$ be an ϵ-net of the unit ball $\mathcal{B}$. Using Lemma E.15 we first observe that the vector maps $\theta \mapsto \widetilde{g}(\Theta)$ and $\Theta \mapsto \nabla_\Theta R(\Theta)$ are both B_q-Lipschitz, where B_q is the constant defined in Lemma E.15. Using the triangle inequality and the fact that $\widehat{g}(\cdot)$ and $\nabla_\Theta R(\cdot)$ are B_q-Lipschitz, we have that

$$\max_{\|\Theta\|_F \leq 1} \|\widehat{g}(\Theta) - \nabla_\Theta R(\Theta)\|_2 \leq 2B_q\epsilon + \max_{\Theta \in \mathcal{N}_\epsilon} \|\widehat{g}(\Theta) - \nabla_\Theta R(\Theta)\|_2 \,.$$

Combining the above, and performing a union bound over the ϵ-net $\mathcal{N}_\epsilon$, we obtain that

$$\mathbf{Pr}\left[\max_{\|\Theta\|_F \leq 1} \|\widetilde{g}(\Theta) - \nabla_\Theta R(\Theta)\|_F \geq (2B_q + 1)\epsilon\right] \leq (1 + 2/\epsilon)^{dL} \exp\left(-\Omega\left(N\epsilon^2/(dL\, L_q^2)\right)\right) \,.$$

We conclude that with $N = \widetilde{\Omega}((dL)^2 L_q^2 B_q^2/\epsilon^2 \log(1/\delta))$ samples, it holds that $\|\widehat{g}(\Theta) - \nabla_\Theta R(\Theta)\|_2 \leq \epsilon$, uniformly for all parameters Θ with $\|\Theta\|_F \leq 1$, with probability at least $1 - \delta$. □

We now have to show that the multi-pass SGD finds an approximate stationary point of the empirical objective. We will use the following result on non-convex projected SGD. To simplify notation, we state the following optimization lemma assuming that the parameter is a vector $\theta \in \mathbb{R}^d$.

Lemma E.16 (Non-Convex Projected Stochastic Gradient Descent [11])**.** *Let $\mathcal{W}$ be a closed convex set of diameter R. Moreover, let $F : \mathbb{R}^d \mapsto \mathbb{R}$ be an L-Lipschitz and B-smooth function. Define the stochastic gradient descent iteration as*

$$\theta^{(0)} \leftarrow 0$$
$$\theta^{(t+1)} \leftarrow \mathrm{proj}_{\mathcal{W}}\left(\theta^{(t)} - \eta^{(t)} g^{(t)}(\theta^{(t)})\right)$$

where $g^{(t)}(\theta^{(t)})$ is an unbiased gradient estimate of $\nabla_\theta F(\theta^{(t)})$. Fix a number of iterations $T \geq 1$ and assume that for all $t \in [T]$ it holds $\|g^{(t)}(\theta)\|_2 \leq L$ for all $\theta \in \mathcal{W}$. Set the step-size $\eta^{(t)} = \Theta(\sqrt{R/(BL^2T)})$. With probability at least 99%, there exists a $t \in \{1, \ldots, T\}$ such that $\theta^{(t)}$ is an $O\left(\frac{\sqrt{BLR}}{T^{1/4}}\right)$-stationary point of $F(\cdot)$ constrained on $\mathcal{W}$.

Theorem E.5 now follows directly by applying Lemma E.16 on the empirical objective to find an ϵ-approximate stationary point and then using Proposition E.12.

□