Analyzing the Generalization Capability of SGLD Using Properties of Gaussian Channels

Anonymous Author(s)
Affiliation
Address
email

Abstract

Understanding the generalization capability of learning algorithms is at the heart of statistical learning theory. In this paper, we investigate the generalization gap of stochastic gradient Langevin dynamics (SGLD), a widely used optimizer for training machine learning models. We derive a generalization bound by using properties of Gaussian channels which have been well understood in information and communication theory. Our bound incorporates the variance of gradients for quantifying a particular kind of "sharpness" of the loss landscape while maintaining $O(1/n)$ dependence on the sample size n. When we recast our bound in the setting of differentially private stochastic gradient descent, our bound includes an additional time-decaying factor which enables the impact of early iterations to reduce with time. This decay factor is established by strong data processing inequalities—a fundamental concept in information theory. We demonstrate our bound through numerical experiments, showing that it can predict the behavior of the true generalization gap.

1 Introduction

Stochastic gradient descent (SGD) and its variants are commonly used optimizers for training machine learning (ML) models. While their generalization behavior have been studied extensively for convex loss functions (see e.g., Lin et al., 2016; Orabona, 2014; Hardt et al., 2016; Yin et al., 2018; He et al., 2019; Cao and Gu, 2019; Bassily et al., 2020), the non-convex setting is not yet fully understood. There is an increasing number of recent efforts (see e.g., Raginsky et al., 2017; Mou et al., 2018; Li et al., 2019; Pensia et al., 2018; Negrea et al., 2019) to fill this gap by investigating the generalization properties of the stochastic gradient Langevin dynamics (SGLD) algorithm (Gelfand and Mitter, 1991; Welling and Teh, 2011)—a special variant of SGD. For example, Mou et al. (2018) proposed two generalization bounds, using algorithmic stability (Bousquet and Elisseeff, 2002) and PAC-Bayesian theory (McAllester, 1999) respectively. Pensia et al. (2018) considered a broad class of noisy iterative algorithms, including SGLD, through an information-theoretic view (Xu and Raginsky, 2017). It is within this body of work that the present paper is inscribed.

In this paper, we derive a new generalization bound for the SGLD algorithm. Our bound incorporates the variance of gradients while maintaining $O(1/n)$ dependence on the sample size n. This variance term can be estimated from the training data and captures a particular kind of "sharpness" of the loss landscape (Keskar et al., 2016). It 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 parameters across all iterations. This is crucial since many theoretical analyses (see e.g., Zhang et al., 2017; Jin et al., 2019) require the final output from the SGLD to be the parameter which achieves the smallest value of the loss function or is closest to a stationary point, but not necessarily the one at the last iteration.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

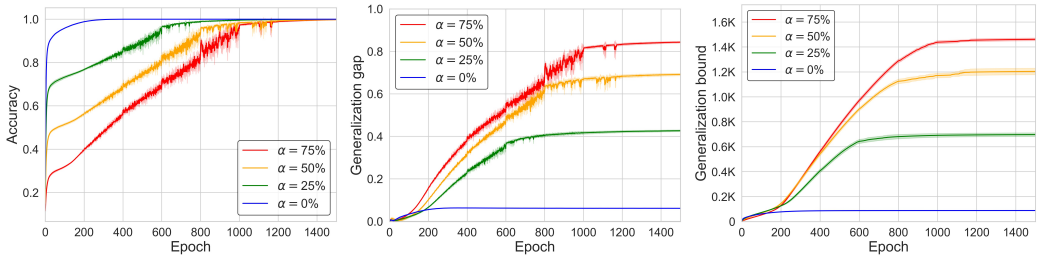

Figure 1: Illustration of the generalization bound in Theorem 2. We use SGLD algorithm to train 3-layer neural networks on MNIST when the training data have different label corruption level $\alpha \in \{0\%, 25\%, 50\%, 75\%\}$. Left: training accuracy. Middle: (empirical) generalization gap. Right: (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 4.

We also apply our bound to analyze the generalization capability of differentially private SGD (DP-SGD) algorithm (see e.g., Song et al., 2013; Feldman et al., 2018; Asoodeh et al., 2020; Balle et al., 2019), a popular SGD variant for protecting user privacy and closely related to SGLD. Recently, open-source libraries, including Opacus (Facebook AI, 2020) and TensorFlow Privacy (Radebaugh and Erlingsson, 2019), have been announced for training ML models with differential privacy (Dwork et al., 2006) guarantees. The DP-SGD algorithm often requires additional assumptions. For example, the training data are drawn without replacement for preventing information leakage and the loss function is Lipschitz continuous for controlling the sensitivity of each update. By leveraging these extra assumptions, we incorporate a time-decaying factor into our generalization bound which enables the impact of early iterations to reduce with time.

The goal of this paper is to derive a generalization bound that can help understand recent empirical observations which are not explained by uniform notions of hypothesis class complexity (Vapnik and Chervonenkis, 1971; Valiant, 1984). For example, a neural network trained using true labels exhibits better generalization ability than a network trained using corrupted labels even when the network architecture is fixed and perfect training accuracy is achieved (Zhang et al., 2016). Distribution-independent bounds may not be able to capture this phenomenon because they are invariant for both true data and corrupted data. In contrast, our bound quantifies this empirical observation, exhibiting a lower value on networks trained on true labels compared to ones trained on corrupted labels (Fig. 1). Another example is that a wider network often has a more favourable generalization capability (Neyshabur et al., 2014). This may seem counter-intuitive at first glance since one may expect that wider networks have a higher VC-dimension and, consequently, would generalize worse. Our bound captures this behaviour and is decreasing with respect to the neural network width (Fig. 2).

The proof techniques of this paper are based on fundamental tools from information theory. We first utilize an information-theoretic framework, proposed by 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. Unfortunately, estimating this mutual information, in general, is intractable since the underlying distribution is unknown. To circumvent this issue, we connect $I(\mathrm{W};\mathrm{Z}_i)$ with a well-understood notion in data transmission, namely additive white Gaussian noise (AWGN) channels. This allows us to use properties of Gaussian channels for analyzing the mutual information. Specifically, we first 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 more reliably estimated from the training data and is highly correlated with the true generalization gap. Second, we incorporate a time-decaying factor into our bound when analyzing the DP-SGD algorithm. 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 contribution to the generalization error keeps on reducing with time due to external Gaussian noise. The above two aspects correspond to Lemma 4, 5 which are the basis of our main results in Theorem 1, 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, 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 since the Lipschitz constant may be large and fails to explain some 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 the 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 is significant recent work which adopts 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 by Bu et al. (2020) in Proposition 3. 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 so-called 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 4) suggest that our bound seems to be more favourably correlated with the true generalization gap. Negrea et al. (2019) also provided a bound which involves a variance term in their Theorem 3.1 albeit with different forms. Nonetheless, this bound is looser than their gradient-prediction-residual bound since the dependence on the number of samples is $O(1/\sqrt{n})$ instead of $O(1/n)$.

More broadly, there are additional related works considering algorithms closely related with 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 experiment. Specifically, they introduced a quantity, namely the squared error probability of the hypothesis test, to their bound in Theorem 4.2 which 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 (2021) investigated the generalization properties of SGD and provided a bound that also involves the variance of gradients. However, 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 which 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 decay factor is much slower.

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:

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

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

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

We project $\mathsf{Proj}_{\mathcal{W}}(\boldsymbol{w}) \triangleq \operatorname{argmin}_{\boldsymbol{w}'\in\mathcal{W}} \|\boldsymbol{w}' - \boldsymbol{w}\|_2$ the parameter onto the domain $\mathcal{W}$ at each iteration. This projection operator can help deal with constrained optimizations or serve as a regularization step. The recursion in (1) is run 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 $\mathrm{W} = \mathrm{W}_T$ or the one which achieves the smallest value of the loss function $\mathrm{W} = \operatorname{argmin}_{\mathrm{W}_i} L_\mu(\mathrm{W}_i)$.

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 an 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 $I(\mathrm{U};\mathrm{Y}) \leq I(\mathrm{U};\mathrm{X})$. 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): $\eta_f(P_{\mathrm{Y}|\mathrm{X}}) \leq \eta_{\mathrm{TV}}(P_{\mathrm{Y}|\mathrm{X}})$. Furthermore, for any Markov chain $\mathrm{U} \to \mathrm{X} \to \mathrm{Y}$, the contraction coefficient of KL-divergence satisfies (Ahlswede and Gács, 1976)

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

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.

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 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 (Polyanskiy and Wu, 2016)

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

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* $m > 0$

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

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 (see Section 2.2.6 in Guo, 2004, for a proof) 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* $m > 0$

$$I(\mathrm{X} + m\mathrm{N}; \mathrm{X}) \leq \frac{1}{2m^2}\mathbb{E}\left[\|\mathrm{X}\|_2^2\right]. \tag{9}$$

3 Algorithm-Dependent Generalization Bounds

In this section, we derive a generalization bound for the SGLD algorithm (Theorem 2) and apply it to the DP-SGD algorithm (Theorem 1). In the latter case, our generalization bound incorporates an additional time-decaying factor which lets the impact of information from the data points used in the early iterations decay with time. This decay factor is established by strong data processing inequalities (Lemma 4). In order to obtain a computable bound, we further upper bound the mutual information by employing properties of Gaussian channels (Lemma 5). We end this section by 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 and the sub-Gaussian constant $\sigma = (b-a)/2$.

We first consider the DP-SGD algorithm which is a special instance of the SGLD algorithm. Specifically, we assume that the training data are drawn without replacement and the algorithm outputs the parameter at the last iteration (i.e., $\mathrm{W} = \mathrm{W}_T$).

Recall that Lemma 1 provides a generalization bound in terms of the mutual information $I(\mathrm{W}_T; \mathrm{Z}_i)$. Intuitively speaking, if a data point Z_i is used at an early iteration, $I(\mathrm{W}_T; \mathrm{Z}_i)$ 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.

Lemma 4. *If data are drawn without replacement and the data point* Z_i *is used at the* t*-th iteration, then we have*

$$I(\mathrm{W}_T; \mathrm{Z}_i) \leq I(\mathrm{W}_t; \mathrm{Z}_i) \cdot \prod_{t'=t+1}^{T} q_{t'}. \tag{10}$$

Here each component in the decay-factor is defined as

$$q_{t'} \triangleq 1 - 2\bar{\Phi}\left((D + 2\eta_{t'}K)\sqrt{\frac{\beta_{t'}}{8\eta_{t'}}}\right) \in [0, 1] \tag{11}$$

where $\bar{\Phi}(\cdot)$ *is the Gaussian CCDF, the constants* D *and* K *are the diameter of the parameter domain and a uniform upper bound of* $\|\nabla_{\boldsymbol{w}}\hat{\ell}(\boldsymbol{w}, \boldsymbol{z})\|_2$, *respectively.*

Lemma 4 explains why our generalization bound in Theorem 1 incorporates a time-decaying factor $\prod q_{t'}$. If the parameter domain is compact and the loss function is Lipschitz continuous, then each $q_{t'}$ in (11) is strictly upper bounded by 1. In this case, one can even prove that $I(\mathrm{W}_T; \mathrm{Z}_i) \to 0$ as $T \to \infty$ if the magnitude of the Gaussian noise in (1) has a lower bound. The above two assumptions (i.e., compact domain and Lipschitz loss function) are often satisfied by the DP-SGD algorithm (Feldman et al., 2018; Asoodeh et al., 2020; Balle et al., 2019). Furthermore, we note that the decay factor also plays an important role for amplifying the privacy guarantees (Asoodeh et al., 2020).

Since the underlying data distribution is unknown, so is the mutual information $I(\mathrm{W}_t; \mathrm{Z}_i)$ in (10), which poses a problem for computing the generalization bound. In order to circumvent this issue, we further upper bound the mutual information by using the variance of gradients, which can be more reliably estimated from the training set.

Lemma 5. *If data are drawn without replacement and the data point* Z_i *is used at the* t*-th iteration, then we have*

$$I(\mathrm{W}_t; \mathrm{Z}_i) \leq \frac{\beta_t \eta_t}{b^2} \cdot \mathsf{Var}\left(\nabla_{\boldsymbol{w}}\hat{\ell}(\mathrm{W}_{t-1}, \mathrm{Z})\right), \tag{12}$$

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

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

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

The variance of gradients in (12) measures a particular kind of "sharpness" of the loss landscape. A recent work (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. With Lemma 4, 5 in hand, we now present the first main result—a generalization bound for the DP-SGD algorithm.

Theorem 1. *Under Assumption 1, if the SGLD algorithm outputs the parameter at the last iteration (i.e.,* $\mathrm{W} = \mathrm{W}_T$*) and data are drawn without replacement, the expected generalization gap* (3) *can be upper bounded by*

$$\frac{\sqrt{2}\sigma}{n} \sum_{t=1}^{T} \sqrt{\beta_t \eta_t \cdot \mathsf{Var}\left(\nabla_{\boldsymbol{w}} \hat{\ell}(\mathrm{W}_{t-1}, \mathrm{Z})\right) \cdot \prod_{t'=t+1}^{T} q_{t'}},$$

where $q_{t'}$ *is defined in* (11).

Next, we present a general result without the conditions made in Theorem 1. Its proof relies on the chain rule for mutual information in order to deal with the case where data are drawn with replacement. Unfortunately, our generalization bound does not incorporate a decay factor any more[3]. To understand why it happens, let us imagine an extreme scenario in which the SGLD algorithm outputs the parameters across all iterations (i.e., $\mathrm{W} = (\mathrm{W}_1, \cdots, \mathrm{W}_T)$). The data processing inequality implies that for a data point Z_i which is used at the t-th iteration, the mutual information $I(\mathrm{W}_1, \cdots, \mathrm{W}_T; \mathrm{Z}_i)$ is always lower bounded by $I(\mathrm{W}_t; \mathrm{Z}_i)$. Hence, it is impossible to have $I(\mathrm{W}_1, \cdots, \mathrm{W}_T; \mathrm{Z}_i) \to 0$ as $T \to \infty$ in contrast with Lemma 4.

Theorem 2. *Under Assumption 1, the expected generalization gap* (3) *of the SGLD algorithm can be upper bounded by*

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

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

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

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 additive noise.

We note that many existing SGLD generalization bounds are expressed as a sum of errors associated with each training iteration. Here we present an analogous bound but we remark that it may be much weaker than Theorem 2 since it is obtained by combining Theorem 2 with Minkowski inequality and Jensen's inequality.

Corollary 1. *Under Assumption 1, the expected generalization gap 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.*

[3] We note that the analysis in Mou et al. (2018) requires $\mathrm{W} = \mathrm{W}_T$. Hence, in the setting we consider (i.e., W is a function of $\mathrm{W}_1, \cdots, \mathrm{W}_T$), it is uncertain if it is still possible to have a decay factor in the bound.

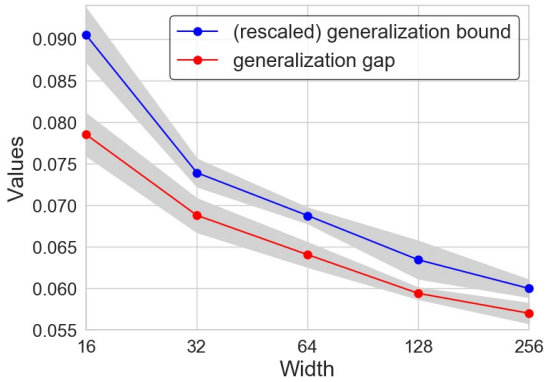

Figure 2: The generalization gap of 3-layer neural networks with varying widths trained on MNIST using SGLD algorithm. To compare with, We plot the generalization bound (Theorem 2) rescaled by multiplying 7e-4.

Theorem 2 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 a 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* $|L_\mu(\mathrm{W}) - L_{\mathrm{S}}(\mathrm{W})|$ *can be upper bounded by*

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

4 Numerical Experiments

In this section, we demonstrate our generalization bound in Theorem 2 through numerical experiments on the MNIST dataset. Our experiments are in the similar spirit of Jiang et al. (2019); Neyshabur et al. (2017); Dziugaite et al. (2020). First, we validate the ability of our generalization bound in explaining 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) and provide experimental results in the supplementary material. Source code for reproducing our experimental results is given at [URL REDACTED FOR BLIND REVIEW].

Corrupted label. As observed in Zhang et al. (2016), DNNs have the potential to memorize the entire training dataset even when the labels are corrupted by a large portion. For networks with identical architecture, those learned using true labels own better generalization capability than those ones learned 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, when the corruption level is increasing, both our bound and the generalization gap are increasing. Furthermore, the curve of our bound has a very similar shape to the generalization gap.

Network width. As observed by several recent studies (see e.g., Neyshabur et al., 2014; Jiang et al., 2019), wider networks could lead to a smaller generalization gap. This may seem contradictory to

dataset	method	lr	width	depth	τ	Ψ	MI
MNIST	OURS (THEOREM 2)	**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.

the traditional wisdom as one may expect that a class of wider networks has a higher VC-dimension and, hence, would generalize worse. In our experiment, we use the SGLD algorithm to train neural networks with different widths. The training process continues 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. We compare Theorem 2 with other SGLD generalization bounds in the literature. We note that many existing approaches may either lack comprehensive empirical evaluations (Mou et al., 2018; Pensia et al., 2018; Rodríguez-Gálvez et al., 2020) or require additional assumptions (Li et al., 2019). To the best of our knowledge, the bound proposed in Negrea et al. (2019) is the only one which has been numerically tested while relying on mild assumptions. Hence, we use it as a benchmark method.

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 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 benchmark (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.

5 Summary

We provide a new generalization bound for the SGLD algorithm in this paper. Our hope is that this bound can help explain some empirical observations (e.g., why over-parameterized DNNs often generalize well in practice) and inspire new regularization techniques. The proof techniques in this paper rely on information-theoretic tools (e.g., strong data processing inequalities and properties of Gaussian channels), which have a long history in that field. We wish that the community studying generalization theory can benefit from these fundamental tools. Our approach can be extended in several directions. For example, one could tighten our time-decaying factor in Theorem 1 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.

References

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*.

Bassily, R., Feldman, V., Guzmán, C., and Talwar, K. (2020). Stability of stochastic gradient descent on nonsmooth convex losses. *arXiv preprint arXiv:2006.06914*.

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.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *J. Mach. Learn. Res.*, 2(Mar):499–526.

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*.

Cao, Y. and Gu, Q. (2019). Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10836–10846.

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.

Dziugaite, G. K., Drouin, A., Neal, B., Rajkumar, N., Caballero, E., Wang, L., Mitliagkas, I., and Roy, D. M. (2020). In search of robust measures of generalization. *arXiv preprint arXiv:2010.11924*.

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. (2004). Gaussian channels: Information, estimation and multiuser detection.

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.

He, F., Liu, T., and Tao, D. (2019). Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In *Advances in Neural Information Processing Systems*, pages 1143–1152.

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*.

Lin, J., Camoriano, R., and Rosasco, L. (2016). Generalization properties and implicit regularization for multiple passes SGM. In *International Conference on Machine Learning*, pages 2340–2348.

McAllester, D. A. (1999). Some PAC-Bayesian theorems. *Mach. Learn.*, 37(3):355–363.

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.

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. (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*.

Orabona, F. (2014). Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124.

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 Trans. Inf. Theory*, 62(1):35–55.

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*.

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.

Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K., and Bartlett, P. (2018). Gradient diversity: a key ingredient for scalable distributed learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1998–2007. PMLR.

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.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This is a theoretical work studying generalization theory. We do not see any direct negative societal impact of this work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See the supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]