Analyzing the Generalization Capability of SGLD Using Properties of Gaussian Channels

Hao Wang
Harvard University
hao_wang@g.harvard.edu

Yizhe Huang
The University of Texas at Austin
yizhehuang@utexas.edu

Rui Gao
The University of Texas at Austin
rui.gao@mccombs.utexas.edu

Flavio P. Calmon
Harvard University
flavio@seas.harvard.edu

Abstract

Optimization is a key component for training machine learning models and has a strong impact on their generalization. In this paper, we consider a particular optimization method—the stochastic gradient Langevin dynamics (SGLD) algorithm—and investigate the generalization of models trained by SGLD. We derive a new generalization bound by connecting SGLD with Gaussian channels found in information and communication theory. Our bound can be computed from the training data and incorporates the variance of gradients for quantifying a particular kind of "sharpness" of the loss landscape. We also consider a closely related algorithm with SGLD, namely differentially private SGD (DP-SGD). We prove that the generalization capability of DP-SGD can be amplified by iteration. Specifically, our bound can be sharpened by including a time-decaying factor if the DP-SGD algorithm outputs the last iterate while keeping other iterates hidden. This decay factor enables the contribution of early iterations to our bound to reduce with time and is established by strong data processing inequalities—a fundamental tool in information theory. We demonstrate our bound through numerical experiments, showing that it can predict the behavior of the true generalization gap.

1 Introduction

Modern deep neural networks (DNNs) are highly expressive: they can memorize an entire training dataset and still generalize well to unseen data (Zhang et al., 2016). This empirical observation is not captured by traditional generalization bounds found in statistical learning theory, which attribute the generalization ability to the use of a hypothesis class with constrained complexity (Vapnik and Chervonenkis, 1971; Valiant, 1984). Recent studies demonstrate that different algorithmic choices and data distributions may yield DNNs with contrasting generalization behaviors (Hardt et al., 2016; Neyshabur et al., 2017; Bartlett et al., 2017). In this paper, we study how one optimization method used for training DNNs, namely the stochastic gradient Langevin dynamics (SGLD) algorithm (Gelfand and Mitter, 1991; Welling and Teh, 2011), may influence their generalization.

The SGLD algorithm is used in different practical settings. For example, it has been implemented in open-source libraries (Facebook AI, 2020; Radebaugh and Erlingsson, 2019) for training models with differential privacy guarantees (Dwork et al., 2006; Song et al., 2013; Abadi et al., 2016). The additive noise in the SGLD algorithm can also mitigate overfitting for DNNs (Neelakantan et al., 2015). Recently, there is an increasing number of efforts (see e.g., Raginsky et al., 2017; Mou et al., 2018; Li et al., 2019; Pensia et al., 2018; Negrea et al., 2019) that investigate the generalization properties of the SGLD algorithm. It is within this body of work that the present paper is inscribed.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

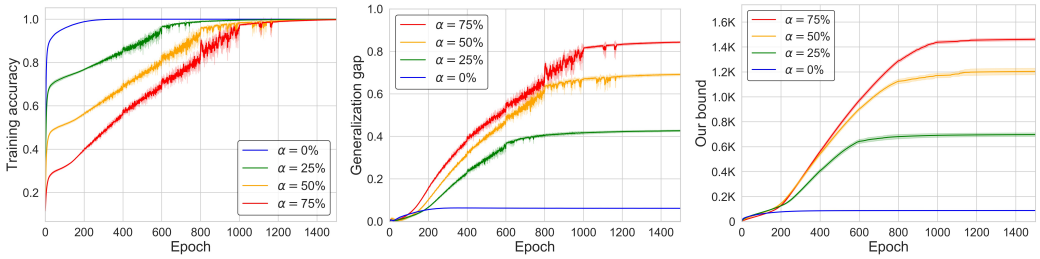

Figure 1: Illustration of our generalization bound in Theorem 1. We use the SGLD algorithm to train 3-layer neural networks on MNIST when the training data have different label corruption level $\alpha \in \{0\%, 25\%, 50\%, 75\%\}$. Left: training accuracy. Middle: (empirical) generalization gap. Right: (empirical) generalization bound. As shown, the generalization gap is increasing with respect to α and our bound can capture this phenomenon. We defer detailed discussions to Section 5.

We derive a new generalization bound (Theorem 1) for the SGLD algorithm in Section 3. Our bound (see Figure 1 for an illustration) incorporates the variance of gradients, which can be estimated from the training data and captures a particular kind of "sharpness" of the loss landscape (Keskar et al., 2016). This variance term can also be predictive of the generalization gap as shown in a recent empirical study (Jiang et al., 2019). We consider a general setting in which the output from the SGLD algorithm can be any function of the iterates. This is crucial since many theoretical analyses (see e.g., Zhang et al., 2017; Jin et al., 2019) require the final output to be the iterate that achieves the smallest value of the loss function or an average of the iterates, but not necessarily the last iterate. Finally, the numerical experiments in Section 5 suggest that our generalization bound is highly correlated with the true generalization gap.

We also investigate the DP-SGD algorithm in Section 4. In particular, we prove that if it is known a prior, that the algorithm outputs the last iterate rather than an arbitrary function of all iterates, then our bound can be further tightened by incorporating a time-decaying factor. Our analysis is motivated by a line of recent works (Feldman et al., 2018; Balle et al., 2019; Asoodeh et al., 2020) on *privacy amplification by iteration*. Specifically, the original work by Feldman et al. (2018) provided two intertwined observations of the DP-SGD algorithm: (i) not releasing the intermediate steps can amplify the privacy guarantees and (ii) data points used in the early iterations get stronger privacy protection than those occurring late. In this paper, we establish two analogous results: (i) our generalization bound can be sharpened by incorporating a time-decaying factor if DP-SGD only outputs the last iterate (Theorem 2) and (ii) this decay factor enables the impact of early iterations on our bound to reduce with time (Lemma 4).

The proof techniques of this paper are based on fundamental tools from information theory. We first use an information-theoretic framework, proposed by Russo and Zou (2016) and Xu and Raginsky (2017) and further tightened by Bu et al. (2020), for deriving an algorithmic generalization bound. This framework relates the generalization gap with the mutual information $I(\mathrm{W};\mathrm{Z}_i)$ between the output parameter W from the SGLD algorithm and each individual data point Z_i. However, estimating the mutual information from data is often intractable. Given this major challenge, our key contribution is to connect the SGLD algorithm with a well-understood notion in data transmission, namely additive white Gaussian noise (AWGN) channels. This connection allows us to use properties of Gaussian channels for analyzing the mutual information. First, we upper bound $I(\mathrm{W};\mathrm{Z}_i)$ using the variance of gradients by exploring the input-output mutual information of a Gaussian channel. This variance term can be estimated from the training data and is highly correlated with the true generalization gap. Second, we incorporate a time-decaying factor into our bound. This factor is established by strong data processing inequalities (Dobrushin, 1956; Cohen et al., 1998) and has an intuitive interpretation: if a data point is used at an early iteration, its impact on the generalization gap reduces with time due to external Gaussian noise. The above two aspects correspond to Lemma 4 and Lemma 5 which, in turn, are the basis of our main results in Theorem 1 and Theorem 2.

The supplementary material of this paper includes: (i) omitted proofs of all theoretical results and (ii) supporting experimental results.

Related Works

We contextualize our contributions in regard to existing literature (Mou et al., 2018; Li et al., 2019; Pensia et al., 2018; Bu et al., 2020; Negrea et al., 2019; Haghifam et al., 2020; Rodríguez-Gálvez et al., 2020; Neu et al., 2021). Among them, Mou et al. (2018) introduced two generalization bounds. The first one (Theorem 1 of Mou et al., 2018), a stability-based bound, achieves $O(1/n)$ rate in terms of the sample size n but relies on the Lipschitz constant of the loss function which makes it distribution-independent. A distribution-independent bound can be potentially loose and may not capture empirical observations (e.g., a network trained using true labels generalizes better than a network trained using corrupted labels as shown in Figure 1). The second one (Theorem 2 of Mou et al., 2018), a PAC-Bayes bound, replaces the Lipschitz constant by an expected-squared gradient norm but suffers from a slower rate $O(1/\sqrt{n})$. In contrast, our bound has order $O(1/n)$ and tightens the expected-squared gradient norm by the variance of gradients. The PAC-Bayes bound in Mou et al. (2018) is the only SGLD bound, besides ours, which incorporates an explicit time-decaying factor. However, their decay factor is incomparable with ours since they are established through distinct proof techniques and rely on different assumptions: their factor relies on an L_2 regularization[1] whereas ours is focused on the DP-SGD algorithm. A follow-up work by Li et al. (2019) combined the algorithmic stability approach with PAC-Bayesian theory and presented a bound which achieves order $O(1/n)$. However, their bound requires the scale of the learning rate to be upper bounded by the inverse Lipschitz constant of the loss function. In contrast, we do not need any assumptions on the learning rate.

There are significant recent works that adopt the information-theoretic framework (Xu and Raginsky, 2017) for bounding the SGLD generalization gap. Among them, Pensia et al. (2018) initially proposed a bound in Corollary 1 for analyzing a class of noisy iterative algorithms, including SGLD, and their bound was extended in Proposition 3 of Bu et al. (2020). However, the bounds in Pensia et al. (2018); Bu et al. (2020) are distribution-independent. Recently, Negrea et al. (2019) improved these bounds by replacing the Lipschitz constant with a gradient prediction residual, which is distribution-dependent. To compare with the generalization bound in Negrea et al. (2019), we incorporate a time-decaying factor into our bound under additional assumptions. Furthermore, the numerical experiments (Section 5) suggest that our bound is more favourably correlated with the true generalization gap. Negrea et al. (2019) provided another bound in their Theorem 3.1 which involves a variance term but it suffers from a slower rate $O(1/\sqrt{n})$.

More broadly, there are several recent works considering algorithms closely related to SGLD. For example, Haghifam et al. (2020) investigated the Langevin dynamics algorithm (i.e., full batch SGLD), which was later extended by Rodríguez-Gálvez et al. (2020) to SGLD, and observed a time-decaying phenomenon in their experiments. Specifically, Haghifam et al. (2020) incorporated a quantity, namely the squared error probability of the hypothesis test, into their bound in Theorem 4.2 and this quantity decays with the number of iterations. This seems to suggest that earlier iterations have a larger impact on their generalization bound. In contrast, our decay factor and the counterpart in Mou et al. (2018) indicate that the impact of earlier iterations is reducing with the total number of iterations. A recent work by Neu et al. (2021) investigated the generalization properties of SGD and provided a bound that also involves the variance of gradients. Note that the generalization bound in their Proposition 3 suffers from a weaker order $O(1/\sqrt{n})$ when the analysis is applied to SGLD.

To summarize, our main contribution is to provide a generalization bound for SGLD that incorporates the variance of gradients for quantifying the "sharpness" of the loss landscape and has $O(1/n)$ sample size dependence. Moreover, our bound holds under mild conditions and is applicable to a general setting in which the output from SGLD can be any function of the iterations. Under additional assumptions that are satisfied by DP-SGD, our bound includes an explicit time-decaying factor. Although each of the above contributions has been discussed in existing works, to the best of our knowledge, our bound is the first one that simultaneously combines all these aspects. Moreover, our proof techniques based on information-theoretic tools (e.g., strong data processing inequalities and properties of Gaussian channels) may be of broader interest to the community studying generalization theory.

[1] We note that Mou et al. (2018) also provided a bound in Theorem 23 without requiring regularization but the rate of the decay factor is slower than our factor in this case.

2 Preliminaries

Consider the following (possibly non-convex) optimization problem:

$$\min_{\boldsymbol{w}\in\mathcal{W}} L_\mu(\boldsymbol{w}) \triangleq \mathbb{E}\left[\ell(\boldsymbol{w},\mathrm{Z})\right] = \int_{\mathcal{Z}} \ell(\boldsymbol{w},\boldsymbol{z})\mathrm{d}\mu(\boldsymbol{z}),$$

where $\boldsymbol{w} \in \mathcal{W} \subseteq \mathbb{R}^d$ is the parameter (e.g., weights of a neural network) to optimize; μ is the underlying data distribution that generates Z; and $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}^+$ is the loss function (e.g., 0-1 loss). Since μ is unknown, $L_\mu(\boldsymbol{w})$ cannot be computed directly. Hence, one can instead minimize the empirical risk using a dataset $\mathrm{S} \triangleq (\mathrm{Z}_1, \cdots, \mathrm{Z}_n)$ which contains n i.i.d. points $\mathrm{Z}_i \sim \mu$:

$$\min_{\boldsymbol{w}\in\mathcal{W}} L_\mathrm{S}(\boldsymbol{w}) \triangleq \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{w},\mathrm{Z}_i).$$

Stochastic gradient Langevin dynamics. We consider the stochastic gradient Langevin dynamics (SGLD) algorithm (Gelfand and Mitter, 1991; Welling and Teh, 2011) for solving this empirical risk optimization. The dataset S is first divided into m disjoint mini-batches:

$$\mathrm{S} = \bigcup_{j=1}^{m} \mathrm{S}_j, \quad \text{where } |\mathrm{S}_j| = b \text{ and } \mathrm{S}_j \cap \mathrm{S}_k = \emptyset \text{ for } j \neq k.$$

We initialize the parameter with a random point $\mathrm{W}_0 \in \mathcal{W}$ and update using the following rule:

$$\mathrm{W}_t = \mathrm{W}_{t-1} - \eta_t \nabla_{\boldsymbol{w}} \hat{\ell}(\mathrm{W}_{t-1}, \mathrm{S}_{B_t}) + \sqrt{\frac{2\eta_t}{\beta_t}}\mathrm{N}, \tag{1}$$

where η_t is the learning rate; β_t is the inverse temperature; $\mathrm{N} \sim N(0, \mathbf{I}_d)$ is a random variable drawn independently from a standard Gaussian distribution; $B_t \in [m]$ is the mini-batch index[2]; $\hat{\ell}$ is a surrogate loss (e.g., hinge loss); and

$$\nabla_{\boldsymbol{w}} \hat{\ell}(\mathrm{W}_{t-1}, \mathrm{S}_{B_t}) \triangleq \frac{1}{b}\sum_{\mathrm{Z}\in\mathrm{S}_{B_t}} \nabla_{\boldsymbol{w}} \hat{\ell}(\mathrm{W}_{t-1}, \mathrm{Z}). \tag{2}$$

The recursion in (1) runs for T iterations and the final output is $\mathrm{W} = f(\mathrm{W}_1, \cdots, \mathrm{W}_T)$ which is a function of the parameters across all iterations. For example, the output can be the parameter at the last iteration $\mathrm{W} = \mathrm{W}_T$, the one which achieves the smallest value of the loss function $\mathrm{W} = \mathrm{argmin}_{\mathrm{W}_t} L_\mu(\mathrm{W}_t)$, or an average of the parameters (i.e., Polyak averaging) $\mathrm{W} = \frac{1}{T}\sum_t \mathrm{W}_t$.

Information-theoretic generalization bounds. The goal of this paper is to derive an upper bound for the *expected generalization gap*:

$$\mathbb{E}\left[L_\mu(\mathrm{W}) - L_\mathrm{S}(\mathrm{W})\right]. \tag{3}$$

A recent work by Xu and Raginsky (2017) provided a new method for bounding the expected generalization gap in terms of the mutual information between the input dataset S and the output parameter W. The bound in Xu and Raginsky (2017) was later tightened by Bu et al. (2020).

Lemma 1 (Bu et al. (2020) Proposition 1)**.** *Let the loss function $\ell(\boldsymbol{w},\mathrm{Z})$ be σ-sub-Gaussian under $\mathrm{Z} \sim \mu$ for all $\boldsymbol{w} \in \mathcal{W}$. For any learning algorithm which takes a dataset $\mathrm{S} = (\mathrm{Z}_1, \cdots, \mathrm{Z}_n)$ as input and outputs* W*,*

$$\left|\mathbb{E}\left[L_\mu(\mathrm{W}) - L_\mathrm{S}(\mathrm{W})\right]\right| \leq \frac{1}{n}\sum_{i=1}^{n} \sqrt{2\sigma^2 I(\mathrm{W};\mathrm{Z}_i)}, \tag{4}$$

where $I(\mathrm{W};\mathrm{Z}_i)$ is the mutual information between the learning algorithm's output W *and an individual data point Z_i.*

[2]For the sake of illustration, we assume that the mini-batch indices are specified before the SGLD is run.

Strong data processing inequalities. In order to characterize the time-decaying phenomenon, we use an information-theoretic tool: strong data processing inequalities (Dobrushin, 1956; Cohen et al., 1998). The data processing inequality (Cover and Thomas, 2012) states that if a Markov chain $\mathrm{U} \to \mathrm{X} \to \mathrm{Y}$ holds, then $I(\mathrm{U};\mathrm{Y}) \le I(\mathrm{U};\mathrm{X})$. In other words, no post-processing of X can increase the information about U. Under certain conditions, the data processing inequality can be sharpened, which leads to a strong data processing inequality, often cast in terms of a contraction coefficient. Next, we recall the contraction coefficients of f-divergences and show their connection with strong data processing inequalities.

Let $f : (0,\infty) \to \mathbb{R}$ be a convex function with $f(1) = 0$ and P, Q be two probability distributions over a set $\mathcal{X} \subseteq \mathbb{R}^d$. The f-divergence (Csiszár, 1967) between P and Q is defined as

$$\mathrm{D}_f(P\|Q) \triangleq \int_{\mathcal{X}} f\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right) \mathrm{d}Q. \tag{5}$$

Examples of f-divergence include KL-divergence ($f(t) = t\log t$) and total variation distance ($f(t) = |t-1|/2$). For a given transition probability kernel $P_{\mathrm{Y}|\mathrm{X}} : \mathcal{X} \to \mathcal{Y}$, let $P_{\mathrm{Y}|\mathrm{X}} \circ P$ be the distribution on $\mathcal{Y}$ induced by the push-forward of the distribution P (i.e., the distribution of Y when the distribution of X is P). The contraction coefficient of $P_{\mathrm{Y}|\mathrm{X}}$ for D_f is defined as

$$\eta_f(P_{\mathrm{Y}|\mathrm{X}}) \triangleq \sup_{P,Q:P\neq Q} \frac{\mathrm{D}_f(P_{\mathrm{Y}|\mathrm{X}} \circ P\|P_{\mathrm{Y}|\mathrm{X}} \circ Q)}{\mathrm{D}_f(P\|Q)} \in [0,1].$$

In particular, when the total variation distance is used, the corresponding contraction coefficient $\eta_{\mathrm{TV}}(P_{\mathrm{Y}|\mathrm{X}})$ is known as the Dobrushin's coefficient (Dobrushin, 1956), which upper bounds all other contraction coefficients (Cohen et al., 1998): $\eta_f(P_{\mathrm{Y}|\mathrm{X}}) \le \eta_{\mathrm{TV}}(P_{\mathrm{Y}|\mathrm{X}})$. Furthermore, for any Markov chain $\mathrm{U} \to \mathrm{X} \to \mathrm{Y}$, the contraction coefficient of KL-divergence satisfies (Ahlswede and Gács, 1976)

$$I(\mathrm{U};\mathrm{Y}) \le \eta_{\mathrm{KL}}(P_{\mathrm{Y}|\mathrm{X}}) \cdot I(\mathrm{U};\mathrm{X}). \tag{6}$$

When $\eta_{\mathrm{KL}}(P_{\mathrm{Y}|\mathrm{X}}) < 1$, the strict inequality $I(\mathrm{U};\mathrm{Y}) < I(\mathrm{U};\mathrm{X})$ improves the data processing inequality and, hence, is referred to as a strong data processing inequality. We refer the reader to Polyanskiy and Wu (2016) and Raginsky (2016) for a more comprehensive review on strong data processing inequalities and Calmon et al. (2017) for non-linear strong data processing inequalities in Gaussian channels.

Gaussian channels. We describe next a few fundamental properties of Gaussian channels. They will be used to derive a closed-form expression of the decay factor and to upper bound the mutual information by a quantity that can be estimated from data.

Consider a pair of random variables (X,Y) related by $\mathrm{Y} = \mathrm{X} + m\mathrm{N}$ where X is lying on $\mathcal{X}$; $m > 0$ is a constant; and $\mathrm{N} \sim N(0, \mathbf{I}_d)$ follows a standard Gaussian distribution. This model can be regarded as a single use of a Gaussian channel, which has a long history in information theory and possesses many interesting properties. For example, if $\mathcal{X}$ is a compact set, the contraction coefficients have a non-trivial upper bound

$$\eta_{\mathrm{KL}}(P_{\mathrm{Y}|\mathrm{X}}) \le \eta_{\mathrm{TV}}(P_{\mathrm{Y}|\mathrm{X}}) = 1 - 2\bar{\Phi}\left(\frac{\mathsf{diam}(\mathcal{X})}{2m}\right), \tag{7}$$

where $\mathsf{diam}(\mathcal{X}) \triangleq \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}} \|\boldsymbol{x}-\boldsymbol{x}'\|_2$ is the diameter of $\mathcal{X}$ and $\bar{\Phi}(t) \triangleq \int_t^\infty \frac{1}{\sqrt{2\pi}} \exp(-v^2/2)\mathrm{d}v$ is the Gaussian complementary cumulative distribution function (CCDF). Another useful property is the following inequality (see Lemma 3.4.2 in Raginsky and Sason, 2012, for a proof) which upper bounds the KL-divergence of the output distributions from the Gaussian channel by the Wasserstein distance of their input distributions. It also serves as a fundamental lemma for proving Otto-Villani's HWI inequality (Otto and Villani, 2000) in the Gaussian case.

Lemma 2. *Let* X *and* X' *be a pair of random variables which are independent of* $\mathrm{N} \sim N(0,\mathbf{I}_d)$. *Then for any* $m > 0$

$$\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{X}+m\mathrm{N}}\|P_{\mathrm{X}'+m\mathrm{N}}) \le \frac{1}{2m^2}\mathbb{W}_2^2(P_{\mathrm{X}}, P_{\mathrm{X}'}). \tag{8}$$

Here $\mathbb{W}_2(P_{\mathrm{X}}, P_{\mathrm{X}'})$ is the 2-Wasserstein distance equipped with the L_2 cost function:

$$\mathbb{W}_2^2(P_{\mathrm{X}}, P_{\mathrm{X}'}) \triangleq \inf \mathbb{E}\left[\|\mathrm{X} - \mathrm{X}'\|_2^2\right],$$

where the infimum is taken over all couplings (i.e., joint distributions) of the random variables X *and* X' *with marginals P_{X} and $P_{\mathrm{X}'}$, respectively.*

We recall an analogous result (Guo et al., 2005) which is also used in our proof. It gives an upper bound for the input-output mutual information of a Gaussian channel.

Lemma 3. *Let* X *be a random variable which is independent of* $\mathrm{N} \sim N(0, \mathbf{I}_d)$. *Then for any $m > 0$*

$$I(\mathrm{X} + m\mathrm{N}; \mathrm{X}) \leq \frac{1}{2m^2}\mathsf{Var}\left(\mathrm{X}\right). \tag{9}$$

3 Generalization Bounds for SGLD

Although Lemma 1 provides a generalization bound for any learning algorithm, estimating the mutual information from data is often difficult. In this section, we further upper bound the mutual information for the SGLD algorithm by using properties of Gaussian channels, discussed in the last section. This effort leads to a generalization bound (Theorem 1) which can be estimated from the training set. We observe in our experiments (Section 5) that Theorem 1 captures some generalization phenomena of DNNs, such as label corruption, and is highly correlated with the true generalization gap. We end this section by comparing our bound with existing SGLD generalization bounds (Corollary 1) and extending our analysis to a high-probability bound (Proposition 1).

Before diving into the analysis, we first discuss the main assumption made in this paper.

Assumption 1. *The loss function $\ell(\boldsymbol{w}, \mathrm{Z})$ is σ-sub-Gaussian under $\mathrm{Z} \sim \mu$ for all $\boldsymbol{w} \in \mathcal{W}$.*

We impose this assumption in order to apply Lemma 1. In particular, if the loss function is bounded between two constants a and b, this assumption is naturally satisfied with sub-Gaussian constant $\sigma = (b-a)/2$.

Now we present the main result in this section—a generalization bound for the SGLD algorithm. Its proof relies on the chain rule of mutual information and properties of Gaussian channels (Lemma 3). As a reminder, the output from SGLD is allowed to be any function of the iterates (i.e., $\mathrm{W} = f(\mathrm{W}_1, \cdots, \mathrm{W}_T)$).

Theorem 1. *Under Assumption 1, the expected generalization gap of the SGLD algorithm has the following upper bound:*

$$\mathbb{E}\left[L_\mu(\mathrm{W}) - L_{\mathrm{S}}(\mathrm{W})\right] \leq \frac{\sqrt{2b}\sigma}{2n}\sum_{j=1}^{m}\sqrt{\sum_{t\in\mathcal{T}_j}\beta_t\eta_t \cdot \mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1}, \mathrm{S}_j)\right)}, \tag{10}$$

where the set $\mathcal{T}_j$ contains the indices of iterations in which the mini-batch S_j is used and

$$\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1}, \mathrm{S}_j)\right) \triangleq \mathbb{E}\left[\|\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1}, \mathrm{S}_j) - \boldsymbol{e}\|_2^2\right]$$

with the vector $\boldsymbol{e} \triangleq \mathbb{E}\left[\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1}, \mathrm{S}_j)\right]$.

Remark 1. Our proof techniques can be extended to analyze a broad class of noisy iterative algorithms besides SGLD. For example, if the probability distribution of the additive noise N is log-Lipschitz (i.e., the logarithmic probability density function (pdf) is Lipschitz) as considered in Li et al. (2019), one can derive an analogous generalization bound by adapting our proof. In contrast, some generalization bounds (e.g., Mou et al., 2018) seem to heavily rely on the Gaussian noise.

The variance of gradients in (10) measures a particular kind of "sharpness" of the loss landscape. A recent work (Section 4.4 in Jiang et al., 2019) has observed empirically that this quantity is predictive of and highly correlated with the true generalization gap. Here we evidence this connection from a theoretical viewpoint by incorporating the gradient variance into the generalization bound.

Many existing SGLD generalization bounds (e.g., Mou et al., 2018; Li et al., 2019; Pensia et al., 2018; Negrea et al., 2019) are expressed as a sum of errors associated with each training iteration. In order to compare with these results, we present an analogous bound in the following corollary. This bound is obtained by combining a key lemma for proving Theorem 1 with Minkowski inequality and Jensen's inequality so it is often much weaker than Theorem 1.

Corollary 1. *Under Assumption 1, the expected generalization gap (3) of the SGLD algorithm can be upper bounded by*

$$\frac{\sqrt{2}\sigma}{2}\min\left\{\frac{1}{n}\sum_{t=1}^{T}\sqrt{\beta_t\eta_t\cdot\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1},\mathrm{Z}_t^{\dagger})\right)},\sqrt{\frac{1}{bn}\sum_{t=1}^{T}\beta_t\eta_t\cdot\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1},\mathrm{Z}_t^{\dagger})\right)}\right\},$$

where $\mathrm{Z}_t^{\dagger}$ is any data point used in the t-th iteration.

Our bound is distribution-dependent through the variance of gradients in contrast with Corollary 1 of Pensia et al. (2018), Proposition 3 of Bu et al. (2020), and Theorem 1 of Mou et al. (2018), which rely on the Lipschitz constant: $\sup_{\boldsymbol{w},\boldsymbol{z}}\|\nabla_{\boldsymbol{w}}\hat{\ell}(\boldsymbol{w},\boldsymbol{z})\|_2$. These bounds fail to explain some generalization phenomena of DNNs, such as label corruption (Zhang et al., 2016), because the Lipschitz constant takes a supremum over all possible weight matrices $\boldsymbol{w}$ and data points $\boldsymbol{z}$. In other words, this Lipschitz constant only relies on the architecture of the network instead of the weight matrices or data distribution. Hence, it is the same for a network trained from corrupted data and a network trained from true data. We remark that the Lipschitz constant used by Pensia et al. (2018); Bu et al. (2020); Mou et al. (2018) is different from the Lipschitz constant of the function corresponding to a network w.r.t. the input variable. The latter one has been used in the literature (see e.g., Bartlett et al., 2017) for deriving generalization bounds and, to some degree, can capture generalization phenomena, such as label corruption.

The order of our generalization bound in Corollary 1 is $\min\left(\frac{1}{n}\sum_{t=1}^{T}\sqrt{\beta\eta_t},\sqrt{\frac{\beta}{bn}\sum_{t=1}^{T}\eta_t}\right)$. It is tighter than Theorem 2 of Mou et al. (2018) whose order is $\sqrt{\frac{\beta}{n}\sum_{t=1}^{T}\eta_t}$. Our bound is applicable regardless of the choice of learning rate while the bound in Li et al. (2019) requires the scale of the learning rate to be upper bounded by the reciprocal of the Lipschitz constant. Our Corollary 1 has the same order with Negrea et al. (2019) but we incorporate an additional decay factor under additional assumptions (see Theorem 2) and numerical experiments suggest that our bound is more favourably correlated with the true generalization gap (see Table 1).

Theorem 1 provides an upper bound for the *expected generalization gap* of the SGLD algorithm. In practice, a bound holding with high probability is also worth investigating since the SGLD algorithm may be run only once using the training set. Next, we leverage the monitor technique (Bassily et al., 2016; Xu and Raginsky, 2017) and derive a concentration inequality for the generalization gap of SGLD.

Proposition 1. *Under Assumption 1, with probability at least $1-\delta$ over the randomness of (S,W), the generalization gap of the SGLD algorithm has the following upper bound:*

$$|L_{\mu}(\mathrm{W})-L_{\mathrm{S}}(\mathrm{W})|\leq\frac{\sqrt{2b}\sigma}{n}\sum_{j=1}^{m}\sqrt{\sum_{t\in\mathcal{T}_j}\frac{\beta_t\eta_t}{\delta}\cdot\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1},\mathrm{S}_j)\right)+\log\frac{2}{\delta}}.$$

4 Amplifying Generalization by Iteration for DP-SGD

In the last section, we derived a generalization bound for the SGLD algorithm and allowed the algorithmic output to be any function of all iterates. Here we consider a closely related algorithm—DP-SGD—and prove that the generalization bound can be sharpened by incorporating a time-decaying factor if the algorithmic output is the last iterate.

We start by recalling an implementation of the (projected) DP-SGD algorithm (see e.g., Algorithm 1 in Feldman et al., 2018). The parameter of the empirical risk is updated using the following rule:

$$\mathrm{W}_t=\mathsf{Proj}_{\mathcal{W}}\left(\mathrm{W}_{t-1}-\eta\left(g(\mathrm{W}_{t-1},\mathrm{Z}_t)+\mathrm{N}\right)\right),\tag{11}$$

where $\mathrm{N}\sim N(0,\mathbf{I}_d)$ and function g indicates a direction for updating the parameter. The parameter is projected $\mathsf{Proj}_{\mathcal{W}}(\boldsymbol{w})\triangleq\operatorname{argmin}_{\boldsymbol{w}'\in\mathcal{W}}\|\boldsymbol{w}'-\boldsymbol{w}\|_2$ onto the domain $\mathcal{W}$ at each iteration. We assume

$$\sup_{\boldsymbol{w}\in\mathcal{W},\boldsymbol{z}\in\mathcal{Z}}\|g(\boldsymbol{w},\boldsymbol{z})\|_2\leq K.\tag{12}$$

This assumption is crucial for guaranteeing differential privacy as it controls the sensitivity of each update. It is satisfied if g is the gradient of a Lipschitz continuous function or the clipped gradient of a generic loss function.

The recursion in (11) is run for T iterations and we assume that $T \leq n$ (i.e., data are drawn without replacement). The final output from the DP-SGD algorithm is the last iterate W_T. Again, our goal is to derive an upper bound for the expected generalization gap:

$$\mathbb{E}\left[L_\mu(\mathrm{W}_T) - L_\mathrm{S}(\mathrm{W}_T)\right]. \tag{13}$$

Recall that Lemma 1 provides a generalization bound in terms of the mutual information $I(\mathrm{W}_T;\mathrm{Z}_t)$. Intuitively speaking, if a data point Z_t is used at an early iteration, $I(\mathrm{W}_T;\mathrm{Z}_t)$ should be small due to the cumulative effect of the noise added in the iterations afterward. We characterize this intuition rigorously in the following lemma, which is established by strong data processing inequalities.

Lemma 4. *For the DP-SGD algorithm, we have*

$$I(\mathrm{W}_T;\mathrm{Z}_t) \leq I(\mathrm{W}_t;\mathrm{Z}_t) \cdot q^{T-t}. \tag{14}$$

Here we define

$$q \triangleq 1 - 2\bar{\Phi}\left(\frac{D + 2\eta K}{2\eta}\right) \in (0,1) \tag{15}$$

where $\bar{\Phi}(\cdot)$ is the Gaussian CCDF and the constant D is the diameter of the parameter domain $\mathcal{W}$.

Lemma 4 explains why our generalization bound in Theorem 2 incorporates a time-decaying factor. In particular, it implies that $I(\mathrm{W}_T;\mathrm{Z}_t) \to 0$ as $T \to \infty$. This time-decaying phenomenon occurs because the output from DP-SGD is W_T while all the intermediate steps are not released. To further illustrate this point, let us imagine the opposite extreme scenario in which the DP-SGD algorithm outputs the parameters across all iterations: $\mathrm{W}_1, \cdots, \mathrm{W}_T$. The data processing inequality yields that for the data point Z_t used at the t-th iteration,

$$I(\mathrm{W}_1, \cdots, \mathrm{W}_T;\mathrm{Z}_t) \geq I(\mathrm{W}_t;\mathrm{Z}_t).$$

Hence, it is impossible to have $I(\mathrm{W}_1, \cdots, \mathrm{W}_T;\mathrm{Z}_t) \to 0$ as $T \to \infty$ unless $I(\mathrm{W}_t;\mathrm{Z}_t) = 0$.

Since the underlying data distribution is unknown, so is the mutual information $I(\mathrm{W}_t;\mathrm{Z}_t)$ in (14), which poses a problem for computing the generalization bound. In order to obtain a computable bound like Theorem 1, we further upper bound the mutual information by employing properties of Gaussian channels (Lemma 2).

Lemma 5. *For the DP-SGD algorithm, we have*

$$I(\mathrm{W}_t;\mathrm{Z}_t) \leq 2 \cdot \mathsf{Var}\left(g(\mathrm{W}_{t-1},\mathrm{Z})\right), \tag{16}$$

where the variance is over the randomness of $(\mathrm{W}_{t-1},\mathrm{Z}) \sim P_{\mathrm{W}_{t-1}} \otimes \mu$.

With Lemma 4, 5 in hand, we now present the main result of this section—a generalization bound for the DP-SGD algorithm.

Theorem 2. *Under Assumption 1, the expected generalization gap of the DP-SGD algorithm has the following upper bound:*

$$\mathbb{E}\left[L_\mu(\mathrm{W}_T) - L_\mathrm{S}(\mathrm{W}_T)\right] \leq \frac{2\sigma}{n} \sum_{t=1}^{T} \sqrt{\mathsf{Var}\left(g\left(\mathrm{W}_{t-1},\mathrm{Z}\right)\right) \cdot q^{T-t}}, \tag{17}$$

where q is defined in (15).

Remark 2. To qualitatively analyze the role of the decay factor, we demonstrate that as $T \to \infty$, the generalization bound in Theorem 2 converges to 0, whereas the counterpart without a decay factor may tend to a large constant. Under assumption (12), we can further upper bound the variance term by K^2. In this case, our bound in Theorem 2 becomes

$$\frac{C}{n} \sum_{t=1}^{T} \sqrt{q^{T-t}} = \frac{C}{n} \frac{1 - q^{T/2}}{1 - \sqrt{q}},$$

where the constant $C \triangleq 2\sigma K > 0$. By choosing $T = n$ and letting them go to infinity, the above bound converges to 0 but the counterpart without the decay factor can only tend to the constant C.

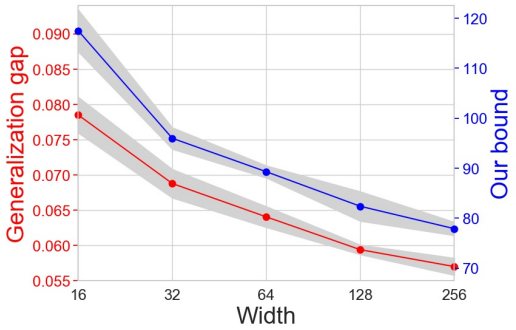

Figure 2: Comparison between the generalization gap and our generalization bound in Theorem 1. We use the SGLD algorithm to train 3-layer neural networks with varying widths on MNIST. As shown, both the generalization gap and our bound are decreasing w.r.t. the network width.

5 Numerical Experiments

In this section, we demonstrate our generalization bound in Theorem 1 through numerical experiments on the MNIST dataset. First, we validate the ability of our generalization bound in capturing generalization phenomena of DNNs observed by empirical studies (see e.g., Zhang et al., 2016). Second, we examine the correlation between our generalization bound and the true generalization gap using the three evaluation criteria suggested by Jiang et al. (2019), and compare it with the benchmarks. We reproduce our experiments on the CIFAR-10 dataset (Krizhevsky et al., 2009) in the supplementary material.

Corrupted label. As observed in Zhang et al. (2016), DNNs have the potential to memorize the entire training dataset even when a large portion of the labels are corrupted. For networks with identical architecture, those trained using true labels have better generalization capability than those ones trained using corrupted labels, although both of them achieve perfect training accuracy.

In our experiment, we randomly select 5000 samples as our training dataset and change the label of $\alpha \in \{0\%, 25\%, 50\%, 75\%\}$ training samples. Then we use the SGLD algorithm to train a 3-layer network under different corruption levels. The training process continues until the training accuracy is 1.0 (see Figure 1 Left). We compare our generalization bound with the generalization gap in Figure 1 Middle and Right. As shown, despite being vacuous, our bound is highly correlated with the true gap. When the corruption level is increasing, both our bound and the generalization gap are increasing and the curve of our bound has a very similar shape to the generalization gap. Finally, we observe that the generalization gap tends to be stable since the algorithm converges (Figure 1 Middle). Our generalization bound captures this phenomenon (Figure 1 Right) as the variance of gradients becomes negligible when the algorithm starts converging. The intuition is that the variance of gradients reflects the sharpness of the loss landscape and as the algorithm converges, the loss landscape becomes flatter.

Network width. As observed by several recent studies (see e.g., Neyshabur et al., 2014; Jiang et al., 2019), wider networks can lead to a smaller generalization gap. This may seem contradictory to the traditional wisdom as one may expect that a class of wider networks has a higher VC-dimension and, hence, would exhibit a higher generalize gap. In our experiments, we use the SGLD algorithm to train neural networks with different widths. The training process runs for 400 epochs until the training accuracy is 1.0. We compare our generalization bound with the generalization gap in Figure 2. As shown, both the generalization gap and our bound are decreasing with respect to the network width.

Comparison with benchmarks. To evaluate our bound, we adopt the three criteria proposed in Jiang et al. (2019): (i) Kendall's rank-correlation coefficient (τ) (Kendall, 1938), (ii) Granulated Kendall's coefficient (Ψ), and (iii) conditional independent test via mutual information (MI) (Verma and Pearl, 1991). In our experiments, we select 3 commonly used hyper-parameters (i.e., learning rate (lr), width, depth), which are believed to influence the generalization gap, and let each hyper-parameter choose three different values. We train 27 neural networks under all combinations of

dataset	method	lr	width	depth	τ	Ψ	MI
MNIST	OURS (THEOREM 1)	**0.70**	**1.00**	**0.56**	**0.50**	**0.75**	**0.34**
	NEGREA ET AL. (2019)	0.26	0.26	0.48	0.25	0.33	0.12

Table 1: We adopt the three evaluation criteria proposed in Jiang et al. (2019) for comparing our generalization bound with the benchmark method (Theorem 3.1 of Negrea et al., 2019): (i) Kendall's rank-correlation coefficient (τ), (ii) Granulated Kendall's coefficient (Ψ), and (iii) conditional independent test (MI). All scores, except MI, are within $[-1, 1]$ and the score of MI is normalized to $[0, 1]$. We also report the correlations when a single hyper-parameter (e.g., learning rate (lr)) is varying.

hyper-parameters and assess the correlations between the generalization bound and the generalization gap.

We compare our generalization bound with the gradient-prediction-residual bound in Theorem 3.1 of Negrea et al. (2019) under the above three evaluation criteria. As shown in Table 1, our generalization bound is highly correlated with the true generalization gap and outperforms the benchmark under all the criteria suggested in Jiang et al. (2019).

6 Summary

We provide new generalization bounds for the SGLD algorithm. Our hope is that these bounds can help explain some empirical observations (e.g., why over-parameterized DNNs often generalize well in practice) and inspire new regularization methods. The proof techniques in this paper rely on information-theoretic tools (e.g., strong data processing inequalities and properties of Gaussian channels). We believe that these tools can find a wider applicability within the community studying generalization theory. Our approach can be extended in several directions. For example, one could tighten our time-decaying factor in Theorem 2 by letting it be distribution-dependent. Moreover, the proof blueprint outlined here can be applied to analyze a broader family of noisy iterative algorithms that rely on additive noise. There are several open questions that deserve further investigation. For example, we prove that our generalization bound can be tightened if the output of the algorithm is the last iterate. Our analysis is inspired by a line of works on privacy amplification by iteration (Feldman et al., 2018; Balle et al., 2019; Asoodeh et al., 2020). On the other hand, there are other ways to amplify privacy, such as subsampling (Chaudhuri and Mishra, 2006) and shuffling (Erlingsson et al., 2019). It would be interesting to understand if the algorithmic generalization capability can be improved by these methods.

Acknowledgments and Disclosure of Funding

The work of H. Wang and F. P. Calmon is supported in part by the National Science Foundation under grants CAREER 1845852, IIS 1926925, and FAI 2040880 and F. P. Calmon also acknowledges a gift from Google Faculty Research Award and an Amazon Research Award.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Ahlswede, R. and Gács, P. (1976). Spreading of sets in product spaces and hypercontraction of the Markov operator. *The annals of probability*, pages 925–939.

Asoodeh, S., Diaz, M., and Calmon, F. P. (2020). Privacy analysis of online learning algorithms via contraction coefficients. In *International Symposium on Information Theory*.

Balle, B., Barthe, G., Gaboardi, M., and Geumlek, J. (2019). Privacy amplification by mixing and diffusion mechanisms. *Advances in neural information processing systems*.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249.

Bassily, R., Nissim, K., Smith, A., Steinke, T., Stemmer, U., and Ullman, J. (2016). Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1046–1059.

Bu, Y., Zou, S., and Veeravalli, V. V. (2020). Tightening mutual information based bounds on generalization error. *IEEE Journal on Selected Areas in Information Theory*.

Calmon, F. P., Polyanskiy, Y., and Wu, Y. (2017). Strong data processing inequalities for input constrained additive noise channels. *IEEE Transactions on Information Theory*, 64(3):1879–1892.

Chaudhuri, K. and Mishra, N. (2006). When random sampling preserves privacy. In *Annual International Cryptology Conference*, pages 198–213. Springer.

Cohen, J. E., Kempermann, J., and Zbaganu, G. (1998). *Comparisons of stochastic matrices with applications in information theory, statistics, economics and population*. Springer Science & Business Media.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318.

Dobrushin, R. L. (1956). Central limit theorem for nonstationary Markov chains. I. *Theory of Probability & Its Applications*, 1(1):65–80.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. (2019). Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM.

Facebook AI (2020). Introducing Opacus: A high-speed library for training PyTorch models with differential privacy.

Feldman, V., Mironov, I., Talwar, K., and Thakurta, A. (2018). Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 521–532. IEEE.

Gelfand, S. B. and Mitter, S. K. (1991). Recursive stochastic algorithms for global optimization in R^d. *SIAM Journal on Control and Optimization*, 29(5):999–1018.

Guo, D., Shamai, S., and Verdú, S. (2005). Mutual information and minimum mean-square error in Gaussian channels. *IEEE Transactions on Information Theory*, 51(4):1261–1282.

Haghifam, M., Negrea, J., Khisti, A., Roy, D. M., and Dziugaite, G. K. (2020). Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms. *arXiv preprint arXiv:2004.12983*.

Hardt, M., Recht, B., and Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. (2019). Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*.

Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. (2019). On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *arXiv preprint arXiv:1902.04811*.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Li, J., Luo, X., and Qiao, M. (2019). On generalization error bounds of noisy gradient methods for non-convex learning. *arXiv preprint arXiv:1902.00621*.

Mou, W., Wang, L., Zhai, X., and Zheng, K. (2018). Generalization bounds of SGLD for non-convex learning: Two theoretical viewpoints. In *Conference on Learning Theory*, pages 605–638.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. (2015). Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.

Negrea, J., Haghifam, M., Dziugaite, G. K., Khisti, A., and Roy, D. M. (2019). Information-theoretic generalization bounds for SGLD via data-dependent estimates. In *Advances in Neural Information Processing Systems*, pages 11015–11025.

Neu, G., Dziugaite, G. K., Haghifam, M., and Roy, D. M. (2021). Information-theoretic generalization bounds for stochastic gradient descent. *arXiv preprint arXiv:2102.00931*.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. (2017). Exploring generalization in deep learning. In *Advances in neural information processing systems*, pages 5947–5956.

Neyshabur, B., Tomioka, R., and Srebro, N. (2014). In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*.

Otto, F. and Villani, C. (2000). Generalization of an inequality by Talagrand and links with the logarithmic Sobolev inequality. *Journal of Functional Analysis*, 173(2):361–400.

Pensia, A., Jog, V., and Loh, P.-L. (2018). Generalization error bounds for noisy, iterative algorithms. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 546–550. IEEE.

Polyanskiy, Y. and Wu, Y. (2016). Dissipation of information in channels with input constraints. *IEEE Transactions on Information Theory*, 62(1):35–55.

Polyanskiy, Y. and Wu, Y. (2019). Lecture notes on information theory. *Lecture Notes for 6.441 (MIT), ECE 563 (UIUC), STAT 364 (Yale)*.

Radebaugh, C. and Erlingsson, U. (2019). Introducing TensorFlow privacy: Learning with differential privacy for training data.

Raginsky, M. (2016). Strong data processing inequalities and Φ-Sobolev inequalities for discrete channels. *IEEE Transactions on Information Theory*, 62(6):3355–3389.

Raginsky, M., Rakhlin, A., and Telgarsky, M. (2017). Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703.

Raginsky, M. and Sason, I. (2012). Concentration of measure inequalities in information theory, communications and coding. *arXiv preprint arXiv:1212.4663*.

Rodríguez-Gálvez, B., Bassi, G., Thobaben, R., and Skoglund, M. (2020). On random subset generalization error bounds and the stochastic gradient Langevin dynamics algorithm. *arXiv preprint arXiv:2010.10994*.

Russo, D. and Zou, J. (2016). Controlling bias in adaptive data analysis using information theory. In *Artificial Intelligence and Statistics*, pages 1232–1240. PMLR.

Song, S., Chaudhuri, K., and Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Vapnik, V. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Measures of Complexity*, 16(2):11.

Verma, T. and Pearl, J. (1991). *Equivalence and synthesis of causal models*. UCLA, Computer Science Department.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML)*, pages 681–688.

Xu, A. and Raginsky, M. (2017). Information-theoretic analysis of generalization capability of learning algorithms. In *Adv. Neural Inf. Process. Syst.*, pages 2524–2533.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhang, Y., Liang, P., and Charikar, M. (2017). A hitting time analysis of stochastic gradient Langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022.

A Omitted Proofs

A.1 Proof of Theorem 1

We first present an extension of Lemma 3 which can be proved by using the technique in Section II. E of Guo et al. (2005).

Lemma 6. *Let* X *be a random variable which is independent of* $\mathrm{N} \sim N(0, \mathbf{I}_d)$*. Then for any* $m > 0$ *and function* f

$$I(f(\mathrm{X}) + m\mathrm{N}; \mathrm{X}) \leq \frac{1}{2m^2}\mathsf{Var}\left(f(\mathrm{X})\right). \tag{18}$$

More generally, if Z *is another random variable which is independent of* N*, then for any fixed* $\boldsymbol{z}$

$$I(f(\mathrm{X}) + m\mathrm{N}; \mathrm{X} \mid \mathrm{Z} = \boldsymbol{z}) \leq \frac{1}{2m^2}\mathsf{Var}\left(f(\mathrm{X}) \mid \mathrm{Z} = \boldsymbol{z}\right). \tag{19}$$

Proof. By the property of mutual information (see Theorem 2.3 in Polyanskiy and Wu, 2019),

$$I(f(\mathrm{X}) + m\mathrm{N}; \mathrm{X}) = I\left(\frac{f(\mathrm{X}) - \boldsymbol{e}}{m} + \mathrm{N}; \mathrm{X}\right) \tag{20}$$

where $\boldsymbol{e} \triangleq \mathbb{E}\left[f(\mathrm{X})\right]$. We denote

$$g(\boldsymbol{x}) \triangleq \frac{f(\boldsymbol{x}) - \boldsymbol{e}}{m}. \tag{21}$$

The golden formula (see Theorem 3.3 in Polyanskiy and Wu, 2019, for a proof) yields

$$\begin{aligned} I\left(g(\mathrm{X}) + \mathrm{N}; \mathrm{X}\right) &= \mathrm{D}_{\mathrm{KL}}\left(P_{g(\mathrm{X})+\mathrm{N}|\mathrm{X}} \| P_{\mathrm{N}} | P_{\mathrm{X}}\right) - \mathrm{D}_{\mathrm{KL}}\left(P_{g(\mathrm{X})+\mathrm{N}} \| P_{\mathrm{N}}\right) \\ &\leq \mathrm{D}_{\mathrm{KL}}\left(P_{g(\mathrm{X})+\mathrm{N}|\mathrm{X}} \| P_{\mathrm{N}} | P_{\mathrm{X}}\right). \end{aligned} \tag{22}$$

Furthermore, since X and N are independent, we have

$$\mathrm{D}_{\mathrm{KL}}\left(P_{g(\mathrm{X})+\mathrm{N}|\mathrm{X}=\boldsymbol{x}} \| P_{\mathrm{N}}\right) = \mathrm{D}_{\mathrm{KL}}\left(P_{g(\boldsymbol{x})+\mathrm{N}} \| P_{\mathrm{N}}\right) = \frac{\|g(\boldsymbol{x})\|_2^2}{2},$$

where the last step is due to the closed-form expression of the KL-divergence between two Gaussian distributions. Finally, by the definition of conditional divergence, we have

$$\mathrm{D}_{\mathrm{KL}}\left(P_{g(\mathrm{X})+\mathrm{N}|\mathrm{X}} \| P_{\mathrm{N}} | P_{\mathrm{X}}\right) = \frac{1}{2}\mathbb{E}\left[\|g(\mathrm{X})\|_2^2\right] = \frac{1}{2m^2}\mathsf{Var}\left(f(\mathrm{X})\right), \tag{23}$$

where the last step is due to the definition of g in (21). Combining (20–23) leads to the desired conclusion. Finally, it is straightforward to obtain (19) by conditioning on $\mathrm{Z} = \boldsymbol{z}$ and repeating our above derivations. □

Next, we present the second lemma which will be used for proving Theorem 1.

Lemma 7. *Under Assumption 1, the expected generalization gap* (3) *of the SGLD algorithm can be upper bounded by*

$$\frac{\sqrt{2}\sigma}{2n}\sum_{j=1}^{m}\sqrt{\sum_{t\in\mathcal{T}_j}\beta_t\eta_t \cdot \mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1}, \bar{\mathrm{Z}}_j)\right)},$$

where the set $\mathcal{T}_j$ *contains the indices of iterations in which the mini-batch* S_j *is used and the variance is over the randomness of* $(\mathrm{W}_{t-1}, \bar{\mathrm{Z}}_j) \sim P_{\mathrm{W}_{t-1}, \bar{\mathrm{Z}}_j}$ *with* $\bar{\mathrm{Z}}_j$ *being any data point in the mini-batch* S_j*.*

Proof. We denote $\mathrm{Z}^{(k)} \triangleq (\mathrm{Z}_1, \cdots, \mathrm{Z}_k)$ for $k \in [n]$ and $\mathrm{W}^{(t)} \triangleq (\mathrm{W}_1, \cdots, \mathrm{W}_t)$ for $t \in [T]$. For simplicity, in what follows we only provide an upper bound for $I(\mathrm{W}; \mathrm{Z}_n)$. Since W is a function of $\mathrm{W}^{(T)} = (\mathrm{W}_1, \cdots, \mathrm{W}_T)$, the data processing inequality yields

$$I(\mathrm{W}; \mathrm{Z}_n) \leq I(\mathrm{W}^{(T)}; \mathrm{Z}_n) \leq I(\mathrm{W}^{(T)}, \mathrm{Z}^{(n-1)}; \mathrm{Z}_n). \tag{24}$$

By the chain rule,

$$I(\mathrm{W}^{(T)}, \mathrm{Z}^{(n-1)}; \mathrm{Z}_n) = I(\mathrm{W}_T; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)}, \mathrm{Z}^{(n-1)}) + I(\mathrm{W}^{(T-1)}, \mathrm{Z}^{(n-1)}; \mathrm{Z}_n). \tag{25}$$

Let $\boldsymbol{w} = (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{T-1})$ and $\boldsymbol{z} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_{n-1})$ be any two vectors. If Z_n is not used at the T-th iteration, without loss of generality we assume that the data points $\mathrm{Z}_1, \cdots, \mathrm{Z}_b$ are used in this iteration. Then

$$\begin{aligned}
&I(\mathrm{W}_T; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}) \\
&= I\left(\boldsymbol{w}_{t-1} - \frac{\eta_T}{b}\sum_{i=1}^{b} \nabla_{\boldsymbol{w}} \hat{\ell}(\boldsymbol{w}_{T-1}, \boldsymbol{z}_i) + \sqrt{\frac{2\eta_T}{\beta_T}}\mathrm{N}; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}\right) \\
&= I\left(\mathrm{N}; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}\right) \\
&= 0.
\end{aligned} \tag{26}$$

On the other hand, if Z_n is used at the T-th iteration, without loss of generality we assume that the other $b-1$ data points which are also used in this iteration are $\mathrm{Z}_1, \cdots, \mathrm{Z}_{b-1}$. Then

$$\begin{aligned}
&I(\mathrm{W}_T; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}) \\
&= I\left(\boldsymbol{w}_{t-1} - \frac{\eta_T}{b}\left(\sum_{i=1}^{b-1} \nabla_{\boldsymbol{w}} \hat{\ell}(\boldsymbol{w}_{T-1}, \boldsymbol{z}_i) + \nabla_{\boldsymbol{w}} \hat{\ell}(\boldsymbol{w}_{T-1}, \mathrm{Z}_n)\right) + \sqrt{\frac{2\eta_T}{\beta_T}}\mathrm{N}; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}\right) \\
&= I\left(-\frac{\eta_T}{b}\nabla_{\boldsymbol{w}} \hat{\ell}(\boldsymbol{w}_{T-1}, \mathrm{Z}_n) + \sqrt{\frac{2\eta_T}{\beta_T}}\mathrm{N}; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}\right).
\end{aligned} \tag{27}$$

By Lemma 6, we have

$$\begin{aligned}
&I\left(-\frac{\eta_T}{b}\nabla_{\boldsymbol{w}} \hat{\ell}(\boldsymbol{w}_{T-1}, \mathrm{Z}_n) + \sqrt{\frac{2\eta_T}{\beta_T}}\mathrm{N}; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}\right) \\
&\leq \frac{\beta_T \eta_T}{4b^2}\mathsf{Var}\left(\nabla_{\boldsymbol{w}} \hat{\ell}(\boldsymbol{w}_{T-1}, \mathrm{Z}_n) \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}\right).
\end{aligned} \tag{28}$$

Substituting (28) into (27) gives

$$I(\mathrm{W}_T; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}) \leq \frac{\beta_T \eta_T}{4b^2}\mathsf{Var}\left(\nabla_{\boldsymbol{w}} \hat{\ell}(\boldsymbol{w}_{T-1}, \mathrm{Z}_n) \mid \mathrm{W}^{(T-1)} = \boldsymbol{w}, \mathrm{Z}^{(n-1)} = \boldsymbol{z}\right).$$

Taking expectation w.r.t. $(\mathrm{W}^{(T-1)}, \mathrm{Z}^{(n-1)})$ on both sides of the above inequality and using the law of total variance lead to

$$I(\mathrm{W}_T; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)}, \mathrm{Z}^{(n-1)}) \leq \frac{\beta_T \eta_T}{4b^2}\mathsf{Var}\left(\nabla_{\boldsymbol{w}} \hat{\ell}(\mathrm{W}_{T-1}, \mathrm{Z}_n)\right). \tag{29}$$

To summarize, (26) and (29) can be rewritten as

$$\begin{aligned}
&I(\mathrm{W}_T; \mathrm{Z}_n \mid \mathrm{W}^{(T-1)}, \mathrm{Z}^{(n-1)}) \\
&\leq \begin{cases} \frac{\beta_T \eta_T}{4b^2}\mathsf{Var}\left(\nabla_{\boldsymbol{w}} \hat{\ell}(\mathrm{W}_{T-1}, \mathrm{Z}_n)\right) & \text{if } \mathrm{Z}_n \text{ is used at the } T\text{-th iteration,} \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{30}$$

Assume that the data point Z_n belongs to the j-th mini-batch S_j. Now substituting (30) into (25) and doing this procedure recursively lead to

$$I(\mathrm{W}^{(T)}, \mathrm{Z}^{(n-1)}; \mathrm{Z}_n) \leq \sum_{t \in \mathcal{T}_j} \frac{\beta_t \eta_t}{4b^2}\mathsf{Var}\left(\nabla_{\boldsymbol{w}} \hat{\ell}(\mathrm{W}_{t-1}, \mathrm{Z}_n)\right),$$

where the set $\mathcal{T}_j$ contains the indices of iterations in which the mini-batch S_j is used. Hence, this upper bound along with (24) naturally gives

$$I(\mathrm{W}; \mathrm{Z}_n) \leq \sum_{t \in \mathcal{T}_j} \frac{\beta_t \eta_t}{4b^2}\mathsf{Var}\left(\nabla_{\boldsymbol{w}} \hat{\ell}(\mathrm{W}_{t-1}, \mathrm{Z}_n)\right). \tag{31}$$

By symmetry, for any data point in S_j besides Z_n, the mutual information between W and this data point can be upper bound by the right-hand side of (31) as well. Finally, recall that Lemma 1 provides an upper bound for the expected generalization gap:

$$\frac{\sqrt{2}\sigma}{n}\sum_{i=1}^{n}\sqrt{I(\mathrm{W}_T;\mathrm{Z}_i)} = \frac{\sqrt{2}\sigma}{n}\sum_{j=1}^{m}\sum_{\mathrm{Z}\in\mathrm{S}_j}\sqrt{I(\mathrm{W}_T;\mathrm{Z})}. \tag{32}$$

By substituting (31) into the above expression, we know the expected generalization gap can be further upper bounded by

$$\frac{\sqrt{2}\sigma}{2n}\sum_{j=1}^{m}\sqrt{\sum_{t\in\mathcal{T}_j}\beta_t\eta_t\cdot\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1},\bar{\mathrm{Z}}_j)\right)},$$

where $\bar{\mathrm{Z}}_j$ is any data point in the mini-batch S_j. □

Now we are in a position to prove Theorem 1.

Proof. Consider a new loss function and the gradient of a new surrogate loss:

$$\ell(\boldsymbol{w},\mathrm{S}_j) \triangleq \frac{1}{b}\sum_{\mathrm{Z}\in\mathrm{S}_j}\ell(\boldsymbol{w},\mathrm{Z}), \quad \nabla_{\boldsymbol{w}}\hat{\ell}(\boldsymbol{w},\mathrm{S}_j) \triangleq \frac{1}{b}\sum_{\mathrm{Z}\in\mathrm{S}_j}\nabla_{\boldsymbol{w}}\hat{\ell}(\boldsymbol{w},\mathrm{Z}).$$

When Assumption 1 holds, $\ell(\boldsymbol{w},\mathrm{S}_j)$ is $\sigma/\sqrt{b}$-sub-Gaussian under $\mathrm{S}_j\sim\mu^{\otimes b}$ for all $\boldsymbol{w}\in\mathcal{W}$. We view each mini-batch S_j as a data point and view $\ell(\boldsymbol{w},\mathrm{S}_j)$ as a new loss function. By using Lemma 7, we obtain:

$$\left|\mathbb{E}\left[L_\mu(\mathrm{W})-L_{\mathrm{S}}(\mathrm{W})\right]\right| \le \frac{\sqrt{2}\sigma}{2m\sqrt{b}}\sum_{j=1}^{m}\sqrt{\sum_{t\in\mathcal{T}_j}\beta_t\eta_t\cdot\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1},\mathrm{S}_j)\right)}. \tag{33}$$

Since the dataset contains n data points and is divided into m disjoint mini-batches with size b, we have $n=mb$. Substituting this into (33) leads to the desired conclusion. □

A.2 Proof of Corollary 1

Proof. The Minkowski inequality implies that for any non-negative x_i, the inequality $\sqrt{\sum_i x_i}\le\sum_i\sqrt{x_i}$ holds. Therefore, we can further upper bound the generalization bound in Lemma 7 by

$$\frac{\sqrt{2}\sigma}{2n}\sum_{j=1}^{m}\sum_{t\in\mathcal{T}_j}\sqrt{\beta_t\eta_t\cdot\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1},\bar{\mathrm{Z}}_j)\right)} = \frac{\sqrt{2}\sigma}{2n}\sum_{t=1}^{T}\sqrt{\beta_t\eta_t\cdot\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1},\mathrm{Z}_t^\dagger)\right)}.$$

Alternatively, by Jensen's inequality and the equality $n=mb$, we can further upper bound the generalization bound in Lemma 7 by

$$\frac{\sqrt{2}\sigma}{2}\sqrt{\frac{1}{bn}\sum_{t=1}^{T}\beta_t\eta_t\cdot\mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1},\mathrm{Z}_t^\dagger)\right)}.$$

□

A.3 Proof of Lemma 4

Proof. For the t-th iteration, we can rewrite the recursion in (11) as

$$\mathrm{U}_t = \mathrm{W}_{t-1}-\eta\cdot g(\mathrm{W}_{t-1},\mathrm{Z}_t) \tag{34a}$$
$$\mathrm{V}_t = \mathrm{U}_t+\eta\cdot\mathrm{N} \tag{34b}$$
$$\mathrm{W}_t = \mathsf{Proj}_{\mathcal{W}}\left(\mathrm{V}_t\right). \tag{34c}$$

Since data are drawn without replacement, the following Markov chain holds:

$$\mathrm{Z}_t\to\mathrm{U}_t\to\mathrm{V}_t\to\mathrm{W}_t\to\cdots\to\mathrm{W}_{T-1}\to\mathrm{U}_T\to\mathrm{V}_T\to\mathrm{W}_T.$$

Let $\mathcal{U}_T$ be the range of U_T. By the triangle inequality,

$$\mathsf{diam}(\mathcal{U}_T) \le \mathsf{diam}(\mathcal{W}) + 2\eta K = D + 2\eta K.$$

Now we leverage strong data processing inequalities and obtain

$$\begin{aligned} I(\mathrm{W}_T; \mathrm{Z}_t) &\le I(\mathrm{V}_T; \mathrm{Z}_t) \\ &\le \left(1 - 2\bar{\Phi}\left(\frac{D + 2\eta K}{2\eta}\right)\right) \cdot I(\mathrm{U}_T; \mathrm{Z}_t) \\ &\le \left(1 - 2\bar{\Phi}\left(\frac{D + 2\eta K}{2\eta}\right)\right) \cdot I(\mathrm{W}_{T-1}; \mathrm{Z}_t), \end{aligned}$$

where the first and last steps are due to the data processing inequality. Applying this procedure recursively leads to the desired conclusion. □

A.4 Proof of Lemma 5

Proof. Recall the definition of U_t, V_t in (34). The data processing inequality yields

$$I(\mathrm{W}_t; \mathrm{Z}_t) \le I(\mathrm{V}_t; \mathrm{Z}_t). \tag{35}$$

By the definition of mutual information, we can write

$$\begin{aligned} I(\mathrm{V}_t; \mathrm{Z}_t) &= \mathbb{E}\left[\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{V}_t|\mathrm{Z}_t} \| P_{\mathrm{V}_t})\right] \\ &= \int_{\mathcal{Z}} \mathrm{D}_{\mathrm{KL}}(P_{\mathrm{V}_t|\mathrm{Z}_t=\boldsymbol{z}} \| P_{\mathrm{V}_t}) \mathrm{d}\mu(\boldsymbol{z}). \end{aligned} \tag{36}$$

Since $\mathrm{V}_t = \mathrm{U}_t + \eta \cdot \mathrm{N}$, Lemma 2 implies

$$\mathrm{D}_{\mathrm{KL}}\left(P_{\mathrm{V}_t|\mathrm{Z}_t=\boldsymbol{z}} \| P_{\mathrm{V}_t}\right) \le \frac{1}{2\eta^2} \mathbb{W}_2^2\left(P_{\mathrm{U}_t|\mathrm{Z}_t=\boldsymbol{z}}, P_{\mathrm{U}_t}\right). \tag{37}$$

To further upper bound the above Wasserstein distance, we construct a special coupling. Let W_{t-1} be the parameter at the $(t-1)$-st iteration. Then we introduce two random variables:

$$\begin{aligned} \mathrm{U}^*_{\boldsymbol{z}} &\triangleq \mathrm{W}_{t-1} - \eta \cdot g(\mathrm{W}_{t-1}, \boldsymbol{z}), \\ \mathrm{U}^* &\triangleq \mathrm{W}_{t-1} - \eta \cdot g(\mathrm{W}_{t-1}, \mathrm{Z}_t). \end{aligned}$$

Here $\mathrm{U}^*_{\boldsymbol{z}}$ and U^* have marginals, $P_{\mathrm{U}_t|\mathrm{Z}_t=\boldsymbol{z}}$ and P_{U_t}, respectively. By the definition of Wasserstein distance, we have

$$\begin{aligned} \mathbb{W}_2^2\left(P_{\mathrm{U}_t|\mathrm{Z}_t=\boldsymbol{z}}, P_{\mathrm{U}_t}\right) &\le \mathbb{E}\left[\|\mathrm{U}^*_{\boldsymbol{z}} - \mathrm{U}^*\|_2^2\right] \\ &= \eta^2 \cdot \mathbb{E}\left[\|(g(\mathrm{W}_{t-1}, \boldsymbol{z}) - g(\mathrm{W}_{t-1}, \mathrm{Z}_t))\|_2^2\right]. \end{aligned} \tag{38}$$

Since Z_t is only used at the t-th iteration, it is independent of W_{t-1}. We introduce two independent copies $\mathrm{Z}, \bar{\mathrm{Z}}$ of Z_t such that $(\mathrm{W}_{t-1}, \mathrm{Z}, \bar{\mathrm{Z}}) \sim P_{\mathrm{W}_{t-1}} \otimes \mu \otimes \mu$. Combining (36–38) and using Tonelli's theorem lead to

$$I(\mathrm{V}_t; \mathrm{Z}_t) \le \frac{1}{2} \cdot \mathbb{E}\left[\|(g(\mathrm{W}_{t-1}, \mathrm{Z}) - g(\mathrm{W}_{t-1}, \bar{\mathrm{Z}}))\|_2^2\right]. \tag{39}$$

Now we introduce a constant vector $\boldsymbol{e} \in \mathbb{R}^d$ whose value will be specified later. Since $\mathrm{Z}, \bar{\mathrm{Z}}$ follow the same distribution and $\|\boldsymbol{a} - \boldsymbol{b}\|_2^2 \le 2(\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2)$, the right-hand side of (39) can be upper bounded by

$$2 \cdot \mathbb{E}\left[\|g(\mathrm{W}_{t-1}, \mathrm{Z}) - \boldsymbol{e}\|_2^2\right]. \tag{40}$$

By choosing $\boldsymbol{e} = \mathbb{E}\left[g(\mathrm{W}_{t-1}, \mathrm{Z})\right]$, the above quantity becomes

$$2 \cdot \mathsf{Var}\left(g(\mathrm{W}_{t-1}, \mathrm{Z})\right). \tag{41}$$

Finally, combining (35) with (39–41) leads to the desired conclusion. □

A.5 Proof of Theorem 2

Proof. Lemma 1 implies that the expected generalization gap can be upper bounded by

$$\frac{1}{n}\sum_{i=1}^{n}\sqrt{2\sigma^2 I(\mathrm{W}_T;\mathrm{Z}_i)} = \frac{1}{n}\sum_{t=1}^{T}\sqrt{2\sigma^2 I(\mathrm{W}_T;\mathrm{Z}_t)}. \tag{42}$$

Lemma 4 and 5 altogether yield

$$I(\mathrm{W}_T;\mathrm{Z}_t) \le I(\mathrm{W}_t;\mathrm{Z}_t)\cdot q^{T-t} \le 2\cdot \mathsf{Var}\left(g(\mathrm{W}_{t-1},\mathrm{Z})\right)\cdot q^{T-t}.$$

Consequently, the expected generalization gap can be upper bounded by

$$\frac{2\sigma}{n}\sum_{t=1}^{T}\sqrt{\mathsf{Var}\left(g(\mathrm{W}_{t-1},\mathrm{Z})\right)\cdot q^{T-t}}.$$

□

A.6 Proof of Proposition 1

The proof of Proposition 1 relies on the following lemma which can be established by a slight tweak of the proof of Theorem 3 in Xu and Raginsky (2017). We reproduce it for the sake of completeness.

Lemma 8. *Under Assumption 1, with probability at least* $1-\delta$*, we have*

$$|L_\mu(\mathrm{W}) - L_\mathrm{S}(\mathrm{W})| \le \frac{2\sqrt{2}\sigma}{n}\sum_{i=1}^{n}\sqrt{\frac{I(\mathrm{W};\mathrm{Z}_i)}{\delta} + \log\frac{2}{\delta}}, \tag{43}$$

where the probability is over (S,W).

Proof. Let $\mathrm{S}_1,\cdots,\mathrm{S}_k$ be k independent copies of the dataset S such that each copy S_j contains n i.i.d. points $\mathrm{S}_j = (\mathrm{Z}_{1,j},\cdots,\mathrm{Z}_{n,j})$. The learning algorithm is applied parallelly to each dataset S_j and outputs W_j. In other words, the pairs $(\mathrm{S}_j,\mathrm{W}_j)$ with $j\in[k]$ are independent copies of (S,W). Imagine that there is a monitor which has access to the underlying distribution, the k independent copies of the dataset, and the outputs from k parallel algorithms. The monitor evaluates these output parameters and finds the one which overfits the most. Specifically, the monitor returns a tuple $(\mathrm{W}^*,\mathrm{J},\mathrm{R})$ defined by

$$(\mathrm{J},\mathrm{R}) \triangleq \operatorname*{argmax}_{j\in[k],\ r\in\{\pm 1\}} r\left(L_\mu(\mathrm{W}_j) - L_{\mathrm{S}_j}(\mathrm{W}_j)\right) \quad \text{and} \quad \mathrm{W}^* \triangleq \mathrm{W}_\mathrm{J}.$$

We view the combination of the k parallel algorithms and the monitor as a new learning algorithm. This learning algorithm receives a dataset $\mathrm{S}^k \triangleq (\mathrm{Z}_1^k,\cdots,\mathrm{Z}_n^k)$ which contains n i.i.d. data points $\mathrm{Z}_i^k \triangleq (\mathrm{Z}_{i,1},\cdots,\mathrm{Z}_{i,k})$ and outputs $(\mathrm{W}^*,\mathrm{J},\mathrm{R})$. The loss function of this new learning algorithm is $\ell : \mathcal{W}\times[k]\times\{\pm1\}\times\mathcal{Z}^k$ defined as

$$\ell(w,j,r;\boldsymbol{z}^k) \triangleq r\ell(w,\boldsymbol{z}_j),$$

where $\boldsymbol{z}_j$ is the j-th coordinate of $\boldsymbol{z}^k$. When Assumption 1 holds, $\ell(w,j,r;\mathrm{Z}^k)$ is also σ-sub-Gaussian under $\mathrm{Z}^k\sim\mu^{\otimes k}$ for all $(w,j,r)\in\mathcal{W}\times[k]\times\{\pm1\}$. Hence, applying Lemma 1 to this new learning algorithm gives

$$\mathbb{E}\left[\mathrm{R}\left(L_\mu(\mathrm{W}^*) - L_{\mathrm{S}_\mathrm{J}}(\mathrm{W}^*)\right)\right] \le \frac{1}{n}\sum_{i=1}^{n}\sqrt{2\sigma^2 I(\mathrm{W}^*,\mathrm{J},\mathrm{R};\mathrm{Z}_i^k)}. \tag{44}$$

By the definition of $(\mathrm{W}^*,\mathrm{J},\mathrm{R})$, we have

$$\mathbb{E}\left[\mathrm{R}\left(L_\mu(\mathrm{W}^*) - L_{\mathrm{S}_\mathrm{J}}(\mathrm{W}^*)\right)\right] = \mathbb{E}\left[\max_{j\in[k]}\left|L_\mu(\mathrm{W}_j) - L_{\mathrm{S}_j}(\mathrm{W}_j)\right|\right]. \tag{45}$$

Let $\mathrm{W}^k \triangleq (\mathrm{W}_1, \cdots, \mathrm{W}_k)$ be the collection of outputs from k parallel algorithms. Using the chain rule for mutual information gives

$$\begin{aligned} I(\mathrm{W}^*, \mathrm{J}, \mathrm{R}; \mathrm{Z}_i^k) &\le I(\mathrm{W}^k, \mathrm{W}^*, \mathrm{J}, \mathrm{R}; \mathrm{Z}_i^k) \\ &= I(\mathrm{W}^k; \mathrm{Z}_i^k) + I(\mathrm{W}^*, \mathrm{J}, \mathrm{R}; \mathrm{Z}_i^k \mid \mathrm{W}^k) \\ &= \sum_{j=1}^{k} I(\mathrm{W}_j; \mathrm{Z}_{i,j}) + I(\mathrm{W}^*, \mathrm{J}, \mathrm{R}; \mathrm{Z}_i^k \mid \mathrm{W}^k), \end{aligned}$$

where the last step is because of the independence among $(\mathrm{W}_j, \mathrm{Z}_{i,j})$ for $j \in [k]$. Since $(\mathrm{W}^*, \mathrm{J}, \mathrm{R})$ can take at most $2k$ different values given W^k, then

$$I(\mathrm{W}^*, \mathrm{J}, \mathrm{R}; \mathrm{Z}_i^k \mid \mathrm{W}^k) \le \log(2k).$$

By symmetry, $I(\mathrm{W}_j; \mathrm{Z}_{i,j})$ is the same for all $j \in [k]$ which is equal to $I(\mathrm{W}; \mathrm{Z}_i)$. Therefore,

$$I(\mathrm{W}^*, \mathrm{J}, \mathrm{R}; \mathrm{Z}_i^k) \le kI(\mathrm{W}; \mathrm{Z}_i) + \log(2k). \tag{46}$$

Combining (44–46) gives

$$\mathbb{E}\left[\max_{j\in[k]} \left|L_\mu(\mathrm{W}_j) - L_{\mathrm{S}_j}(\mathrm{W}_j)\right|\right] \le \frac{1}{n}\sum_{i=1}^{n} \sqrt{2\sigma^2\left(kI(\mathrm{W}; \mathrm{Z}_i) + \log(2k)\right)}. \tag{47}$$

Since $(\mathrm{S}_j, \mathrm{W}_j)$ with $j \in [k]$ are independent copies of (S, W), then for any $\alpha > 0$,

$$\Pr\left(\max_{j\in[k]} \left|L_\mu(\mathrm{W}_j) - L_{\mathrm{S}_j}(\mathrm{W}_j)\right| < \alpha\right) = \Pr\left(|L_\mu(\mathrm{W}) - L_{\mathrm{S}}(\mathrm{W})| < \alpha\right)^k. \tag{48}$$

By Markov's inequality,

$$\Pr\left(\max_{j\in[k]} \left|L_\mu(\mathrm{W}_j) - L_{\mathrm{S}_j}(\mathrm{W}_j)\right| \ge \alpha\right) \le \frac{1}{\alpha}\mathbb{E}\left[\max_{j\in[k]} \left|L_\mu(\mathrm{W}_j) - L_{\mathrm{S}_j}(\mathrm{W}_j)\right|\right]. \tag{49}$$

Substituting (47), (48) into (49) leads to

$$1 - \Pr\left(|L_\mu(\mathrm{W}) - L_{\mathrm{S}}(\mathrm{W})| < \alpha\right)^k \le \frac{1}{\alpha n}\sum_{i=1}^{n} \sqrt{2\sigma^2\left(kI(\mathrm{W}; \mathrm{Z}_i) + \log(2k)\right)},$$

which is equivalent to

$$\Pr\left(|L_\mu(\mathrm{W}) - L_{\mathrm{S}}(\mathrm{W})| < \alpha\right) \ge \left(1 - \frac{1}{\alpha n}\sum_{i=1}^{n} \sqrt{2\sigma^2\left(kI(\mathrm{W}; \mathrm{Z}_i) + \log(2k)\right)}\right)^{1/k}. \tag{50}$$

For any given $\delta \in (0, 1)$, we take $k = \lfloor 1/\delta \rfloor$ and

$$\alpha^* = \frac{2}{n}\sum_{i=1}^{n} \sqrt{2\sigma^2\left(\frac{I(\mathrm{W}; \mathrm{Z}_i)}{\delta} + \log\frac{2}{\delta}\right)}.$$

Hence, (50) indicates that

$$\begin{aligned} \Pr\left(|L_\mu(\mathrm{W}) - L_{\mathrm{S}}(\mathrm{W})| < \alpha^*\right) &\ge \left(1 - \frac{1}{\alpha^* n}\sum_{i=1}^{n} \sqrt{2\sigma^2\left(\frac{I(\mathrm{W}; \mathrm{Z}_i)}{\delta} + \log\frac{2}{\delta}\right)}\right)^{1/\lfloor 1/\delta \rfloor} \\ &= \frac{1}{2}^{1/\lfloor 1/\delta \rfloor} \ge 1 - \delta. \end{aligned}$$

□

B Additional Experiments

We conduct additional numerical experiments on CIFAR-10 (Krizhevsky et al., 2009) to further validate our generalization bound in Theorem 1 (see Figure 3, 4 and Table 2).

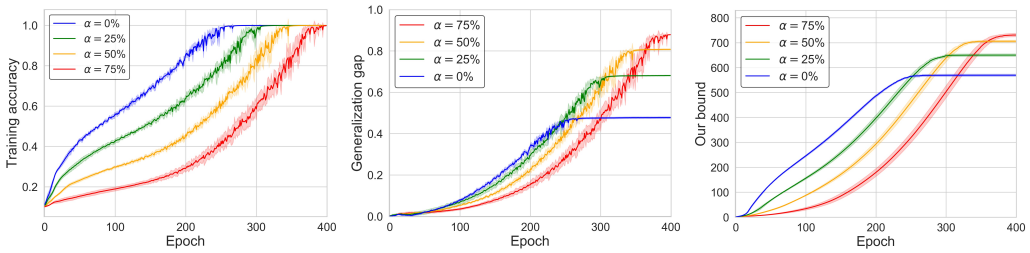

Figure 3: Illustration of our generalization bound in Theorem 1. We use the SGLD algorithm to train convolutional neural networks (CNNs) on CIFAR-10 when the training data have different label corruption level $\alpha \in \{0\%, 25\%, 50\%, 75\%\}$. Left: training accuracy. Middle: (empirical) generalization gap. Right: (empirical) generalization bound.

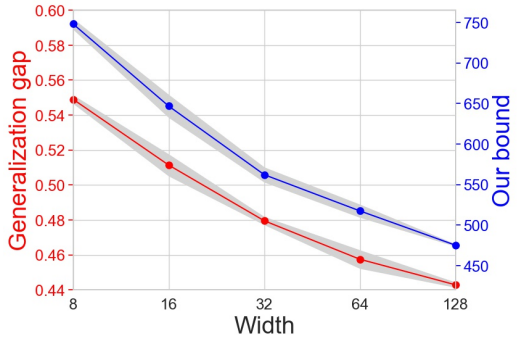

Figure 4: Comparison between the generalization gap and our generalization bound in Theorem 1. We use the SGLD algorithm to train CNNs with varying widths (i.e., number of filters in CNN) on CIFAR-10. As shown, both the generalization gap and our bound are decreasing w.r.t. the network width.

dataset	method	lr	width	depth	τ	Ψ	MI
CIFAR-10	OURS (THEOREM 1)	**0.41**	**0.93**	**1.00**	**0.45**	**0.78**	**0.25**
	NEGREA ET AL. (2019)	0.33	0.41	0.85	0.38	0.53	0.16

Table 2: We adopt the three evaluation criteria proposed in Jiang et al. (2019) for comparing our generalization bound with the benchmark method (Negrea et al., 2019): (i) Kendall's rank-correlation coefficient (τ), (ii) Granulated Kendall's coefficient (Ψ), and (iii) conditional independent test (MI). All scores, except MI, are within $[-1, 1]$ and the score of MI is normalized to $[0, 1]$. We also report the correlations when a single hyper-parameter (e.g., learning rate (lr)) is varying.

Parameter	Details
Dataset	MNIST
Number of training data	5000
Batch size	500
Learning rate	Initialization = 0.03, decay rate = 0.96, decay steps=2000
Inverse temperature	$\beta_t = 10^6/(2\eta_t)$
Architecture	MLP with ReLU activation
Depth	3 layers
Width	64 hidden units
Objective function	Cross-entropy loss
Loss function	0-1 loss

Table 3: Experiment details of Figure 1, 2 and Table 1. For Figure 2, the network width is varying among $\{16, 32, 64, 128, 256\}$ hidden units. For Table 1, we run the SGLD algorithm 600 epochs and vary three hyper-parameters: learning rate initialization $\in \{0.03, 0.06, 0.09\}$, depth $\in \{2, 3, 4\}$, and width $\in \{16, 32, 64\}$.

C Supporting Experimental Results

Recall that our generalization bound in Theorem 1 involves the variance of gradients. To estimate this quantity from data, we repeat our experiments 4 times and record the batch gradient at each iteration. This batch gradient is the one used for updating the parameters in the SGLD algorithm so it does not require any additional computations. Then we estimate the variance of gradients by using the population variance of the recorded batch gradients. Finally, we repeat the above procedure 4 times for computing the standard deviation, leading to e.g., the shaded areas in Figure 1. We provide experimental details in Table 3 and 4 for reproducing our experiments.

Parameter	Details
Dataset	CIFAR-10
Number of training data	5000
Batch size	500
Number of epochs	400
Learning rate	Initialization = 0.03, decay rate = 0.96, decay steps = 2000
Inverse temperature	$\beta_t = 10^6/(2\eta_t)$
Architecture	$\mathsf{conv}(5, 32)$ $\mathsf{pool}(2)$ $\mathsf{conv}(5, 32)$ $\mathsf{pool}(2)$ $\mathsf{fc}(120)$ $\mathsf{fc}(84)$ $\mathsf{fc}(10)$
Objective function	Cross-entropy loss
Loss function	0-1 loss

Table 4: Experiment details of Figure 3, 4 and Table 2. Here $\mathsf{conv}(k, w)$ is a $k \times k$ convolutional layer with w filters; $\mathsf{pool}(k)$ is a $k \times k$ max pooling layer; and $\mathsf{fc}(k)$ is a fully connected layer with k units. The convolutional layers and the fully connected layers all use ReLU activation function. For Figure 4, the network width (i.e., number of filters in CNN) is varying among $\{8, 16, 32, 64, 128\}$. For Table 2, we are varying three hyper-parameters: learning rate initialization $\in \{0.03, 0.06, 0.09\}$, number of convolutional layers $\in \{2, 3, 4\}$, and width $\in \{32, 64, 128\}$.