# Learning to Sample in Stochastic Optimization

Sijia Zhou[1] Yunwen Lei[2] Ata Kabán[1]

[1]University of Birmingham, Birmingham B15 2TT, United Kingdom
[2]University of Hong Kong, Pokfulam, Hong Kong, China

## Abstract

We consider a PAC-Bayes analysis of stochastic optimization algorithms, and devise a new SGDA algorithm inspired from our bounds. Our algorithm learns a data-dependent sampling scheme along with model parameters, which may be seen as assigning a probability to each training point. We demonstrate that learning the sampling scheme increases robustness against misleading training points, as our algorithm learns to avoid bad examples during training. We conduct experiments in both standard and adversarial learning problems on several benchmark datasets, and demonstrate various applications including interpretability upon visual inspection, and robustness to the ill effects of bad training points. We also extend our analysis to pairwise SGD to demonstrate the generalizability of our methodology.

## 1 INTRODUCTION

Stochastic optimization is a cornerstone in training deep learning models on large-scale datasets. These algorithms employ sampling strategies to estimate gradients and improve computational efficiency. While uniform sampling of training points is the classic approach in these optimization methods, recent studies have explored data-dependent sampling to accelerate convergence, reduce the variance of the gradient estimates, and enhance prediction accuracy [Zhao and Zhang, 2015, Allen-Zhu et al., 2016, Katharopoulos and Fleuret, 2017, Johnson and Guestrin, 2018, Wu et al., 2017, Han et al., 2022].

In real-world datasets, training examples may have a varying degree of relevance to the target, some are less typical than others, or data may contain noise or outliers, and identifying difficult examples [Agarwal et al., 2022] is an active area of research. Recent works [London, 2017, Zhou et al., 2023] made a start at developing theory to explain generalization of Stochastic Gradient Descent (SGD) with non-uniform sampling, by combining algorithmic stability [Bousquet and Elisseeff, 2002, Bousquet et al., 2020] with the PAC-Bayes framework [Shawe-Taylor and Williamson, 1997, McAllester, 1999]. However, those works only considered SGD, and only a few give practical algorithms to exploit the potential of the analysis.

In this paper, we take advantage of the PAC-Bayes machinery, namely that bounds hold uniformly for all sampling distributions over the indices of training points. This allows us to learn a sampling strategy from the data to maximize accuracy by minimizing the generalization bound. We also anticipate increased robustness due to the model averaging built into the PAC-Bayes framework. In addition, the learned sampling distribution is readily interpretable as weights on individual training examples, providing new avenues.

Moreover we develop the analytic framework to Stochastic Gradient Descent Ascent (SGDA). SGDA has gained attention in various areas, such as adversarial training [Sinha et al., 2017], generative adversarial networks (GANs) [Sanjabi et al., 2018], robust optimization [Namkoong and Duchi, 2016], and reinforcement learning [Dai et al., 2018]. In particular, it plays a key role in adversarial training, a primary defense against adversarial attacks on deep neural networks. This process is framed as a min-max optimization problem, where the goal is to optimize the model while accounting for worst-case perturbations introduced by adversaries.

Beyond SGDA, we extend our analysis to pairwise SGD. Pairwise comparisons provide deeper insights into model behavior, particularly in distinguishing human and machine visual scene recognition, which cannot be fully addressed through pointwise analysis alone. By visualizing pairwise comparisons, we uncover models' behavior when faced with atypical examples, offering a deeper understanding of the optimization process.

Our contributions are summarized as follows:

*Proceedings of the 41$^{st}$ Conference on Uncertainty in Artificial Intelligence* (UAI 2025), PMLR 244:5142–5158.

- We prove the sub-exponential stability property of SGDA and establish PAC-Bayes generalization bounds in smooth and non-smooth cases. Contrary to classic PAC-Bayes, our methodology does not require randomization of the model parameters but exploits the stochasticity of the gradient-based optimizer instead.
- Our methodology framework can be used to devise new stochastic optimization algorithms by minimizing generalization bounds w.r.t. the sampling distribution. We demonstrate the generality of this approach, obtaining two algorithms: SGDA-Q and pairwise SGD-Q.
- We conduct experiments to evaluate the proposed algorithms in both adversarial and standard training. Our results demonstrate the robustness of these algorithms on several tasks, including pairwise learning.

## 2 RELATED WORK

Non-uniform, data-dependent sampling strategies are widely used in stochastic optimization. One example is importance sampling [Zhao and Zhang, 2015], where samples are selected proportional to the gradient norm in order to reduce the variance of the gradient. This was shown to accelerate training. Various approximation methods have been devised to enhance the computational efficiency in implementing this idea [Johnson and Guestrin, 2018, Katharopoulos and Fleuret, 2018].

Some works use loss-based sampling for faster convergence [Katharopoulos and Fleuret, 2017, London, 2017], while others propose upper bounds on gradient norms for improved performance [Katharopoulos and Fleuret, 2018]. Alternative approaches include distance-based sampling [Wu et al., 2017], multi-armed bandit frameworks [Salehi et al., 2018, Liu et al., 2020], and data-dependent sampling for coordinate selection [Allen-Zhu et al., 2016]. Despite these advances, generalization analysis for non-uniform sampling remains limited, which we address in this paper.

The classic PAC-Bayes framework can compute numerical generalization bounds where the weights follow the prior and posterior distributions [Pérez-Ortiz et al., 2021b, Dziugaite and Roy, 2017], with tighter results achieved using a learned, data-dependent prior [Ambroladze et al., 2006, Parrado-Hernández et al., 2012, Rivasplata et al., 2018, Dziugaite et al., 2021, Dziugaite and Roy, 2018, Pérez-Ortiz et al., 2021a].

In this paper, we use the idea of algorithms inspired by PAC-Bayes bounds. However, contrary to existing works, we exploit the intrinsic randomness of stochastic algorithms. The indices of examples chosen for estimating the gradient directions during training are treated as hyperparameters, which follow a uniform PAC-Bayes prior at first, and a data-dependent PAC-Bayes posterior that we learn from data by minimizing the bound.

## 3 PRELIMINARIES

Let $\mathcal{D}$ be an unknown distribution on a sample space $\mathcal{Z}$. We denote by $\mathcal{W}$, $\mathcal{V} \subseteq \mathbb{R}^d$ the parameter space, and $\Phi$ will be a hyperparameter space. In the context of stochastic optimization algorithms, the hyperparameter is the random sequence of indices of training inputs used to approximate the gradient throughout iterations. The PAC-Bayes framework allows us to model this stochasticity by defining two discrete distributions on $\Phi$: the prior denoted by $\mathbb{P}$ and the PAC-Bayes posterior denoted by $\mathbb{Q}$. In the paper, we always set prior $\mathbb{P}$ as the uniform distribution and learn the posterior $\mathbb{Q}$ from the data.

Given a training set $S = \{z_1, \ldots, z_n\}$ drawn i.i.d. from $\mathcal{D}$, and a hyperparameter $\phi \in \Phi$, a learning algorithm $A$ returns a model parameterized by $A(S;\phi)$, mapping the training inputs to a hypothesis $h \in \mathcal{H}$.

The generalization error, or risk, relative to a loss function $\mathcal{L}$, is defined as

$$R(A(S;\phi)) = \mathbb{E}_{z\sim\mathcal{D}}[\mathcal{L}(A(S;\phi), z)]. \tag{1}$$

Since $\mathcal{D}$ is unknown, the empirical risk serves as a proxi:

$$R_S(A(S;\phi)) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(A(S;\phi), z_i). \tag{2}$$

We denote the difference between the risk and the empirical risk (i.e., the generalization gap) by $G(S,\phi) := R(A(S;\phi)) - R_S(A(S;\phi))$. In the PAC-Bayes framework, we work with the expected risk and expected empirical risk w.r.t. $\mathbb{Q}(S)$, to which we refer as $\mathbb{Q}$ for brevity, defined as:

$$R(\mathbb{Q}) = \mathop{\mathbb{E}}_{\phi\sim\mathbb{Q}}[R(A(S;\phi))], \quad R_S(\mathbb{Q}) = \mathop{\mathbb{E}}_{\phi\sim\mathbb{Q}}[R_S(A(S;\phi))].$$

## 4 MAIN RESULTS

Our first result is generalization bounds on SGDA with adaptive sampling (proof given in Appendix A.3). We first introduce several notations and definitions [Zhang et al., 2021].

For SGDA, we have $\mathcal{L}(\mathbf{w};\cdot) := \max_{\mathbf{v}} \ell((\mathbf{w},\mathbf{v});\cdot)$, where $\ell : \mathcal{W}\times\mathcal{V}\times\mathcal{Z} \mapsto \mathbb{R}_+$ and we define

$$A(S;\phi) := A_{\mathrm{w},\mathrm{v}}(S;\phi) = (\mathbf{w},\mathbf{v}) \in \mathcal{W}\times\mathcal{V}.$$

In SGDA, we seek the minimizer of the true risk,

$$\min_{\mathbf{w}\in\mathcal{W}} R(A_{\mathrm{w},\mathrm{v}}(S;\phi)) = \min_{\mathbf{w}\in\mathcal{W}} \max_{\mathbf{v}\in\mathcal{V}} \mathbb{E}_{z\sim\mathcal{D}}[\ell(A_{\mathrm{w},\mathrm{v}}(S;\phi), z)].$$

Since the true risk is unknown, the minimizer of the empirical risk defines the following minimax optimization problem:

$$\min_{\mathbf{w}\in\mathcal{W}} R_S(A_{\mathrm{w},\mathrm{v}}(S;\phi)) = \min_{\mathbf{w}\in\mathcal{W}} \max_{\mathbf{v}\in\mathcal{V}} \frac{1}{n}\sum_{i=1}^{n} \ell(A_{\mathrm{w},\mathrm{v}}(S;\phi), z_i).$$

| Algorithm | Sampling | Reference | Assumption | Bound Type | Rate |
|---|---|---|---|---|---|
| SGDA | Adaptive | Theorem 4.4 in this work | L,C-C(S) | w.h.p. | $\tilde{O}(1/\sqrt{n})$ |
| | Uniform | Lei et al. [2021b] | L,C-C(S) | In expectation | $O(1/\sqrt{n})$ |
| pairwise SGD | Adaptive | Zhou et al. [2025] | L,C(S) | w.h.p. | $\tilde{O}(1/\sqrt{n})$ |
| | Uniform | Lei et al. [2020] | L,S,C | w.h.p. | $\tilde{O}(1/\sqrt{n})$ |
| | Uniform | Lei et al. [2021a] | L,C | w.h.p. | $O(1/\sqrt{n})$ |

Table 1: Summary of generalization rates, either in expectation or with high probability (w.h.p.), for optimization algorithms (SGDA and pairwise SGD) under various assumptions—including Lipschitz continuity (L), smoothness (S), convexity (C), and convex-concavity (C-C)—as a function of the sample size $n$.

We say $\ell$ is convex if for any $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{W}$, we have $\ell(\mathbf{w}_1, \cdot) \geq \ell(\mathbf{w}_2, \cdot) + \langle \nabla \ell(\mathbf{w}_2, \cdot), \mathbf{w}_1 - \mathbf{w}_2 \rangle$. We say that $\ell$ is concave if $-\ell$ is convex.

**Definition 4.1** (Convexity-Concavity)**.** *We say $\ell : (\mathbf{w}, \mathbf{v}) \mapsto \ell((\mathbf{w}, \mathbf{v}); z)$ is convex-concave if for any $\mathbf{v} \in \mathcal{V}$, the function $\mathbf{w} \mapsto \ell((\mathbf{w}, \mathbf{v}), \cdot)$ is convex and for any $\mathbf{w} \in \mathcal{W}$, the function $\mathbf{v} \mapsto \ell((\mathbf{w}, \mathbf{v}), \cdot)$ is concave.*

**Definition 4.2** (Lipschitz)**.** *Let $L \geq 0$. For any $z$, we say $\ell : (\mathbf{w}, \mathbf{v}) \mapsto \ell((\mathbf{w}, \mathbf{v}); z)$ is $L$-Lipschitz if the following inequalities hold for all $\mathbf{w} \in \mathcal{W}$, $\mathbf{v} \in \mathcal{V}$ and $z \in \mathcal{Z}$*

$$\|\nabla_{\mathbf{w}} \ell((\mathbf{w}, \mathbf{v}); z)\|_2 \leq L \quad \textit{and} \quad \|\nabla_{\mathbf{v}} \ell((\mathbf{w}, \mathbf{v}); z)\|_2 \leq L.$$

**Definition 4.3** (Smoothness)**.** *$\ell : (\mathbf{w}, \mathbf{v}) \mapsto \ell((\mathbf{w}, \mathbf{v}); z)$ is said to be $\alpha$-smooth, $\alpha > 0$, if for all $\mathbf{w}_1$, $\mathbf{w}_2 \in \mathcal{W}$, $\mathbf{v}_1$, $\mathbf{v}_2 \in \mathcal{V}$ and $z \in \mathcal{Z}$, the following holds*

$$\left\| \begin{pmatrix} \nabla_{\mathbf{w}} \ell((\mathbf{w}_1, \mathbf{v}_1); z) - \nabla_{\mathbf{w}} \ell((\mathbf{w}_2, \mathbf{v}_2); z) \\ \nabla_{\mathbf{v}} \ell((\mathbf{w}_1, \mathbf{v}_1); z) - \nabla_{\mathbf{v}} \ell((\mathbf{w}_2, \mathbf{v}_2); z) \end{pmatrix} \right\|_2 \leq \alpha \left\| \begin{pmatrix} \mathbf{w}_1 - \mathbf{w}_2 \\ \mathbf{v}_1 - \mathbf{v}_2 \end{pmatrix} \right\|_2. \quad (3)$$

Let $\mathbb{Q}$ be a probability measure over $[n]^T$. SGDA with sampling scheme $\mathbb{Q}$ updates $\mathbf{w}_t(\phi)$ and $\mathbf{v}_t(\phi)$ by

$$\begin{cases} \mathbf{v}_{t+1} = \mathbf{v}_t + \eta_t \nabla_{\mathbf{v}} \ell((\mathbf{w}_t, \mathbf{v}_t); z_{i_t}), \\ \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} \ell((\mathbf{w}_t, \mathbf{v}_t); z_{i_t}), \end{cases} \quad (4)$$

where at the $t$-th iteration, $z_{i_{t \in [n]}}$ is such that $i_t = \phi_t$ where $\phi \in [n]^T$ is drawn from $\mathbb{Q}$. For sampling distributions of indices, both $\mathbb{P}$ and $\mathbb{Q}$ are discrete distributions over $\Phi$, so their KL divergence is

$$\mathrm{KL}(\mathbb{Q} \| \mathbb{P}) := \sum_{\phi \in \Phi} \mathbb{Q}(\phi) \log \frac{\mathbb{Q}(\phi)}{\mathbb{P}(\phi)}.$$

We will assume throughout that $\mathrm{KL}(\mathbb{Q} \| \mathbb{P}) \in \tilde{O}(1)$ when quantifying the rate of convergence of the forthcoming bounds. With the choice of $\mathbb{P}$ taken as the uniform distribution, this will be sufficient to allow us to account for a small fraction of outliers in algorithmic applications.

**Theorem 4.4** (Generalization bounds for SGDA)**.** *Assume $\ell$ is $M$-bounded, convex-concave and $L$-Lipschitz. For any $\delta \in (0, 1)$ and uniform prior $\mathbb{P}$, with probability at least $1 - \delta$ over $S$, the following holds for SGDA with fixed $\eta$ and all posterior sampling distribution $\mathbb{Q}$ on $[n]^T$,*

$$\mathbb{E}_{\phi \sim \mathbb{Q}} [G(S, \phi)] \lesssim \Big(\mathrm{KL}(\mathbb{Q} \| \mathbb{P}) + \log \frac{1}{\delta}\Big) \max \left\{ L^2 \eta (\sqrt{T} + T/n) \log^2 n, \frac{M}{\sqrt{n}} \right\}.$$

*In addition, if $\ell$ is $\alpha$-smooth, we have*

$$\mathbb{E}_{\phi \sim \mathbb{Q}} \big[G(S, \phi)\big] \lesssim \big(\mathrm{KL}(\mathbb{Q} \| \mathbb{P}) + \log(1/\delta)\big) \max \left\{ L^2 \eta \exp(\alpha^2 t \eta^2) \Big( \frac{T}{n} + 1 + \sqrt{\frac{T}{n}} \Big) \log^2 n, \frac{M}{\sqrt{n}} \right\}.$$

Theorem 4.4 implies that choosing $T = O(n^2)$ and $\eta = O(T^{-\frac{3}{4}})$ gives nonvacuous results of the order $\widetilde{O}(\frac{1}{\sqrt{n}})$, based on the previous assumption that $\mathrm{KL}(\mathbb{Q} \| \mathbb{P}) \in \tilde{O}(1)$. In smooth cases, if we choose $T = O(n)$ and $\eta = O(\frac{1}{\sqrt{n}})$, this gives the bounds of the order $\widetilde{O}(\frac{1}{\sqrt{n}})$.

The key benefit of our PAC-Bayes analysis is that results hold with uniform sampling transfer to guarantee holding for all posterior sampling distributions.

Next, we give results for pairwise SGD.

**Pairwise SGD:** For a pairwise loss, we define $\mathcal{L}(\mathbf{w}; \cdot) := \ell(\mathbf{w}; \cdot, \cdot)$, where $\ell : \mathcal{W} \times (\mathcal{Z} \times \mathcal{Z}) \mapsto \mathbb{R}_+$ and the risk is

$$R(A(S; \phi)) = \mathbb{E}_{z, \tilde{z} \sim \mathcal{D}} [\ell(A(S; \phi), z, \tilde{z})]. \quad (5)$$

For a pairwise loss, the empirical risk is

$$R_S(A(S; \phi)) = \frac{1}{n(n-1)} \sum_{i,j \in [n] : i \neq j} \ell(A(S; \phi); z_i, z_j).$$

At the $t$-th iteration for pairwise SGD, a pair of sample indices $\phi_t = (i_t, j_t)$ is drawn from the set $\{(i_t, j_t) : i_t, j_t \in [n], i_t \neq j_t\}$ from $\mathbb{Q}$ over $([n] \times [n])^T$. The update rule is $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell(\mathbf{w}_t; z_{i_t}, z_{j_t})$. Recent work gives generalization bounds for pairwise SGD [Zhou et al., 2025].

**Theorem 4.5** (Paiwise SGD, [Zhou et al., 2025])**.** *Assume $\ell$ is $M$-bounded, convex and $L$-Lipschitz. For any $\delta \in (0,1)$ and uniform prior $\mathbb{P}$, the following bounds hold for pairwise SGD with fixed step sizes and all posterior sampling distribution $\mathbb{Q}$*

$$\mathbb{E}_{\phi\sim\mathbb{Q}}\left[G(S,\phi)\right] \lesssim \Big(\mathrm{KL}(\mathbb{Q}\|\mathbb{P})+\log\frac{1}{\delta}\Big)\max\left\{L^2\eta(\sqrt{T}+T/n)\log^2 n, \frac{M}{\sqrt{n}}\right\}.$$

*In addition, if $\ell$ is $\alpha$-smooth and $\eta \leq 2/\alpha$, we have*

$$\mathbb{E}_{\phi\sim\mathbb{Q}}\big[G(S,\phi)\big] \lesssim \big(\mathrm{KL}(\mathbb{Q}\|\mathbb{P})+\log\frac{1}{\delta}\big)\max\left\{L^2\eta\Big(\frac{T}{n}+1+\sqrt{\frac{T}{n}}\Big)\log^2 n, \frac{M}{\sqrt{n}}\right\}.$$

Table 1 summarizes the generalization bounds for SGDA and pairwise SGD under various assumptions considered in our paper and in recent work [Lei et al., 2021a, 2020, 2021b, Zhou et al., 2025], where $n$ is the sample size.

Next, we consider to develop learning algorithms based on the generalization bounds for these stochastic optimization methods.

### 4.1 OPTIMIZATION OF THE BOUNDS W.R.T. POSTERIOR $\mathbb{Q}$

Inspired by the r.h.s. of the PAC-Bayes generalization bound of Theorem 4.4, in this section we devise a new SGDA-Q algorithm that learns a sampling distribution (along with the model's parameters) from the data.

Recall that, at the $t$-th iteration, a sample index $\phi_t = \{i_t\}$ is randomly selected from $S$, giving a sequence of indices $\phi = (\phi_1, ..., \phi_T)$ and parameter $\nu$.

The PAC-Bayes posterior $\mathbb{Q}$ in our bounds was a distribution on the set for trajectories. Here we will denote by $q(S)$ the sampling distribution over $[n]$, which we refer to simply as $q$ for brevity, used to pick the next training point in the trajectory or training sequence. Consequently, the following objective function resembles the form of the r.h.s. of the bounds.

$$\mathcal{L}(q) = \sum_{i=1}^{n} q(i)\mathcal{L}\left(h; z_i\right) + \nu\cdot\mathrm{KL}(q\|p) + \lambda\Big(\sum_{i=1}^{n} q(i) - 1\Big), \tag{6}$$

where $h$ contains the model parameters, $p(i) = 1/n, \forall i \in [n]$. We minimize this objective w.r.t. $q$, that is to find $q$ that minimizes the expected empirical loss while staying close to the prior.

Depending on the choice of $\mathcal{L}$, minimizing Eq. (6) leads to our new algorithms. When $\mathcal{L} = \max_{\mathbf{v}}\ell\left((\mathbf{w},\mathbf{v});\cdot\right)$, the minimization of (6) is carried out by SGDA-Q. Here, the suffix Q signifies that we are learning a sampling distribution $q$ alongside the parameter values.

The pseudo code of the resulting algorithms, SGDA-Q and pairwise SGD-Q are given in Algorithm 1. The derivation for pairwise SGD-Q is also given in Appendix B for completeness.

**Algorithm 1** SGDA-Q/Pairwise SGD-Q

```
 1: Inputs: $S$, $\ell$, $\nu$
    Optimize : $\mathbf{w}$, $\mathbf{v}$, $q$
 2: $q \leftarrow$ uniform, $t \leftarrow 1$
 3: for $k = 0, \ldots$ do
 4:   repeat
 5:     Sample $\phi_t \sim q_t$
 6:     SGDA-Q:
 7:       $i_t \leftarrow \phi_t$
 8:       $\mathbf{v}_{t+1} = \mathbf{v}_t + \eta\nabla_{\mathbf{v}}\ell((\mathbf{w}_t, \mathbf{v}_t); z_{i_t})$;
 9:       $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\nabla_{\mathbf{w}}\ell((\mathbf{w}_t, \mathbf{v}_t); z_{i_t})$;
10:     Pairwise SGD-Q:
11:       $(i_t, j_t) \leftarrow \phi_t$
12:       $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\nabla_{\mathbf{w}}\ell(\mathbf{w}_t; z_{i_t}, z_{j_t})$;
13:     $t = t + 1$
14:   until $t > T_{iter}$
15:   Update $q$ by Eq. (7) for SGDA-Q / Eq. (15) for pairwise SGD-Q.
16: end for
17: return $\mathbf{w}$, $q$
```

The minimization yields the following alternating updates.

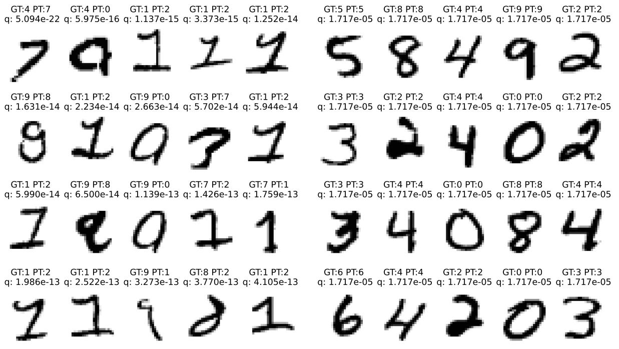
First, keeping $q$ fixed, we approximate the adversarial loss by taking one stochastic gradient ascent step w.r.t. $\mathbf{v}$, followed by taking a gradient descent step w.r.t. $\mathbf{w}$ (lines 7-8 in the pseudo code). These steps represent the vanilla SGDA updates.

Then, keeping $\mathbf{w}$ and $\mathbf{v}$ fixed, derive the update $q$ as follows. Observe that all terms depend on $q$; taking derivative w.r.t. each $q(i)$ and rearranging the stationary equation yields

$$q(i) = \frac{\exp\left(-\frac{1}{\nu}\mathcal{L}(h; z_i)\right)}{\sum_{j=1}^{n}\exp(-\frac{1}{\nu}\mathcal{L}(h; z_j))} \propto \exp\left(-\frac{1}{\nu}\mathcal{L}(h; z_i)\right). \tag{7}$$

This iterative approach updates the data-dependent posterior $q$ conditioned on the optimized parameters and the training sample. At the beginning, the sampling distribution is initialized with $q = p$.

The algorithm updates the sampling distribution conditioned on the training data. During training, an index is drawn from $q$. According to Eq. (7), the probability of selecting the $i$-th input is proportional to the exponential of the negative of its loss from the previous epoch, denoted as $q_t(i)$. This differs from AdaSamp [London, 2017] – an existing adaptive sampling algorithm inspired from PAC-Bayes bounds, where



(a) Top k smallest Q-scores

(b) Top k largest Q-scores

Figure 1: Examples with lowest and highest Q-scores as found by SGDA-Q in MNIST. 'PT' denotes the predicted label, 'GT' the ground truth, and 'q' the Q-scores.

the probability is proportional to the loss. As a comparison, for our algorithms, data points that have a large loss are less likely to be selected, and so potential outliers or noisy examples are automatically down-weighted.

Learning the sampling distribution balances the minimization of the expected empirical risk – which down-weights certain examples – and the minimization of divergence from a uniform prior – which weights all examples equally. Thus, the learned sampling distribution will only deviate from uniform sampling for a gain in the expected empirical risk. Next, we apply our algorithms to see the benefits of this trade-off.

## 5 EMPIRICAL EVALUATION

In this section, we empirically evaluate our algorithms, highlighting their ability to increase robustness and interpretability, in both standard training via pairwise SGD-Q and adversarial training via SGDA-Q. Results are presented across various architectures and datasets.

The visualizations in the pairwise setting inspire an interesting question about the differences between human and machine visual scene recognition [Bamber, 1969], which cannot be solved in the pointwise setting. These results further suggest the generalizability of our algorithms.

### 5.1 EXPERIMENTAL RESULTS

First, we introduce the datasets and architectures used in our experiments. We evaluate on MNIST and CIFAR-10 dataset. The parameter settings follow those in [Shah et al., 2020, Nouiehed et al., 2019, Chen et al., 2024]. **MNIST**: We use a four-layer neural network and train over 100 epochs using an initial learning rate of 0.001 for SGDA-Q with the decaying schedule of factor 5 after every 50 epochs. Our code is publicly available [1].

**Application 1: Estimating Example Difficulty**

Identifying challenging examples and estimating the level of difficulty of individual data points is crucial for detecting abnormal cases and samples needing further human evaluation. As discussed by Agarwal et al. [2022], methods that are able to do this have potential to improve the safe use of data, as well as model interpretability. We verify that our algorithms can identify difficult or atypical examples, often corresponding to blurry or noisy data. We evaluate this on the MNIST and CUHK03 datasets for pointwise and pairwise cases, ranking training-set data by Q-scores. The results are shown in Figure 1 and Figure 7.

In Figure 1 (a), we list the examples with lowest estimated values of $q$ and highest estimated values of $q$ in Figure 1 (b) for MNIST. We can see from Figure 1 and Figure 7 that high Q-score images typically have clear, uncluttered backgrounds and contain typical and well-visible objects, while low Q-score images are atypical, blurry or unclear, making object identification difficult. In pairwise tasks on similarity shown in Figure 7, low Q-score image pairs often show objects from unconventional angles, hindering the recognition of their similarity. Our algorithms effectively identify challenging or atypical examples by assigning them low Q-scores, which may be used to prompt human input or further review.

**Application 2: Training in the Presence of Label Noise**

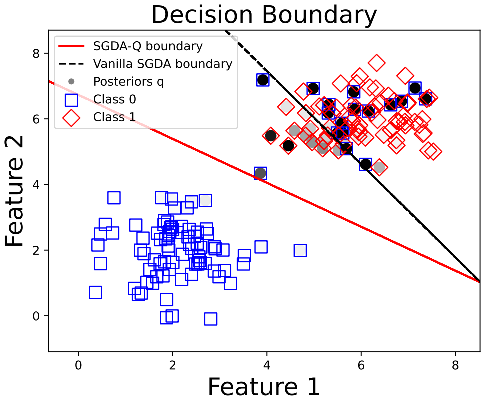


Figure 2: Decision boundaries obtained using SGDA-Q vs. vanilla SGDA on a simple 2D dataset with noisy labels. Dark filling means low Q-score.

We evaluate SGDA-Q here and pairwise SGD-Q in Section 5.2, in conditions of label noise to test their ability to identify and downweight the noisy examples and hence achieve robustness in the presence of out-of-distribution (OOD) samples that could bias estimates.

We first generated a 2D toy example with asymmetric label

[1] Code available at https://github.com/git0405/UAI-Learning-to-Sample-in-Stochastic-Optimization

| **Noise Rate** | **Alg.** | **Natural** | PGD$^{40}$ $L_\infty$ [Kurakin et al., 2016] | | | | FGSM $L_\infty$ [Goodfellow et al., 2014] | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Symmetric | | | $\epsilon = 0.1$ | $\epsilon = 0.2$ | $\epsilon = 0.3$ | $\epsilon = 0.4$ | $\epsilon = 0.1$ | $\epsilon = 0.2$ | $\epsilon = 0.3$ | $\epsilon = 0.4$ |
| Sym 0.2 | SGDA-Q | **99.26**% | **99.05**% | **98.69**% | **98.25**% | **97.68**% | **97.71**% | **94.65**% | **91.26**% | **86.02**% |
| | SGDA | 98.98% | 98.72% | 98.39% | 97.95% | 97.37% | 97.28% | 94.16% | 89.27% | 83.01% |
| | MART | 98.84% | 98.60% | 98.34% | 97.98% | 97.57% | 96.70% | 93.33% | 88.47% | 82.44% |
| | TRADES | 98.71% | 98.48% | 98.21% | 97.88% | 96.37% | 96.18% | 90.55% | 81.18% | 70.85% |
| Sym 0.4 | SGDA-Q | **99.12**% | **98.91**% | **98.53**% | **98.09**% | **97.47**% | **97.51**% | **94.37**% | **90.70**% | **85.37**% |
| | SGDA | 98.58% | 98.32% | 97.96% | 97.48% | 96.96% | 96.58% | 93.31% | 88.76% | 83.15% |
| | MART | 98.21% | 97.90% | 97.57% | 97.14% | 96.67% | 95.62% | 91.69% | 86.11% | 79.05% |
| | TRADES | 98.35% | 98.07% | 97.73% | 97.32% | 96.77% | 95.35% | 89.40% | 79.60% | 70.16% |
| Asymmetric | | | | | | | | | | |
| Asym 0.2 | SGDA-Q | **99.33** % | **99.06**% | **98.76**% | 98.29% | 97.72% | **98.02**% | 95.72% | 93.32% | **90.38**% |
| | SGDA | 99.26% | 99.00% | 98.66% | 98.19% | 97.57% | 97.98% | **95.97**% | **93.45**% | 90.37% |
| | MART | 99.28 % | 99.03% | 98.70% | 98.29% | 97.65% | 97.18% | 92.61% | 87.82% | 82.36% |
| | TRADES | 99.24% | 99.02% | 98.74% | **98.38**% | **97.89**% | 97.48% | 94.34% | 88.95% | 81.18% |
| Asym 0.4 | SGDA-Q | 98.88% | 98.52% | 98.07% | **97.50**% | 96.83% | **97.23**% | **94.98**% | **92.69**% | **89.60**% |
| | SGDA | 98.73% | 98.39% | 97.89% | 97.32% | 96.67% | 95.28% | 92.03% | 88.48% | 83.76% |
| | MART | **99.11**% | **98.79**% | **98.39**% | 97.45% | **97.40**% | 96.34% | 92.55% | 87.42% | 80.83% |
| | TRADES | 98.35% | 98.33% | 97.87% | 97.29% | 96.73% | 93.60% | 85.37% | 74.86% | 63.33% |

Table 2: Comparison of natural and adversarial accuracy under FGSM and PGD attacks with symmetric and asymmetric noise on four algorithms. The maximum results in each column are highlighted in bold font.

noise rate of 0.1 to illustrate the working of our algorithms. We ran logistic regression trained with SGDA and SGDA-Q for a comparison; the obtained decision boundaries are shown in Figure 2. The filling of markers reflects the Q-scores – darker means lower Q-score. In this comparison, we see that, in vanilla SGDA the decision boundary shifts due to the label noise. However, our algorithms demonstrate robustness to such noise. This is because our methods learn to down-weight and consequently avoid training on the misleading mislabeled points.

**Adversarial Training of SGDA-Q.**

Next, we evaluate the robustness of our SGDA-Q algorithm. Similarly to SGDA, this method is applicable to adversarial training. It aims to reduce the effect of test-time adversarial perturbations by training with a loss function that simulates adversarial examples. Indeed it is well known that addition certain imperceptible noises can fool the models into making wrong predictions [Goodfellow et al., 2014]. Adversarial training is an effective way to defend against these adversarial attacks.

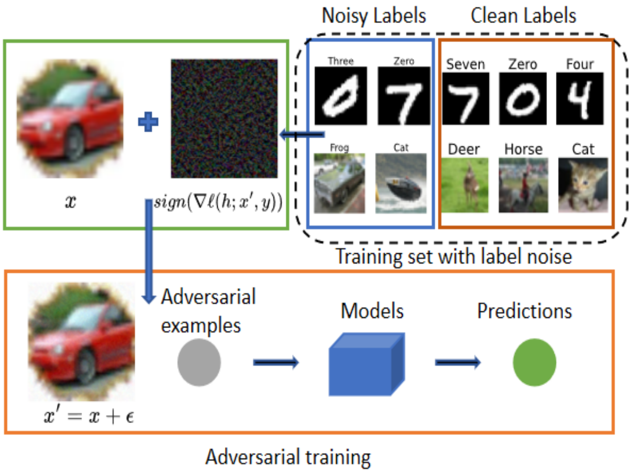


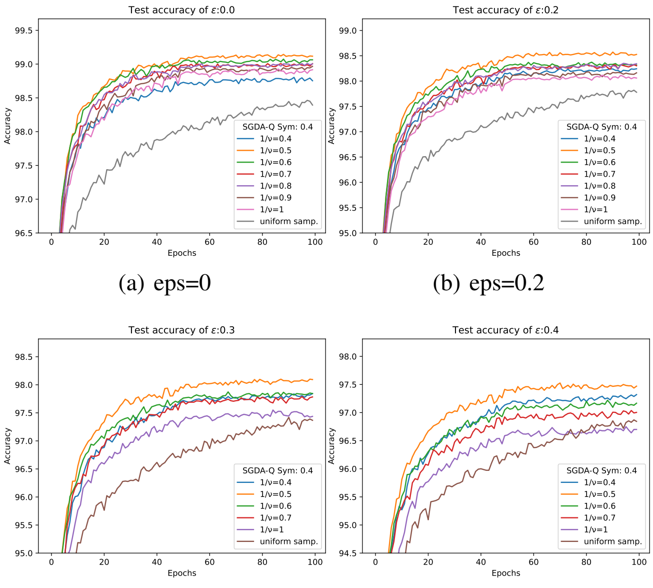
Figure 3: Adversarial training setting, with inherent noise.

Let us denote an adversarial example by $\mathbf{x}'$, obtained by adding an adversarial perturbation to a natural example $\mathbf{x}$. In the case of an $\ell_\infty$ adversarial attack, an adversarial example $\mathbf{x}'$ is chosen such that $\|\mathbf{x}' - \mathbf{x}\|_\infty \leq \epsilon$. Such perturbation is often imperceptible to humans but can cause the classifier $h$ to mispredict [Goodfellow et al., 2014]. In adversarial training, the goal is to guard against the ill effect of adversarial perturbations by solving the empirical adversarial risk minimization problem

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \max_{\|\mathbf{x}'_i - \mathbf{x}_i\|_\infty \leq \epsilon} \ell(h(\mathbf{x}'_i), y_i). \tag{8}$$

This is also a min-max problem, where $\mathbf{v}$ in our earlier general formulation is instantiated as $(\mathbf{x}'_1, \ldots, \mathbf{x}'_n)$.

For each training point, the maximum in the loss function searches for the worst-case perturbation of the input features,



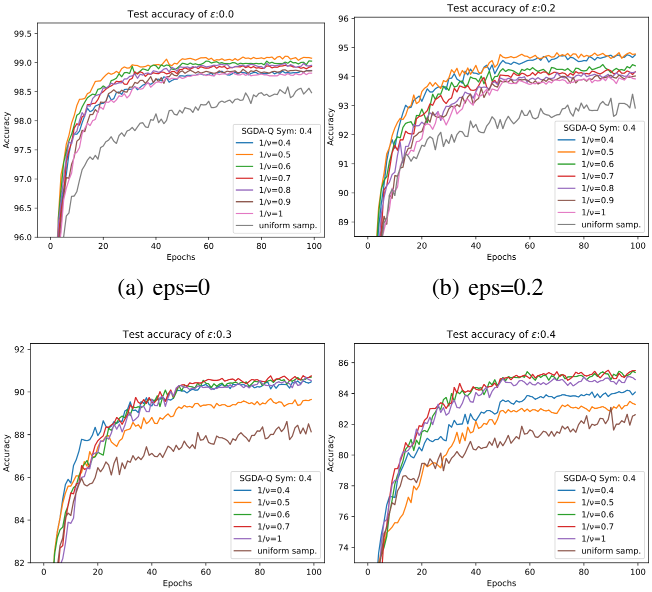
(a) eps=0

(b) eps=0.2

(c) eps=0.3

(d) eps=0.4

Figure 4: The impact of $1/\nu$ on test accuracy under PGD attack across different values of $\epsilon$ under a symmetric noise rate of 0.4.



(a) eps=0

(b) eps=0.2

(c) eps=0.3

(d) eps=0.4

Figure 5: The impact of $1/\nu$ on test accuracy under FGSM attack across different values of $\epsilon$ under a symmetric noise rate of 0.4.

while the outer minimization aims to reduce this worst-case value of the loss by adjusting the model parameters. In addition to this adversarial training min-max problem classically approached by SGDA, our SGDA-Q algorithm also adjusts the sampling probabilities to minimize the expected worst-case loss. This creates a fine balance between the adversarial training creating hard examples and our updates of $q$ potentially down-weighting them. Therefore, we expect that SGDA-Q is best suited when there are outliers or mislabeled points in the data set. Indeed, recent literature [Chen et al., 2024] reported that having both adversarial attacks and label noise is both realistic and challenging. We expect our method to identify and down-weight the noisy points while at the same time carrying out adversarial training.

We shall now examine our SGDA-Q algorithm in adversarial training in the presence of mislabeled samples to demonstrate the enhanced robustness of min-max stochastic optimization based learning. The problem setting is depicted in Figure 3.

Table 2 presents the results of our algorithms compared to vanilla SGDA (i.e. SGDA with uniform sampling), and two other state-of-the-art algorithms, namely MART [Wang et al., 2019], and TRADES [Zhang et al., 2019]. We use MNIST in the presence of random symmetric and asymmetric label noises at rates of 0.2 and 0.4, following the setting in [Chen et al., 2024, Shah et al., 2020]. For both SGDA and SGDA-Q, we use the cross-entropy loss along with the training protocol described in [Nouiehed et al., 2019]. Test accuracy was evaluated under FGSM [Goodfellow et al., 2014] and PGD-40 [Kurakin et al., 2016] attacks with results averaged over 3 independent runs. Based on the results in Table 2, our method achieves competitive accuracy performance, scoring best in most cases and at least second best in all cases tested.

We further investigate how $1/\nu$ affects the balance between the KL term and the expected empirical risk, influencing the posterior update and its impact on generalization. Figure 4 and Figure 5 show the effect of varying parameter $1/\nu$. We give the test accuracy in problems with FGSM and PGD attacks on clean data, across different values of $\epsilon$, in the presence of random symmetric label noise rate of 0.4. Results for other noise proportions are shown in the Appendix C. A grid search over $1/\nu \in (0, 3]$ reveals that smaller values of $1/\nu$ give better performance for smaller $\epsilon$, while larger $1/\nu$ achieves higher accuracy for larger $\epsilon$. We find that $1/\nu$ higher than 1 leads to decreased training performance.

To evaluate the robustness of our sampling strategy, we conduct additional experiments on the CIFAR-10 dataset. We use the loss function of TRADES, and only modify their algorithm by replacing uniform sampling with our adaptive sampling. We compare with the original TRADES (uniform sampling), as well as with MART, in the presence of label noise rates of 0.2 and 0.4, following the same setting as in [Chen et al., 2024]. We train ResNet-18 neural networks [He et al., 2016] for 200 epochs using an initial learning rate of 0.05, which decays by a factor of 10 at the 150th and 200th epochs. Table 3 presents both adversarial and

Table 3: Best and last accuracy (%) on CIFAR-10 with inherent symmetric and asymmetric label noise with 20% and 40% noise rate with PGD attack.

|  | Alg. | Sym0.2 |  | Sym0.4 |  | Asym0.2 |  | Asym0.4 |  |
|---|---|---|---|---|---|---|---|---|---|
|  |  | Natural | PGD-20 | Natural | PGD-20 | Natural | PGD-20 | Natural | PGD-20 |
| Best | Ours | 81.13 | **58.40** | 74.94 | 52.91 | **84.29** | **60.78** | **78.25** | **56.03** |
|  | MART | 78.96 | 48.01 | 74.97 | 45.82 | 83.04 | 54.19 | 76.85 | 47.97 |
|  | TRADES | **81.37** | 56.71 | **75.80** | **54.80** | 82.46 | 54.12 | 77.44 | 50.46 |
| Last | Ours | **74.94** | **40.73** | 59.12 | **26.38** | 80.37 | 47.43 | 71.68 | **40.67** |
|  | MART | 74.11 | 37.07 | 54.54 | 22.89 | **80.81** | 42.62 | **71.88** | 39.68 |
|  | TRADES | 74.76 | 39.74 | **60.23** | 26.12 | 77.71 | **49.31** | 70.87 | 40.62 |

natural accuracy results in the presence of label noise on the CIFAR-10 dataset. Here, “Best" means highest accuracy from across all epochs, and “Last” means the accuracy at the last training epoch. Our sampling strategy again achieves competitive performance, with higher test accuracy under PGD-20 attacks in more than half of the cases tested.

### 5.2 RESULTS FOR PAIRWISE SGD-Q

In this section, we test and demonstrate our pairwise SGD-Q algorithm. We consider a problem of similarity learning. Given an input pair, the goal is to predict if they belong to the same class or not.

**Architecture:** We employ the Siamese architecture depicted in Figure 6, built on the work of [Lv et al., 2018, Zheng et al., 2017]. It learns feature representations of the input pairs and their corresponding similarities. This framework consists of two modules to extract features from $(\mathbf{x}, \tilde{\mathbf{x}})$, both sharing the same weights.

The outputs of these two modules are flattened into one-dimensional feature vectors ($f_1$ and $f_2$ as shown in Figure 6). The element-wise squared difference between $f_1$ and $f_2$ is fed into a fully connected layer with a softmax, outputting the probability that the input pair belongs to the same class.

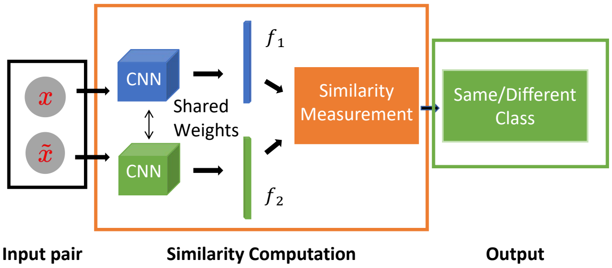


Figure 6: In the Siamese network architecture, when an input pair $(\mathbf{x}, \tilde{\mathbf{x}})$ is provided, two models with shared weights generate feature embeddings $f_1$ and $f_2$, which are then utilized to evaluate the similarity between the inputs.

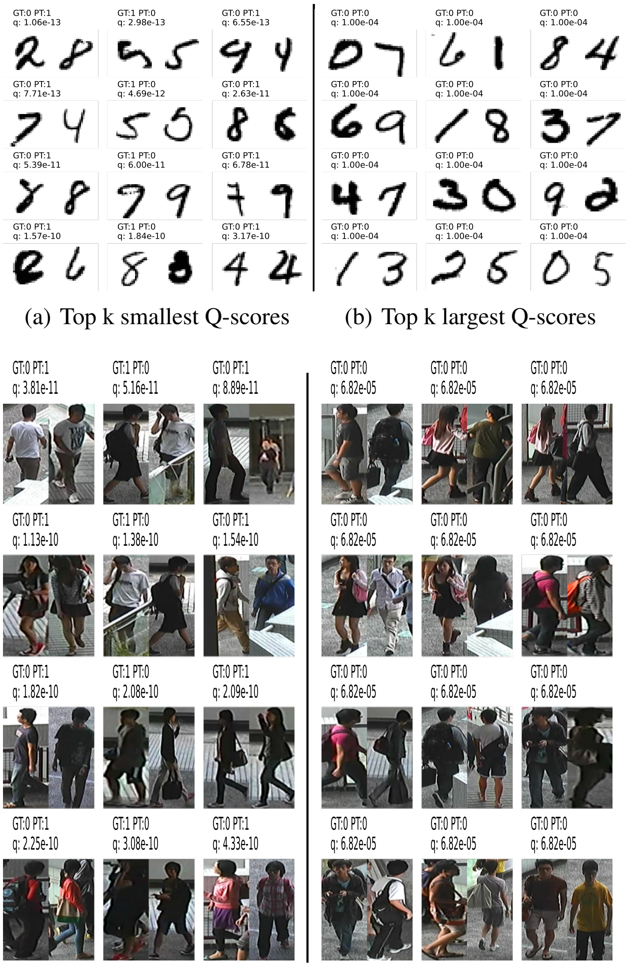
We tested two different base network modules with two datasets as follows. **MNIST**: We use a two-layer convolutional network as the CNN modules in Siamese networks and train over 100 epochs using an initial learning rate of 0.01 with the decaying schedule of factor 5 after every 30 epochs. **CUHK03 [Li et al., 2014]**: The CNN modules are based on the ResNet-18, pre-trained on the ImageNet dataset [Deng et al., 2009]. The model is trained for 65 epochs with an initial learning rate of 0.01, employing a decay schedule that decreases the learning rate by a factor of 5 every 20 epochs. The CUHK03 dataset comprises 14,097 images of 1,467 individuals.

In Figure 7 (a) and (c), we list the example pairs with lowest estimated values of $q$ for MNIST and CUHK03. In Figure 7 (b) and (d), we list the pairs with highest estimated values of $q$ in MNIST and CUHK03. An interesting observation from these figures is that low Q-score pairs tend to be same-label pairs whereas the high Q-score pairs tend to be different-label pairs. It seems to be an intriguing observation from these pairwise experiments that identifying two images as representing different content appears to be an easier problem than recognizing them as representing similar content. This suggests that machines (contrary to humans) may find it easier to detect differences than similarities, in the datasets tested.

**Training with Pairwise SGD-Q in the Presence of Label Noise.**

Next, we test our algorithms in accuracy comparison experiments. The results shown in Figure 8 are obtained in the presence of random label noise in a setup similar to [Shah et al., 2020]: 10% of the samples are randomly selected and assigned incorrect (opposite) labels. We report the accuracy measured on a clean independent test set, averaged over 5 independent repetitions.

We compare our algorithms with three methods, with vanilla SGD, AdaSamp [London, 2017] and MKL-SGD [Shah et al., 2020] – a Min-k Loss SGD that aims to improve robustness against outliers. MKL-SGD is an existing variant of SGD that previously demonstrated the robustness achieved by



(a) Top k smallest Q-scores (b) Top k largest Q-scores

(c) Top k smallest Q-scores (d) Top k largest Q-scores

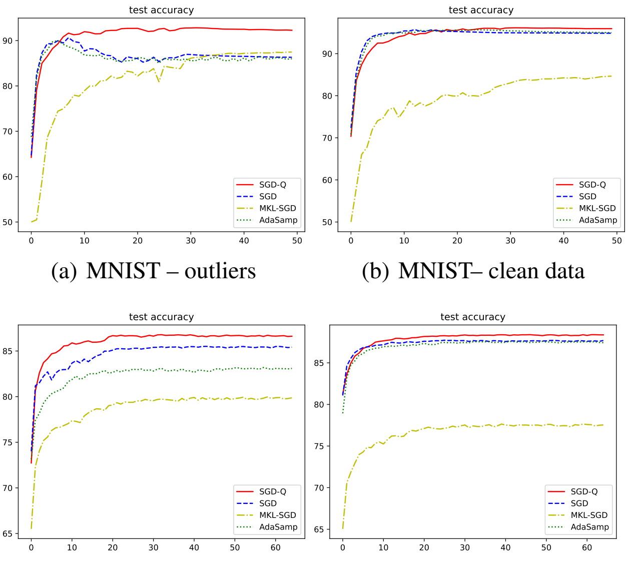
Figure 7: Pairwise: the top-k training-set pairs, with the lowest and highest Q-scores on dataset MNIST and CUHK03.

discarding high-loss examples, however, without any generalization guarantees. Recall that AdaSamp is an adaptive sampling algorithm inspired from PAC-Bayes bounds, with sampling probability proportional to the loss.

According to Figure 8, pairwise SGD-Q demonstrates superior test accuracy under label noise compared to both MKL-SGD and AdaSamp on MNIST and CUHK03, highlighting its robustness and enhanced generalization performance.

## 6 CONCLUSIONS

We considered a PAC-Bayes analysis of stochastic optimization algorithms, and based on this, learning the adaptive sampling scheme. We introduced new bounds-based algorithms that demonstrate strong robustness and offer insights into model behavior regarding example difficulty. Future research could explore the performance of these algorithms under different attacks and investigate their application with other optimization methods, such as randomized coordinate descent. It would also be interesting to follow up on our observations in pairwise learning in machines vs. humans.



(a) MNIST – outliers (b) MNIST– clean data

(c) CUHK03 – outliers (d) CUHK03 – clean data

Figure 8: Pairwise SGD-Q: Comparison of the test accuracy on MNIST and CUHK03 with and without outliers.

## Acknowledgements

The work of Sijia Zhou is funded by CSC and UoB scholarship. The work of Yunwen Lei is partially supported by the Research Grants Council of Hong Kong [Project No. 22303723]. Ata Kabán acknowledges past funding from EPSRC fellowship EP/P004245/1. The experiments were conducted using the University of Birmingham's Baskerville and BlueBEAR HPC services.

## References

Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2022.

Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119. PMLR, 2016.

Amiran Ambroladze, Emilio Parrado-Hernández, and John Shawe-Taylor. Tighter pac-bayes bounds. *Advances in Neural Information Processing Systems*, 19, 2006.

Donald Bamber. Reaction times and error rates for "same"-

"different" judgments of multidimensional stimull. *Perception & Psychophysics*, 6:169–174, 1969.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar): 499–526, 2002.

Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Learning Theory*, pages 610–626, 2020.

Zhen Chen, Fu Wang, Ronghui Mu, Peipei Xu, Xiaowei Huang, and Wenjie Ruan. Nrat: towards adversarial training with inherent label noise. *Machine Learning*, 113(6): 3589–3610, 2024.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International conference on machine learning*, pages 1125–1134. PMLR, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *33-rd Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2017.

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent pac-bayes priors via differential privacy. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in pac-bayes bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 604–612. PMLR, 2021.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Xumeng Han, Xuehui Yu, Guorong Li, Jian Zhao, Gang Pan, Qixiang Ye, Jianbin Jiao, and Zhenjun Han. Rethinking sampling strategies for unsupervised person reidentification. *IEEE Transactions on Image Processing*, 32:29–42, 2022.

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.

Tyler B Johnson and Carlos Guestrin. Training deep models faster with robust, approximate importance sampling. *Advances in Neural Information Processing Systems*, 31, 2018.

Angelos Katharopoulos and François Fleuret. Biased importance sampling for deep neural network training. *arXiv preprint arXiv:1706.00043*, 2017.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Yunwen Lei, Antoine Ledent, and Marius Kloft. Sharper generalization bounds for pairwise learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21236–21246, 2020.

Yunwen Lei, Mingrui Liu, and Yiming Ying. Generalization guarantee of sgd for pairwise learning. *Advances in Neural Information Processing Systems*, 34:21216–21228, 2021a.

Yunwen Lei, Zhenhuan Yang, Tianbao Yang, and Yiming Ying. Stability and generalization of stochastic gradient methods for minimax problems. In *International Conference on Machine Learning*, pages 6175–6186, 2021b.

Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person reidentification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 152–159, 2014.

Rui Liu, Tianyi Wu, and Barzan Mozafari. Adam with bandit sampling for deep learning. *Advances in Neural Information Processing Systems*, 33:5393–5404, 2020.

Ben London. A PAC-bayesian analysis of randomized learning with application to stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2931–2940, 2017.

Jianming Lv, Weihang Chen, Qing Li, and Can Yang. Unsupervised cross-dataset person re-identification by transfer learning of spatial-temporal patterns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7948–7956, 2018.

David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29, 2016.

Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. Pac-bayes bounds with data dependent priors. *The Journal of Machine Learning Research*, 13(1):3507–3531, 2012.

María Pérez-Ortiz, Omar Rivasplata, Benjamin Guedj, Matthew Gleeson, Jingyu Zhang, John Shawe-Taylor, Miroslaw Bober, and Josef Kittler. Learning pac-bayes priors for probabilistic neural networks. *arXiv preprint arXiv:2109.10304*, 2021a.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22, 2021b.

Omar Rivasplata, Emilio Parrado-Hernández, John S Shawe-Taylor, Shiliang Sun, and Csaba Szepesvári. Pac-bayes bounds for stable algorithms with instance-dependent priors. In *Advances in Neural Information Processing Systems*, pages 9214–9224, 2018.

Farnood Salehi, Patrick Thiran, and Elisa Celis. Coordinate descent with bandit sampling. *Advances in Neural Information Processing Systems*, 31, 2018.

Maziar Sanjabi, Jimmy Ba, Meisam Razaviyayn, and Jason D Lee. On the convergence and robustness of training gans with regularized optimal transport. *Advances in Neural Information Processing Systems*, 31, 2018.

Vatsal Shah, Xiaoxia Wu, and Sujay Sanghavi. Choosing the sample with lowest loss makes sgd robust. In *International Conference on Artificial Intelligence and Statistics*, pages 2120–2130. PMLR, 2020.

John Shawe-Taylor and Robert C Williamson. A pac analysis of a bayesian estimator. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 2–9, 1997.

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*, 2019.

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2840–2848, 2017.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

Junyu Zhang, Mingyi Hong, Mengdi Wang, and Shuzhong Zhang. Generalization bounds for stochastic saddle point problems. In *International Conference on Artificial Intelligence and Statistics*, pages 568–576. PMLR, 2021.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *International Conference on Machine Learning*, pages 1–9, 2015.

Zhedong Zheng, Liang Zheng, and Yi Yang. A discriminatively learned cnn embedding for person reidentification. *ACM transactions on multimedia computing, communications, and applications (TOMM)*, 14(1):1–20, 2017.

Sijia Zhou, Yunwen Lei, and Ata Kabán. Toward better pac-bayes bounds for uniformly stable algorithms. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Sijia Zhou, Yunwen Lei, and Ata Kabán. Randomized pairwise learning with adaptive sampling: A pac-bayes analysis. *arXiv preprint arXiv:2504.02957*, 2025.

# A APPENDIX: APPLICATIONS TO BOUNDING THE ERROR OF STOCHASTIC GRADIENT DESCENT ASCENT

We study SGDA for addressing min-max optimization problems in the convex-concave setting. We analyze SGDA under a general sampling scheme, where the random index is drawn from an arbitrary distribution.

**Definition A.1** (SGDA with general sampling)**.** *Let $\mathbf{w}_1$ and $\mathbf{v}_1$ denote the initial points, and $\nabla_{\mathbf{w}}\ell$ represent the gradient with respect to $\mathbf{w}$. Consider a probability measure $\mathbb{P}$ over $[n]^T$ and a training dataset $S = \{z_1, \ldots, z_n\}$. A sequence $(i_1, \ldots, i_T)$ is sampled according to $\mathbb{P}$. At the t-th iteration, SGDA with the sampling scheme $\mathbb{P}$ updates the model as follows:*

$$\begin{cases} \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} \ell((\mathbf{w}_t, \mathbf{v}_t); z_{i_t}), \\ \mathbf{v}_{t+1} = \mathbf{v}_t + \eta_t \nabla_{\mathbf{v}} \ell((\mathbf{w}_t, \mathbf{v}_t); z_{i_t}), \end{cases}$$

*where $\{\eta_t\}$ is a positive step-size sequence. If $\mathbb{P}$ is the uniform distribution, then we call it SGDA with uniform sampling (SGDAU).*

Next we give stability bounds to develop PAC-Bayes bounds for SGDA, covering both smooth and non-smooth cases. We first introduce sub-exponential stability [Zhou et al., 2023].

**Assumption A.2** (Sub-exponential stability)**.** *Let $\mathbb{P}$ be a fixed probability distribution. We say that a stochastic algorithm is sub-exponentially $\beta_\phi$-stable (w.r.t. $\mathbb{P}$) if, given any fixed instance of $\phi \sim \mathbb{P}$, it is $\beta_\phi$-uniformly stable, and there exist $b_1, b_2 \in \mathbb{R}$ such that for any $\delta \in (0, 1/n]$, the following holds with probability at least $1 - \delta$*

$$\beta_\phi \leq b_1 + b_2 \log(1/\delta). \tag{9}$$

## A.1 NON-SMOOTH CASE

The following lemma shows that SGDAU applied to non-smooth problems enjoys the sub-exponential stability. The proof is given in Appendix A.3.1.

**Lemma A.3** (Stability bound)**.** *Let $S$ and $S'$ be neighboring datasets. Suppose for all $z \in \mathcal{Z}$ the loss function is convex-concave and L-Lipschitz. Let $\{\mathbf{w}_t, \mathbf{v}_t\}, \{\mathbf{w}'_t, \mathbf{v}'_t\}$ be the sequence produced by SGDAU on $S$ and $S'$ respectively with fixed step sizes. Then SGDAU with $t$ iterations and the hyperparameter $\phi$ is $\beta_\phi$-uniformly stable with $\beta_\phi = 4\sqrt{e}L^2\eta\big(\sqrt{t} + \max_{k\in[n]} \sum_{j=1}^{t} \mathbb{I}[i_j = k]\big)$. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$ we have*

$$\beta_\phi \leq b_1 + 8\sqrt{e}L^2\eta(1 + \sqrt{t/n}) \log(1/\delta).$$

*That is, Assumption A.2 holds with $b_2 = 8\sqrt{e}L^2\eta(1 + \sqrt{t/n})$ w.r.t. $\mathbb{P}$.*

## A.2 SMOOTH CASE

In the following lemma to be proved in Appendix A.3.2, we give stability bounds for SGDA which satisfy Assumption A.2.

**Lemma A.4** (Stability bound)**.** *Let $S$ and $S'$ be neighboring datasets. Suppose for all $z \in \mathcal{Z}$ the loss function is convex-concave, $\alpha$-smooth and L-Lipschitz. Let $\{\mathbf{w}_t, \mathbf{v}_t\}, \{\mathbf{w}'_t, \mathbf{v}'_t\}$ be the sequence produced by SGDA on $S$ and $S'$ respectively with fixed step sizes. Then at $t$ iterations, SGDA with uniform sampling and the hyperparameter $\phi$ is $\beta_\phi$-uniformly stable with*

$$\beta_\phi = 4\sqrt{e}L^2\eta \exp(\frac{1}{2}\alpha^2\eta^2 t) \max_{k\in[n]} \left(1 + \sum_{r=1}^{t} \mathbb{I}[i_r = k]\right).$$

*If $\eta_t = \eta$, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ we have*

$$\beta_\phi \leq b_1 + 8\sqrt{e}L^2\eta \exp(\frac{1}{2}\alpha^2\eta^2 t)(1 + \sqrt{t/n}) \log(1/\delta).$$

*That is, Assumption A.2 holds with $b_2 = 8\sqrt{e}L^2\eta \exp(\frac{1}{2}\alpha^2\eta^2 t)(1 + \sqrt{t/n})$ w.r.t. $\mathbb{P}$.*

We can derive Theorem 4.4, based on the above lemmas. Next, we proof the above stability bounds for SGDA.

### A.3 PROOFS ON APPLICATIONS OF SGDA

**Lemma A.5** (Chernoff's Bound). *Let $X_1, \ldots, X_t$ be independent random variables taking values in $\{0,1\}$. Let $X = \sum_{k=1}^{t} X_k$ and $\mu = \mathbb{E}[X]$. Then for any $\epsilon > 0$ with probability at least $1 - \exp(-\mu\epsilon^2/(2+\epsilon))$ we have $X \leq (1+\epsilon)\mu$. Furthermore, for any $\delta \in (0,1)$ with probability at least $1-\delta$ we have*

$$X \leq \mu + \log(1/\delta) + \sqrt{2\mu \log(1/\delta)}.$$

First, we present the proofs on the generalization bounds for SGDA with smooth and non-smooth convex loss functions. Before that, we need to prove that SGDA meets Assumption A.2.

#### A.3.1 Non-smooth case

*Proof of Lemma A.3.* Without loss of generality, we first assume $S$ and $S'$ differ by the last example. According to the SGDA update rule and proof of Theorem 2(c) in [Lei et al., 2021b], for $p > 0$, we get

$$\left\| \begin{pmatrix} \mathbf{w}_{t+1} - \mathbf{w}'_{t+1} \\ \mathbf{v}_{t+1} - \mathbf{v}'_{t+1} \end{pmatrix} \right\|_2^2 \leq 8L^2\eta^2(1+p)^{\sum_{j=1}^{t} \mathbb{I}_{[i_j=n]}} \left( t + \sum_{k=1}^{t} \mathbb{I}_{[i_k=n]}/p \right).$$

We set $p = 1/\sum_{j=1}^{t} \mathbb{I}[i_j = n]$ and use the inequality $(1+1/x)^x \leq e$ to derive

$$\left\| \begin{pmatrix} \mathbf{w}_{t+1} - \mathbf{w}'_{t+1} \\ \mathbf{v}_{t+1} - \mathbf{v}'_{t+1} \end{pmatrix} \right\|_2^2 \leq 8eL^2\eta^2 \left( t + \Big( \sum_{k=1}^{t} \mathbb{I}[i_k = n] \Big)^2 \right).$$

It then follows that

$$\left\| \begin{pmatrix} \mathbf{w}_{t+1} - \mathbf{w}'_{t+1} \\ \mathbf{v}_{t+1} - \mathbf{v}'_{t+1} \end{pmatrix} \right\|_2 \leq \sqrt{8e}L\eta \left( \sqrt{t} + \sum_{k=1}^{t} \mathbb{I}[i_k = n] \right).$$

Based on Eq. (11), we further know that SGDA is $\beta_\phi$-uniformly stable with

$$\beta_\phi = 4\sqrt{e}L^2\eta\Big(\sqrt{t} + \max_{k\in[n]} \sum_{j=1}^{t} \mathbb{I}[i_j = k]\Big). \tag{10}$$

For simplicity, let $\beta_{\phi,k} = 4\sqrt{e}L^2\eta\big(\sqrt{t} + \sum_{j=1}^{t} \mathbb{I}[i_j = k]\big)$. Take the expectation of the above inequality, then gives the following bound

$$\mathbb{E}_{\phi\sim\mathbb{P}}[\beta_{\phi,k}] = 4\sqrt{e}L^2\eta\big(\sqrt{t} + t/n\big).$$

Applying Lemma A.5 to Eq. (10), with probability at least $1 - \delta/n$, we have

$$\beta_{\phi,k} \leq 4\sqrt{e}L^2\eta(\sqrt{t} + t/n + \log(n/\delta) + \sqrt{2t/n \log(n/\delta)}).$$

Therefore, with probability at least $1-\delta$, the following inequality holds simultaneously for all $k \in [n]$

$$\beta_{\phi,k} \leq 4\sqrt{e}L^2\eta(\sqrt{t} + t/n + \log(n/\delta) + \sqrt{2t/n \log(n/\delta)}),$$

which implies the following inequality with probability at least $1-\delta$

$$\beta_\phi \leq 4\sqrt{e}L^2\eta(\sqrt{t} + t/n + 2\log(1/\delta) + \sqrt{4t/n \log(1/\delta)}),$$

where we have used $\delta \in (0, 1/n)$ in the above inequality. Combining the stability bounds above, then we can prove that SGDAU with the hyperparameter $\phi$ meets the Assumption A.2 with

$$b_1 \geq 4\sqrt{e}L^2\eta(\sqrt{t} + t/n), \quad b_2 = 8\sqrt{e}L^2\eta(1 + \sqrt{t/n}).$$

The proof is completed. □

### A.3.2 Smooth case

*Proof of Lemma A.4.* Without loss of generality, we first assume $S$ and $S'$ differ by the last example. According to the SGDA update rule and proof of Theorem 2(d) in [Lei et al., 2021b], for $p > 0$ and fixed step sizes, we get

$$\begin{aligned}\left\|\begin{pmatrix} \mathbf{w}_{t+1}-\mathbf{w}'_{t+1} \\ \mathbf{v}_{t+1}-\mathbf{v}'_{t+1}\end{pmatrix}\right\|_2^2 \leq & 8(1+1/p)L^2\eta^2\prod_{j=1}^{t}\left(1+\alpha^2\eta_j^2\right)\prod_{j=1}^{t}(1+p)^{\mathbb{I}_{[i_j=n]}}\sum_{k=1}^{t}\mathbb{I}_{[i_k=n]} \\ \leq & 8(1+1/p)L^2\eta^2\exp\left(\alpha^2\sum_{j=1}^{t}\eta_j^2\right)(1+p)^{\sum_{j=1}^{t}\mathbb{I}_{[i_j=n]}}\sum_{k=1}^{t}\mathbb{I}_{[i_k=n]}.\end{aligned}$$

We set $p = 1/\sum_{j=1}^{t}\mathbb{I}[i_j = n]$ and use the inequality $(1+1/x)^x \leq e$ to derive

$$\left\|\begin{pmatrix} \mathbf{w}_{t+1}-\mathbf{w}'_{t+1} \\ \mathbf{v}_{t+1}-\mathbf{v}'_{t+1}\end{pmatrix}\right\|_2^2 \leq 8e\left(1+\sum_{k=1}^{t}\mathbb{I}_{[i_k=n]}\right)^2 L^2\eta^2\exp\left(\alpha^2\sum_{j=1}^{t}\eta_j^2\right).$$

Based on the Lipschitz continuity and above inequality, $\forall S \sim S' \in \mathcal{Z}^n, \forall z \in \mathcal{Z}$ we have the following, where we use the notation $(\mathbf{w}, \mathbf{v}) \equiv A_{\mathbf{w},\mathbf{v}}(S;\phi)$ and $(\mathbf{w}', \mathbf{v}') \equiv A_{\mathbf{w},\mathbf{v}}(S';\phi)$:

$$\begin{aligned}|\ell\left(A_{\mathrm{w},\mathrm{v}}(S;\phi),z\right)-\ell\left(A_{\mathrm{w},\mathrm{v}}(S';\phi),z\right)| &= |\ell((\mathbf{w},\mathbf{v});z)-\ell((\mathbf{w}',\mathbf{v}');z)| \\ &\leq |\ell((\mathbf{w},\mathbf{v});z)-\ell((\mathbf{w}',\mathbf{v});z)|+|\ell((\mathbf{w}',\mathbf{v});z)-\ell((\mathbf{w}',\mathbf{v}');z)| \leq L\left(\|\mathbf{w}-\mathbf{w}'\|_2+\|\mathbf{v}-\mathbf{v}'\|_2\right) \\ &\leq 4\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)\max_{k\in[n]}\left(1+\sum_{r=1}^{t}\mathbb{I}[i_r=k]\right). \qquad (11)\end{aligned}$$

Based on the above inequalities, we know that SGDA is $\beta_\phi$-uniformly stable with

$$\beta_\phi = 4\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)\max_{k\in[n]}\left(1+\sum_{r=1}^{t}\mathbb{I}[i_r=k]\right).$$

For simplicity, let $\beta_{\phi,k} = 4\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)\left(1+\sum_{j=1}^{t}\mathbb{I}[i_j=k]\right)$ for any $k \in [n]$. Taking the expectation over both sides of above inequality, we derive

$$\mathbb{E}_{\phi\sim\mathbb{P}}[\beta_\phi] \geq \mathbb{E}_{\phi\sim\mathbb{P}}[\beta_{\phi,k}] = 4\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)(1+t/n), \qquad (12)$$

where $\mathbb{E}[\mathbb{I}[i_j = k]] = 1/n$. Based on the above stability bounds, it remains to show that the stability parameter of SGDA meets Assumption A.2. According to Lemma A.5 with $X_j = \mathbb{I}[i_j = k]$, we get the following inequality with probability at least $1 - \delta/n$

$$\beta_{\phi,k} \leq 4\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)(1+t/n+\log(n/\delta)+\sqrt{2t/n\log(n/\delta)}). \qquad (13)$$

By the union of probability, with probability at least $1-\delta$, Eq. (13) holds for all $k \in [n]$. Therefore, with probability at least $1-\delta$

$$\begin{aligned}\beta_\phi &\leq 4\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)(1+t/n+\log(n/\delta)+\sqrt{2t/n\log(n/\delta)}) \\ &\leq 4\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)(1+t/n+2\log(1/\delta)+2\sqrt{t/n\log(1/\delta)}) \\ &\leq 4\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)(1+t/n)+8\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)(1+\sqrt{t/n})\log(1/\delta) \\ &\leq \mathbb{E}_{\phi\sim\mathbb{P}}[\beta_\phi]+8\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)(1+\sqrt{t/n})\log(1/\delta),\end{aligned}$$

where we have used $\delta \in (0, 1/n)$ in the second inequality, and Eq. (12) in the last inequality. Therefore, Assumption A.2 holds with $b_2 = 8\sqrt{e}L^2\eta\exp(\frac{1}{2}\alpha^2\eta^2 t)(1+\sqrt{t/n})$.

The proof is completed. □

Based on the above lemmas, we are ready to state generalization bounds in Corollary 4.4 for SGDA in non-smooth and smooth cases. We derive the generalization bounds for SGDA with general sampling based on the stability analysis for SGDA with uniform sampling.

*Proof of Corollary 4.4.* With $A(S;\phi) = (A_{\mathrm{w},\mathrm{v}}(S;\phi))$ , it is then clear that SGDA with convex-concave loss functions in both smooth and non-smooth cases satisfies Assumption A.2 based on Lemma A.3 and Lemma A.4. Therefore, the results are derived by applying the upper bound on $\beta_\phi$ to Theorem 1 in [Zhou et al., 2023]. □

# B ALGORITHM FOR PAIRWISE SGD

Inspired by the r.h.s. of PAC-Bayes generalization bound of Theorem 4.5, in this section we devise new pairwise SGD-Q algorithms that learn a sampling distribution (along with the model's parameters) from the data. The following objective function resembles the form of the r.h.s. of the bounds of pairwise SGD.

$$\mathcal{L}(q(i,j)) = \sum_{i,j\in[n]:i\neq j} q(i,j)\ell\left(h;z_i,z_j\right) + \nu\cdot \mathrm{KL}(q\|p) + \lambda\Big(\sum_{i,j\in[n]:i\neq j} q(i,j)-1\Big), \tag{14}$$

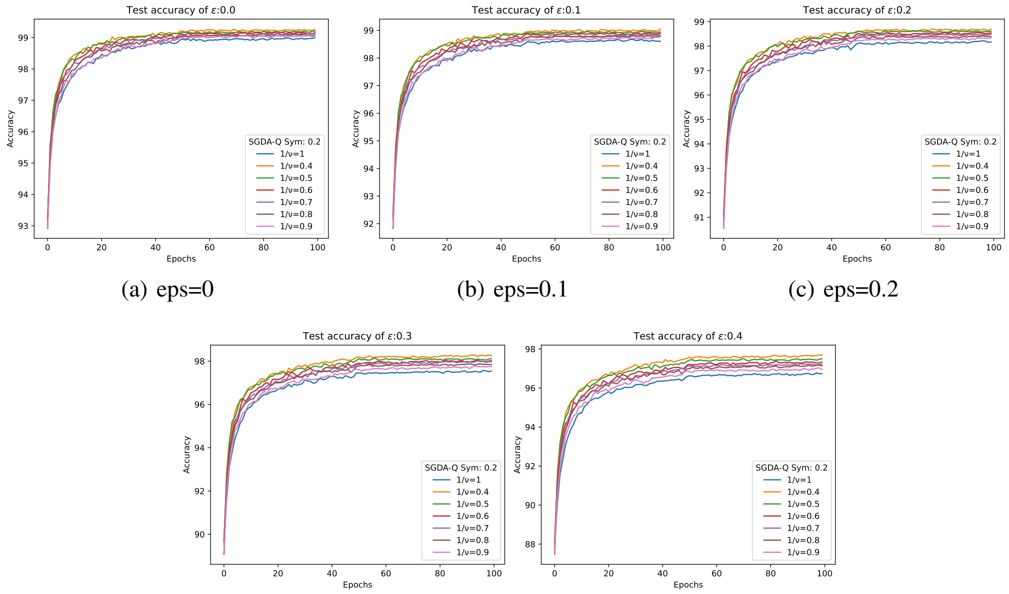
where $p(i,j) = \frac{1}{n(n-1)}, \forall i,j\in[n]: i\neq j$. We want to minimize this objective w.r.t. $q$, that is to find $q$ that minimizes the expected empirical loss while staying close to the prior.

All terms depend on $q$; taking derivative w.r.t. each $q(i,j)$, and rearranging the stationary equation yields the update for $q$:

$$q(i,j) = \frac{\exp\left(-\frac{1}{\nu}\ell(h;z_i,z_j)\right)}{\sum_{a,b\in[n]:a\neq b}\exp\left(-\frac{1}{\nu}\ell(h;z_a,z_b)\right)} \propto \exp\Big(-\frac{1}{\nu}\ell(h;z_i,z_j)\Big). \tag{15}$$

# C APPENDIX: ADDITIONAL EXPERIMENTS

Figures 9-12 provide results on the effect of $1/\nu$ in SGDA-Q for adversarial training, while varying the label noise proportions, and diameter of the adversarial perturbation $\epsilon$.



(a) eps=0

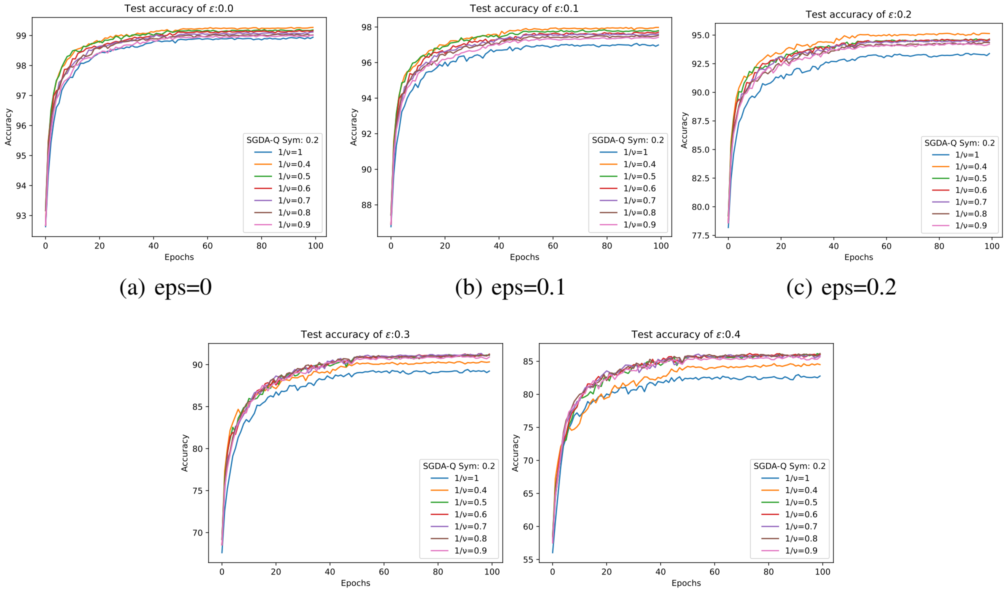
(b) eps=0.1

(c) eps=0.2

(d) eps=0.3

(e) eps=0.4

Figure 9: The impact of $1/\nu$ on test accuracy under PGD attack across different values of $\epsilon$ under symmetric noise rate 0.2.



(a) eps=0

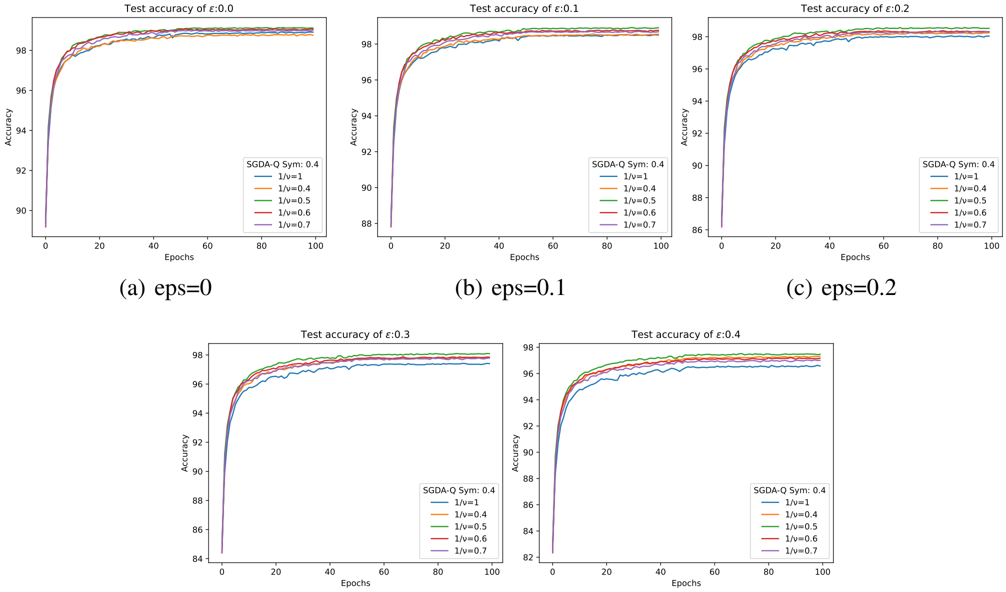
(b) eps=0.1

(c) eps=0.2

(d) eps=0.3

(e) eps=0.4

Figure 10: The impact of $1/\nu$ on test accuracy under FGSM attack across different values of $\epsilon$ under symmetric noise rate 0.2.



(a) eps=0

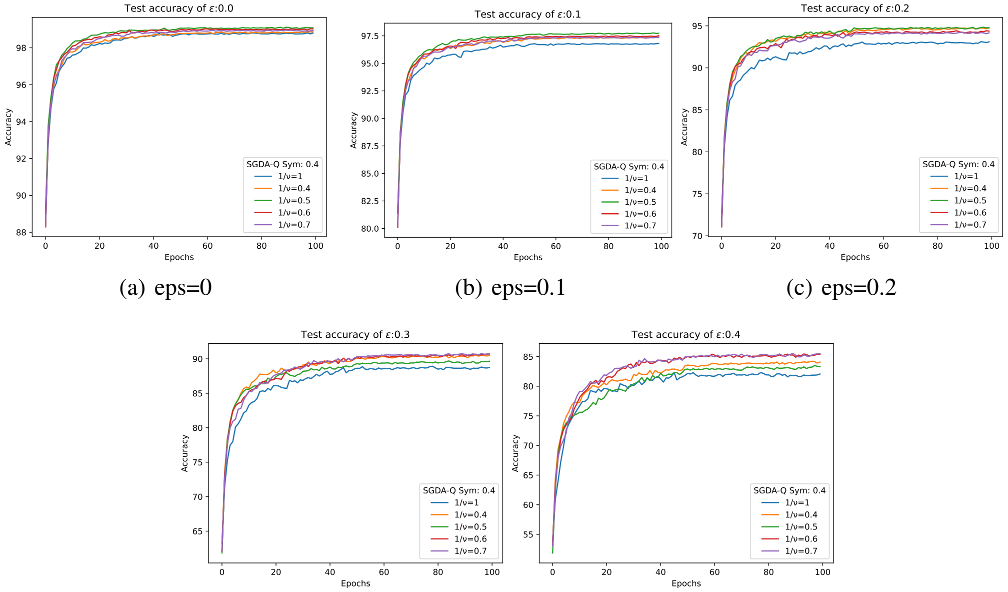
(b) eps=0.1

(c) eps=0.2

(d) eps=0.3

(e) eps=0.4

Figure 11: The impact of $1/\nu$ on test accuracy under PGD attack across different values of $\epsilon$ under symmetric noise rate 0.4.



(a) eps=0

(b) eps=0.1

(c) eps=0.2

(d) eps=0.3

(e) eps=0.4

Figure 12: The impact of $1/\nu$ on test accuracy under FGSM attack across different values of $\epsilon$ under symmetric noise rate 0.4.