Non-Vacuous Generalisation Bounds for Shallow Neural Networks

Felix Biggs [1] **Benjamin Guedj** [1]

Abstract

We focus on a specific class of shallow neural networks with a single hidden layer, namely those with L_2-normalised data and either a sigmoid-shaped Gaussian error function ("erf") activation or a Gaussian Error Linear Unit (GELU) activation. For these networks, we derive new generalisation bounds through the PAC-Bayesian theory; unlike most existing such bounds they apply to neural networks with *deterministic* rather than randomised parameters. Our bounds are empirically non-vacuous when the network is trained with vanilla stochastic gradient descent on MNIST, Fashion-MNIST, and binary classification versions of the above.

1. Introduction

The study of generalisation properties of deep neural networks is arguably one of the topics gaining most traction in deep learning theory (see, *e.g.*, the recent surveys Kawaguchi et al., 2020; Jiang et al., 2020b). In particular, a characterisation of out-of-sample generalisation is essential to understand where trained neural networks are likely to succeed or to fail, as evidenced by the recent NeurIPS 2020 competition "Predicting Generalization in Deep Learning" (Jiang et al., 2020a). One stream of this joint effort, which the present paper contributes to, is dedicated to the study of shallow neural networks, potentially paving the way to insights on deeper architectures.

Despite numerous efforts in the past few years, non-vacuous generalisation bounds for *deterministic* neural networks with many more parameters than data remain generally elusive. Those few non-vacuous bounds that exist primarily report bounds for networks with randomised parameters, for example Gaussian weights, which are re-drawn for every prediction (a non-exhaustive list of references would begin with Dziugaite & Roy, 2017; 2018; Neyshabur et al., 2017; 2018; Hellström & Durisi, 2021), or for compressed versions of the trained networks (Zhou et al., 2019). While these undoubtedly advanced knowledge on generalisation in deep learning theory, this is far from contemporary practice which generally focuses on deterministic networks obtained directly through stochastic gradient descent (SGD), as we do.

The PAC-Bayesian theory (we refer to the recent Guedj, 2019 and Alquier, 2021 for a gentle introduction) is thus far the only framework within which non-vacuous bounds have been provided for networks trained on common classification tasks. Given its focus on randomised or "Gibbs" predictors, the aforementioned lack of results for deterministic networks is unsurprising. However, the framework is not limited to such results: one area within PAC-Bayes where deterministic predictors are often considered lies in a range of results for the "majority vote", or the expected overall prediction of randomised predictors, which is itself deterministic.

Computing the average output of deep neural networks with randomised parameters is generally intractable: therefore most such works have focused on cases where the average output is simple to compute, as for example when considering linear predictors. Here, building on ideas from Biggs & Guedj (2022), we show that provided our predictor structure factorises in a particular way, more complex majority votes can be constructed. In particular, we give formulations for randomised predictors whose majority vote can be expressed as a deterministic single-hidden-layer neural network. Through this, we obtain classification bounds for these *deterministic* predictors that are non-vacuous on the celebrated baselines MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and binarised versions of the above. We believe these are the first such results.

Our work fundamentally relates to the question: what kind of properties or structures in a trained network indicate likely generalisation to unseen data? It has been shown by Zhang et al. (2017) that neural networks trained by SGD can perfectly overfit large datasets with randomised labels, which would indicate a lack of capacity control, while simultaneously generalising well in a variety of scenarios. Thus, clearly any certification of generalisation must involve ex-

[1]Centre for Artificial Intelligence and Department of Computer Science, University College London and Inria London, UK. Correspondence to: Felix Biggs <contact@felixbiggs.com>, Benjamin Guedj <b.guedj@ucl.ac.uk>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

tracting additional information other than the train loss—for example, the specific final network chosen by SGD. How do the final parameters of a neural network trained on an "easy" data distribution as opposed to a pathological (*e.g.*, randomised label) one differ? A common answer to this has involved the return of capacity control and the norms of the weight matrices, often measured as a distance to the initialisation (as done, *e.g.*, in Dziugaite & Roy, 2017; Bartlett et al., 2017; Neyshabur et al., 2018).

We suggest, following insights from Dziugaite et al. (2021), that a better answer lies in utilising the empirically-observed stability of SGD on easy datasets. We give bounds that are tightest when a secondary run of SGD on some subset of the training set gives final weights that are close to the full-dataset derived weights. This idea combines naturally in the PAC-Bayes framework with the requirement of perturbation-robustness of the weights—related to the idea of flat-minima (Hinton & van Camp, 1993; Hochreiter & Schmidhuber, 1997)—to normalise the distances between the two runs. By leveraging this commonly-observed empirical form of stability we effectively incorporate information about the inherent easiness of the dataset and how adapted our neural network architecture is to it. Although it is a deep and interesting theoretical question as to when and why such stability occurs under SGD, we believe that by making the link to generalisation explicit we solve some of the puzzle.

Setting. We consider D-class classification on a set $\mathcal{X} \subset \mathbb{R}^d$ with "score-output" predictors returning values in $\hat{\mathcal{Y}} \subset \mathbb{R}^D$ with multi-class label space $\mathcal{Y} = [D]$, or in $\hat{\mathcal{Y}} = \mathbb{R}$ with binary label space $\mathcal{Y} = \{+1, -1\}$. The prediction is the argmaximum or sign of the output and the misclassification loss is defined as $\ell(f(x), y) = \mathbf{1}\{\mathrm{argmax}_{k\in[D]}\, f(x)[k] \neq y\}$ or $\ell(f(x), y) = \mathbf{1}\{yf(x) \leq 0\}$ respectively. It is will prove useful that scaling does not enter into these losses and thus the outputs of classifiers can be arbitrarily re-scaled by $c > 0$ without affecting the predictions. We write $L(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\ell(f(x), y)$ and $\hat{L}(f) := m^{-1}\sum_{(x,y)\in S}\ell(f(x), y)$ for the risk and empirical risk of the predictors with respect to data distribution $\mathcal{D}$ and i.i.d. m-sized sample $S \sim \mathcal{D}^m$.

Overview of our contributions. We derive generalisation bounds for a single-hidden-layer neural network $F_{U,V}$ with first and second layer weights U and V respectively taking the form

$$F_{U,V}(x) = V\,\phi\left(\beta\frac{Ux}{\|x\|_2}\right)$$

with ϕ being an element-wise activation. If the data is normalised to have $\|x\|_2 = \beta$ these are simply equivalent to one-hidden-layer neural networks with activation ϕ and the given data norm. We provide high-probability bounds on $L(F_{U,V})$ of the approximate form

$$2\mathbb{E}_{f\sim Q}\hat{L}(f) + \mathcal{O}\left(\frac{\beta\|U - U^n\|_F + \|V - V^n\|_F}{\sqrt{m-n}}\right),$$

where Q is a distribution over predictors f, which depends on U and V but does not necessarily take the form of a neural network. The construction of this randomised proxy Q is central to our PAC-Bayes derived proof methods. The bounds hold uniformly over any choice of weight matrices, but for many choices the bounds obtained will be vacuous; what is interesting is that they are non-vacuous for SGD-derived solutions on some real-world datasets. U^n and V^n are matrices constructed using some subset $n < m$ of the data. Since we consider SGD-derived weights, we can leverage the empirical stability of this training method (through an idea introduced by Dziugaite et al., 2021) to construct U^n, V^n which are quite close to the final true SGD-derived weights U, V, essentially by training a prior on the n-sized subset in the same way.

Outline. In Section 2 we give an overview of results from previous works which we use. In Section 3 we give a bound on the generalisation error of binary classification SHEL networks, which are single hidden layer networks with "erf" activations. In Section 4 we extend to multi-class classification using a simple assumption, giving a general formulation as well as results for "erf"- and GELU-activated networks. In Section 5 we discuss our experimental setting and give our numerical results, which we discuss along with future work in Section 6.

2. Background and Related Work

PAC-Bayesian bounds. Originated by McAllester (1998; 1999), these generally consider the expected loss or Gibbs risk $L(Q) := \mathbb{E}_{f\sim Q}L(f)$ and analogously for the empirical risk, where $Q \in \mathcal{M}_1^+(\mathcal{F})$ (with $\mathcal{M}_1^+(\mathcal{A})$ denoting the set of measures on $\mathcal{A}$) is a distribution over randomised predictors $f \in \mathcal{F}$. The high-probability bounds take the rough form (although numerous variations using variance terms or attaining fast rates also exist – see the aforecited Guedj, 2019 and Alquier, 2021 for a survey)

$$L(Q) \leq \hat{L}(Q) + \mathcal{O}\left(\sqrt{\frac{\mathrm{KL}(Q, P) + \log(1/\delta)}{m}}\right) \quad (1)$$

holding with at least $1-\delta$ probability over the draw of the dataset. Here $\mathrm{KL}(Q, P)$ is the Kullback-Leibler divergence and $P \in \mathcal{M}_1^+(\mathcal{F})$ is the PAC-Bayesian "prior" distribution, which must be chosen in a data-independent way (but is not subject to the same requirements as a standard Bayesian prior for the validity of the method). This bound holds over all "posterior" distributions Q, but a poor choice (for example, one over-concentrated on a single predictor) will

lead to a vacuous bound. We note in particular the following, which we use to prove our main results.

Theorem 2.1. Langford & Seeger (2001), Maurer (2004). *Given data distribution $\mathcal{D}$, $m \in \mathbb{N}^+$, prior $P \in \mathcal{M}_1^+(\mathcal{F})$, and $\delta \in (0,1)$, with probability $\geq 1-\delta$ over $S \sim D^m$, for all $Q \in \mathcal{M}_1^+(\mathcal{H})$*

$$L(Q) \leq \mathrm{kl}^{-1}\left(\hat{L}(Q),\ \frac{1}{m}\left(\mathrm{KL}(Q,P) + \log\frac{2\sqrt{m}}{\delta}\right)\right)$$

where $\mathrm{kl}^{-1}(u,c) := \sup\{v \in [0,1] : \mathrm{kl}(u,v) \leq c\}$ *and* $\mathrm{kl}(q:p) := q\log(q/p) + (1-q)\log((1-q)/(1-p))$.

We note the relaxation $\mathrm{kl}^{-1}(u,c) \leq u + \sqrt{c/2}$ which gives an idea of the behaviour of Theorem 2.1; however in the case of u close to 0 the original formulation is considerably tighter.

Data-Dependent Priors. A careful choice of the prior is essential to the production of sharp PAC-Bayesian results. A variety of works going back to Ambroladze et al. (2006) and Parrado-Hernández et al. (2012) (and further developed by Dziugaite & Roy, 2018; Dziugaite et al., 2021; Rivasplata et al., 2018; Perez-Ortiz et al., 2021a;b, among others) have considered dividing the training sample into two parts, one to learn the prior and another to evaluate the bound. Formally, we divide $S = S^{\mathrm{prior}} \cup S^{\mathrm{bnd}}$ and use S^{prior} to learn a prior P^n where $n = |S^{\mathrm{prior}}|$, then apply the PAC-Bayesian bound using sample S^{bnd} to a posterior Q learned on the entirety of S. The resulting bound replaces $\hat{L}$ by $\hat{L}_{\mathrm{bnd}}$, P by the data-dependent P^n, and m by $m - n = |S^{\mathrm{bnd}}|$; thus the KL complexity term may be reduced at the cost of a smaller dataset to apply the bound to.

Dziugaite et al. (2021) used this when considering training neural networks by constructing a so-called "coupled" prior P^n which is trained in the same way from the same initialisation as the posterior Q by stochastic gradient descent with the first n examples from the training set forming one epoch. Due to the stability of gradient descent, the weights of P^n and Q evolve along similar trajectories; thus stability of the training algorithm is leveraged to tighten bounds without explicit stability results being required (and we do not study the conditions under which SGD provides such solutions). In many ways this can be seen as an extension of previous work such as Dziugaite & Roy (2017) relating generalisation to the distance from initialisation rather than total weight norms.

Majority Votes. Since PAC-Bayesian bounds of the form in (1) and Theorem 2.1 generally consider the risk of randomised predictors, a natural question is whether prediction accuracy can be improved by "voting" many independently drawn predictions; such a majority vote predictor takes the deterministic form $\mathrm{MV}_Q(x) := \mathrm{argmax}_k\, \mathbb{P}_{f\sim Q}(\mathrm{argmax}\, f(x) = k)$. Several strategies have been devised to obtain bounds for these predictors via PAC-Bayesian theorems, with the simplest (and often most successful) being the unattributed first-order bound $\ell(\mathrm{MV}_Q(x), y) \leq 2\mathbb{E}_{f\sim Q}\ell(f(x), y)$ valid for all (x,y), called the "folk theorem" by Langford & Shawe-Taylor (2003) and the *first-order* bound elsewhere. This can be substituted directly into PAC-Bayesian theorems such as Theorem 2.1 above to obtain bounds for the majority vote at a de-randomisation cost of a factor of two. This is the result we use, since across a variety of preliminary experiments we found other strategies including the tandem bound of Masegosa et al. (2020) and the C-bound of Lacasse et al. (2006) were uniformly worse, as also discussed by Zantedeschi et al. (2021).

Gaussian Sign Aggregation. To exploit the useful relationship above, Germain et al. (2009) considered aggregating a kind of linear prediction function of the form $f(x) = \mathrm{sign}(w \cdot x)$ with $w \sim Q = N(u, I)$. In this case the aggregation can be stated in closed form using the Gaussian error function "erf" as

$$\mathbb{E}_{w\sim N(u,I)}\, \mathrm{sign}(w \cdot x) = \mathrm{erf}\left(\frac{u \cdot x}{\sqrt{2}\|x\|_2}\right). \qquad (2)$$

This closed-form relationship has been used since by Letarte et al. (2019) and Biggs & Guedj (2021) in a PAC-Bayesian context for neural networks with sign activation functions and Gaussian weights; Biggs & Guedj (2022) used it to derive a generalisation bound for SHEL (single hidden erf layer) networks, which have a single hidden layer with erf activation function. We will consider deriving a different PAC-Bayesian bound for this same situation and develop this method further in this work.

Other Approaches. A wide variety of other works have derived generalisation bounds for deterministic neural networks without randomisation. We note in particular the important works of Bartlett et al. (2017), Neyshabur et al. (2017) (using PAC-Bayesian ideas in their proofs) and Arora et al. (2018), but contrary to us, they do not provide empirically non-vacuous bounds. Nagarajan & Kolter (2019a) de-randomise PAC-Bayesian bounds by leveraging the notion of noise-resilience (how much the training loss of the network changes with noise injected into the parameters), but they note that in practice their bound would be numerically large. Many of these approaches utilise uniform convergence, which may lead to shortcomings as discussed at length by Nagarajan & Kolter (2019b); we emphasise that the bounds we give are non-uniform and avoid these shortcomings. Finally, we also highlight the works of Neyshabur et al. (2015; 2019) which specifically consider single-hidden-layer networks as we do – as in the recent study from Tinsi & Dalalyan (2021). Overall we emphasise that, to the best of our knowledge, all existing bounds for deterministic networks are vacuous when networks are

trained on real-world data.

3. Binary SHEL Network

We begin by giving a bound for binary classification by a single hidden layer neural network with error function ("erf") activation. Binary classification takes $\mathcal{Y} = \{+1, -1\}$, with prediction the sign of the prediction function. The specific network takes the following form with output dimension $D = 1$. Although the erf activation function is not a commonly-used one, it is very close in value to the more common $\tanh$ activation. It can also be rescaled to a Gaussian CDF activation, which is again very close to the classical sigmoid activation (and is itself the CDF of the probit distribution).

Definition 3.1. *SHEL Network.* (Biggs & Guedj, 2022) For $U \in \mathbb{R}^{K\times d}$, $V \in \mathbb{R}^{K\times D}$, and $\beta > 0$, a β-normalised single hidden erf layer (SHEL) network is defined by

$$F^{\mathrm{erf}}_{U,V}(x) := V \cdot \mathrm{erf}\left(\beta \frac{Ux}{\|x\|_2}\right).$$

The above is a single-hidden-layer network with a first normalisation layer, or if the data is already normalised the overall scaling $\|x\|_2$ can be absorbed into the β parameter. This parameter β could easily be absorbed into the matrix U and mainly has the effect of scaling the relative learning rate for U versus V when training by gradient descent, as shown by looking at $\frac{\partial}{\partial U} F^{\mathrm{erf}}_{U,V}(x)$, something which would normally be affected by the scaling of data. A higher β means more "feature learning" takes place as U has a relatively larger learning rate.

For binary classification, the majority vote of distribution Q is $\mathrm{MV}_Q(x) = \mathrm{sign}(\mathbb{E}_{f\sim Q}\,\mathrm{sign}(f(x)))$. By expressing the (binary classification) SHEL network directly as the majority vote of a randomised prediction function, we can prove a PAC-Bayesian generalisation bound on its error using the first-order bound. The misclassification error of the randomised function can further be stated in closed form using the Binomial cumulative distribution function (CDF), giving rise to a bound where the distribution Q does not appear directly.

Theorem 3.2. *In the binary setting, fix prior parameters $u^0_1, \ldots, u^0_K \in \mathbb{R}^d, v^0 \in \mathbb{R}^K$, $T \in \mathbb{N}^+$, $\beta > 0$, and data distribution $\mathcal{D}$. For $\delta \in (0,1)$, with probability at least $1-\delta$ under the sample $S \sim \mathcal{D}^m$, simultaneously for any $U \in \mathbb{R}^{K\times d}, v \in \mathbb{R}^K$,*

$$L(F^{\mathrm{erf}}_{U,v}) \le 2\ \mathrm{kl}^{-1}\left(\hat{L}(Q^{\otimes T}),\ \frac{T\kappa + \log\frac{2\sqrt{m}}{\delta}}{m}\right).$$

Here $F^{\mathrm{erf}}_{U,v}$ is a SHEL network with β-normalised activation,

$$\kappa := \sum_{k=1}^{K} \frac{|v_k|}{\|v\|_1}\left(\beta^2\|u_k - u^0_k\|_2^2 + \log\left(2\frac{|v_k|/\|v\|_1}{|v^0_k|/\|v^0\|_1}\right)\right)$$

and

$$\hat{L}(Q^{\otimes T}) := \frac{1}{m}\sum_{(x,y)\in S} \mathrm{Bin}\left(\frac{T}{2}; T, \frac{1}{2}\left(1 + \frac{yF_{U,v}(x)}{\|v\|_1}\right)\right),$$

for $\mathrm{Bin}(k; r, p)$ the CDF of a Binomial distribution with parameters r, p.

4. Multi-class Networks

We now go further and show that various single-hidden-layer multi-class neural networks can also be expressed as the expectation of randomised predictors. We show specific results for multi-class SHEL networks as well as GELU-activation (Hendrycks & Gimpel, 2016) networks as defined below. We also give a more general form of the result as a aggregation of individual aggregated predictors which allows these results to be extended further.

We make a simple assumption based on the first-order bound to extend PAC-Bayesian bounds to this case. This is necessary because under certain choices of PAC-Bayes posterior Q, the *majority vote* does not give the same prediction as the *expected vote* as was the case in Section 3, i.e. there exist Q such that $\mathrm{argmax}_k\, \mathbb{E}_{f\sim Q} f(x)[k] \neq \mathrm{MV}_Q(x)$ at certain adversary-chosen values of x. Thus we assume that $L(\mathbb{E}_{f\sim Q} f(x)) \le 2\mathbb{E}_{f\sim Q} L(f)$, (denoted $\star$), which follows from the first order bound in the case $\mathbb{E}_Q f(x) \approx \mathrm{MV}_Q(x)$, which we later verify empirically.

4.1. SHEL Networks

Here we give a generalisation bound for a multi-class variant of the SHEL network using the above assumption. The proof is slightly different from the binary case, but still relies on the useful fact that the SHEL network can be written as the expectation of a randomised predictor. This predictor however takes a slightly different form to that in the binary case.

Theorem 4.1. *In the multi-class setting, fix prior parameters $U^n \in \mathbb{R}^{K\times d}$ and $V^n \in \mathbb{R}^{D\times K}$, $\sigma_V > 0$, $\beta > 0$, and data distribution $\mathcal{D}$. For $\delta \in (0,1)$, with probability at least $1-\delta$ under the sample $S \sim \mathcal{D}^m$, simultaneously for any $U \in \mathbb{R}^{K\times d}, V \in \mathbb{R}^{D\times K}$ such that assumption ($\star$) is satisfied,*

$$L(F^{\mathrm{erf}}_{U,V}) \le 2\ \mathrm{kl}^{-1}\left(\hat{L}(Q),\ \frac{\kappa + \log\frac{2\sqrt{m}}{\delta}}{m}\right).$$

Here $F^{\text{erf}}_{U,V}$ is a SHEL network with β-normalised activation,

$$\kappa := \beta^2 \|U - U^0\|_F^2 + \frac{\|V - V^0\|_F^2}{2\sigma_V^2},$$

and

$$\hat{L}(Q) := \frac{1}{m} \sum_{(x,y)\in S} \mathbb{P}\left\{\text{argmax}\left[W_2 \,\text{sign}(W_1 x)\right] \neq y\right\},$$

with the probability over draws of $\text{vec}(W_2) \sim N(\text{vec}(V), \sigma_V^2 I), \text{vec}(W_1) \sim N(\text{vec}(U), \frac{1}{2}\beta^{-2} I)$. *Note that* vec *is the vectorisation operator and* sign *is applied element-wise.*

Differences to Biggs & Guedj (2022). In their Theorem 5, Biggs & Guedj (2022) give a bound for generalisation in SHEL networks, with $L(F^{\text{erf}}_{U,V})$ upper bounded under similar conditions to Theorem 4.1 by

$$\hat{L}^{\gamma}(F^{\text{erf}}_{U,V}) + \tilde{O}\left(\frac{\sqrt{K}}{\gamma\sqrt{m}}\left(V_\infty \|U - U^0\|_F + \|V\|_F\right)\right),$$

where $\hat{L}^{\gamma}(g) = m^{-1}|\{(x,y) \in S : g(x)[y] - \max_{k\neq y} g(x)[k] \leq \gamma\}|$, the proportion of γ-margin errors in the training set, and $V_\infty := \max_{ij} |V_{ij}|$. Thus a margin loss of the actual predictor used rather than a stochastic one appears. A tighter formulation more similar to Theorem 4.1 is also given in an appendix and the bound could be similarly adapted to a data-dependent prior.

The derivation of the bound is quite different from ours, relying on a quite differently-constructed randomised version of Q (which is however constructed to have mean $F^{\text{erf}}_{U,V}$), and a de-randomisation procedure relying on margins and concentration rather than a majority vote bound. Both the form of Q used and the de-randomisation step lead to issues which we have addressed through our alternative formulation of Q and a majority vote bound: de-randomisation requires a very low variance Q, leading to the $\sqrt{K}/\gamma$ term in the bound, which is empirically very large for low margin losses. Thus as demonstrated in their experiments, the big-O term increases with widening networks. Finally we note the most important distinction to our work: contrary to the present work, Biggs & Guedj (2022) do not obtain non-vacuous bounds in practice.

4.2. GELU Networks

The Gaussian Error Linear Unit is a commonly-used alternative to the ReLU activation defined by $\text{GELU}(t) := \Phi(t)\, t$ where $\Phi(t)$ is the standard normal CDF. Far from the origin, the $\Phi(t)$ is saturated at zero or one so it looks much like a smoothed ReLU or SWISH activation (defined by Ramachandran et al., 2018 as $x/(1 + e^{-cx})$ for some $c > 0$). It was introduced to lend a more probabilistic interpretation to activation functions, and fold in ideas of regularisation by effectively averaging the output of adaptive dropout (Ba & Frey, 2013); its wide use reflects excellent empirical results in a wide variety of settings.

Definition 4.2. *GELU Network.* For $U \in \mathbb{R}^{K\times d}$, $V \in \mathbb{R}^{K\times D}$, and $\beta > 0$, a β-normalised single hidden layer GELU network is defined by

$$F^{\text{GELU}}_{U,V}(x) := V \cdot \text{GELU}\left(\beta \frac{Ux}{\|x\|_2}\right)$$

where $\text{GELU}(t) := \Phi(t)\, t$.

Theorem 4.3. *In the multi-class setting, fix prior parameters $U^n \in \mathbb{R}^{K\times d}$ and $V^n \in \mathbb{R}^{D\times K}$, $\sigma_V > 0$, $\sigma_U > 0$ $\beta > 0$, and data distribution $\mathcal{D}$. For $\delta \in (0,1)$, with probability at least $1-\delta$ under the sample $S \sim \mathcal{D}^m$, simultaneously for any $U \in \mathbb{R}^{K\times d}, V \in \mathbb{R}^{D\times K}$ such that assumption ($\star$) is satisfied,*

$$L(F^{\text{GELU}}_{U,V}) \leq 2\ \text{kl}^{-1}\left(\hat{L}(Q),\ \frac{\kappa + \log\frac{2\sqrt{m}}{\delta}}{m}\right). \qquad (3)$$

Here $F^{\text{GELU}}_{U,V}$ is a single-hidden-layer GELU network with β-normalised activation,

$$\kappa := \left(\beta^2 + \frac{1}{\sigma_U^2}\right)\frac{\|U - U^0\|_F^2}{2} + \frac{\|V - V^0\|_F^2}{2\sigma_V^2},$$

and $\hat{L}(Q)$ is

$$\frac{1}{m}\sum_{(x,y)\in S} \mathbb{P}\left\{\text{argmax}\left[W_2(\mathbf{1}_{W_1 x} \otimes (W_1' x))\right] \neq y\right\},$$

with the probability is over draws of $\text{vec}(W_2) \sim N(\text{vec}(V), \sigma_V^2 I), \text{vec}(W_1), \sim N(\text{vec}(U), \beta^{-2} I)$ *and* $\text{vec}(W_1') \sim N(\text{vec}(V), \sigma_U^2 I)$. *Here* vec *is the vectorisation operator and the indicator function $\mathbf{1}_y$ is applied element-wise.*

Although the proof method for Theorem 4.3 and the considerations around the hyper-parameter β are the same as for Theorem 4.1 and SHEL networks, one notable difference is the inclusion of the σ_U parameter. When this is very small, the stochastic predictions are effectively just a linear two-layer network with adaptive dropout providing the non-linearity. The ability to adjust the variability of the stochastic network hidden layer and thus $\hat{L}(Q)$ is a major advantage over the SHEL network; in SHEL networks this variability can only be changed through β, which is a fixed parameter related to the deterministic network, not just a quantity appearing only in the bound.

4.3. General Form

Both of the above bounds can effectively be derived from the same formulation, as both take the form

$$F(x) := \mathbb{E}_{f\sim Q} f(x) = \sum_{k=1}^{K} v_k H_k(x), \tag{4}$$

where $v_k \in \mathbb{R}^D$ are the column vectors of a matrix $V \in \mathbb{R}^{D\times K}$ and $H_k : \mathcal{X} \to \mathbb{R}$ is itself a predictor of a form expressible as the expectation of another predictor. This means that there exists a distribution on functions $Q^k \in \mathcal{M}_1^+(\mathcal{F}^k)$ such that for each $x \in \mathcal{X}$, $H_k(x) = \mathbb{E}_{h\sim Q^k}[h(x)]$. The bound on the generalisation of such predictors takes essentially the same form those given in the rest of this section.

Theorem 4.4. *Fix a set of priors $P^k \in \mathcal{M}_1^+(\mathcal{F}^k)$ for $k \in [K]$, a prior weight matrix $V^0 \in \mathbb{R}^{D\times K}$, $\sigma_V > 0$, $\delta \in (0,1)$. With probability at least $1-\delta$ under the sample $S \sim \mathcal{D}^m$ simultaneously for any $V \in \mathbb{R}^{D\times K}$ and set of $Q^k \in \mathcal{M}_1^+(\mathcal{F}^k)$ such that assumption ($\star$) holds,*

$$L(F) \le 2\ \mathrm{kl}^{-1}\left(\hat{L}(Q),\ \frac{\kappa + \log\frac{2\sqrt{m}}{\delta}}{m}\right) \tag{5}$$

where F is the deterministic predictor given in Equation (4),

$$\kappa := \sum_{k=1}^{K} \mathrm{KL}(Q^k, P^k) + \frac{\|V - V^0\|_F^2}{2\sigma_V^2},$$

and

$$\hat{L}(Q) := \frac{1}{m}\sum_{(x,y)\in S} \mathbb{P}\left\{\mathrm{argmax}\left[\sum_{k=1}^{K} w^k h^k(x)\right] \neq y\right\}$$

is the stochastic predictor sample error where the probability is over independent draws of $w^k \sim N(v_k, \sigma_V^2 I), h^k \sim Q^k$ for all $k \in [K]$.

5. Numerical Experiments

For numerical evaluation and the tightest possible values of bounds, a few further ingredients are needed, which are here described. We also give the specific way these are evaluated in our later experiments.

Bounding the empirical error term. We note that there is rarely a closed form expression for $\hat{L}(Q)$, as there is in the binary SHEL bound. In the multi-class bounds, this term must be estimated and bounded by making many independent draws of the parameters and using the fact that the quantity is bounded in $[0,1]$ to provide a concentration bound through, for example, Hoeffding's inequality. This adds a penalty to the bound which reduces with the number of independent draws and thus the amount of computing time invested in calculating the bound, but this is not a theoretical drawback of the bound. We give here a form which is useful in the neural network setting, where it is computationally efficient to re-draw predictors for every prediction, but we make T passes through the dataset to ensure a tight bound. This formulation is considerably more computationally efficient than drawing a single h for every pass of the dataset.

Theorem 5.1 (Train Set Bound)**.** *Let Q be some distribution over predictors and $h^{i,t} \sim Q$ be i.i.d. draws for $i \in [m], t \in [T]$. Then with probability at least $1-\delta'$,*

$$\hat{L}(Q) \le \frac{1}{mT}\sum_{i=1}^{m}\sum_{t=1}^{T} \ell(h^{i,t}(x_i), y_i) + \sqrt{\frac{\log\frac{1}{\delta'}}{2mT}}.$$

In our results, we will set $\delta' = 0.01$ (zero in the binary SHEL case), $T = 20$, and the generalisation bound $\delta = 0.025$; combining them our overall results will hold with probability at least $\delta + \delta' = 0.035$, as in Dziugaite & Roy (2017).

Variance Parameters β and σ. The parameters β, σ_V and σ_U control the variances of the weights in the stochastic estimator defined by Q, but fulfil different functions. The β parameter appears in the non-stochastic shallow network $F_{U,V}$ and thus affects the final predictions made and the training by SGD, and can be related to data normalisation as discussed above. We therefore set it to the fixed value of $\beta = 5$ in all our experiments.

However the σ parameters appear only on the right hand side of the bounds for multi-class SHEL and GELU, and can be tuned to provide the tightest bounds—as they grow the KL term reduces but the performance of Q will degrade. We therefore optimise the final bounds over a grid of σ values as follows: choose a prior grid of σ_V values, $\sigma_V \in \{\sigma_V^1, \ldots, \sigma_V^r\}$, and combine via a union bound argument to add a $\log(r)$ term to κ where r is the number of grid elements. The same practice is applied to σ_U in the GELU case. In practice we use a grid $\sigma \in \{0.05, 0.06, \ldots, 0.2\}$ for both. Thus the tuning of σ_U and σ_V is not a feature of the bound like β, but rather a tool to optimise the tightness of the bounds.

The parameter T appearing in Theorem 3.2 fufils a similar function, trading off the performance of $\hat{L}(Q^{\otimes T})$ versus the complexity term, but we do not optimise it like the above in our experiments, fixing it to $T = 500$ in all our results.

Coupling Procedure. We adopt a 60%-prefix coupling procedure for generating the prior weights U^n, V^n (rather than U^0, V^0, and similarly in the binary case) as in Dziugaite et al. (2021). This works by taking the first 60% of training examples used in our original SGD run and looping them in the same order for up to 4000 epochs. Note that this also

replaces m by $m - n$ and S by S^{bnd} in the bounds, so we are making a trade off between optimising the prior and the tightness of the bound (affected by $m - n$). These are used to train a prior model of the same architecture with the same learning rate from the same initialisation (this is valid because the initialisation is data-independent). The best bound from the generated prior weights was chosen (with a small penalty for this choice added to the bound via a union argument).

Numerical Results. In order to evaluate the quality of the bounds provided, we made many evaluations of the bound under many different training scenarios. In particular we show that the bound behaves in similar ways to the test error on changes of the width, learning rate, training set size and random relabelling of the data.

The following results follow by training β-normalised SHEL and GELU networks with stochastic gradient descent on the cross-entropy loss to a fixed cross entropy value of 0.3 for Fashion-MNIST and 0.1 for MNIST. When evaluating the binary SHEL bound (Theorem 3.2) we use binarised versions of the datasets where the two classes consist of the combined classes $\{0, \ldots, 4\}$ and $\{5, \ldots, 9\}$ respectively (following Dziugaite & Roy, 2017; Letarte et al., 2019), training to cross-entropy values of 0.2 for Bin-F (binarised Fashion-MNIST) and 0.1 for Bin-M (binarised MNIST) respectively. We trained using SGD with momentum $= 0.9$ (as suggested by Hendrycks & Gimpel, 2016 and following Biggs & Guedj, 2022) and a batch size of 200, or without momentum and a batch size of 1000 (with this larger batch size stabilising training). We evaluated for ten different random seeds, a grid search of learning rates $\in \{0.1, 0.03, 0.01\}$ without momentum, and additionally $\in \{0.003, 0.001\}$ with momentum (where small learning rate convergence was considerably faster), and widths $\in \{50, 100, 200, 400, 800, 1600\}$ to generate the bounds in Table 1.

From these results we also show plots in Figure 1 of the test error, stochastic error $\hat{L}_{\text{bnd}}(Q)$ and best prior bound versus width for the different dataset/activation combinations, with more plots given in the appendix. We also note here that in all except the width $= 50$ case, our neural networks have more parameters than there are train data points (60000). Using the test set, we also verified that assumption ($\star$) holds in all cases in which it is used to provide bounds.

6. Discussion

In Table 1 we have given the first non-vacuous bounds for two types of deterministic neural networks trained on MNIST and Fashion-MNIST through a standard SGD learning algorithm, both with and without momentum. The coupled bounds are in all cases far from vacuous, with even the full bounds being non-vacuous in most cases, particularly on the easier MNIST task. Further, Figures 1 and 2 show that the bounds are robustly non-vacuous across a range of widths and learning rates. Since these are direct bounds on $L(F_{U,V})$ rather than the usual PAC-Bayes $L(Q)$, we emphasise that (for fixed hyper-parameters) no trade off is made between the tightness of the bound and the real test set performance, which is usually worse for a higher-variance (and thus more tightly bounded) Q.

Best Coupled Bounds with Momentum

	Data	Test Err	Full Bnd	Coupled Bnd
SHEL	Bin-M	0.038	0.837	0.286
SHEL	Bin-F	0.085	0.426	0.297
SHEL	MNIST	0.046	0.772	0.490
SHEL	Fashion	0.150	0.984	0.727
GELU	MNIST	0.043	0.693	0.293
GELU	Fashion	0.153	0.976	0.568

Best Coupled Bounds without Momentum

	Data	Test Err	Full Bnd	Coupled Bnd
SHEL	Bin-M	0.037	0.835	0.286
SHEL	Bin-F	0.085	0.425	0.300
SHEL	MNIST	0.038	0.821	0.522
SHEL	Fashion	0.136	1.109	0.844
GELU	MNIST	0.036	0.742	0.317
GELU	Fashion	0.135	1.100	0.709

Table 1. Results for β-normalised (with $\beta = 5$) SHEL and GELU networks trained with and without momentum SGD on MNIST, Fashion-MNIST and binarised versions of the above, after a grid search of learning rates and widths as described above. Results shown are those obtaining the tightest coupled bound (calculated using Theorem 4.1 and Theorem 4.3 for the multi-class datasets, and Theorem 3.2 for the binary datasets), with the accompanying full train set bound and test error for the same hyper-parameter settings.

Stability and Robustness Trade-Off. The two main contributions to the bound are the empirical error $\hat{L}(Q)$ and the KL divergence incorporated in κ. $\hat{L}(Q)$ can be seen roughly as measuring a combination of the difficulty of the task for our predictor $F_{U,V}$ combined with some kind of perturbation resistance of its weights (like the idea of a flat minimum originated in Hinton & van Camp, 1993 and discussed at length by Dziugaite & Roy, 2017); while κ is here an empirical measure of the stability of the training method, scaled by the inverse width of the perturbation robustness.

When optimising the trade-off between these terms through a choice of σ_U, σ_V values, we find that the complexity contribution to the bound remains relatively consistent across datasets and architectures, while it is the stochastic error that

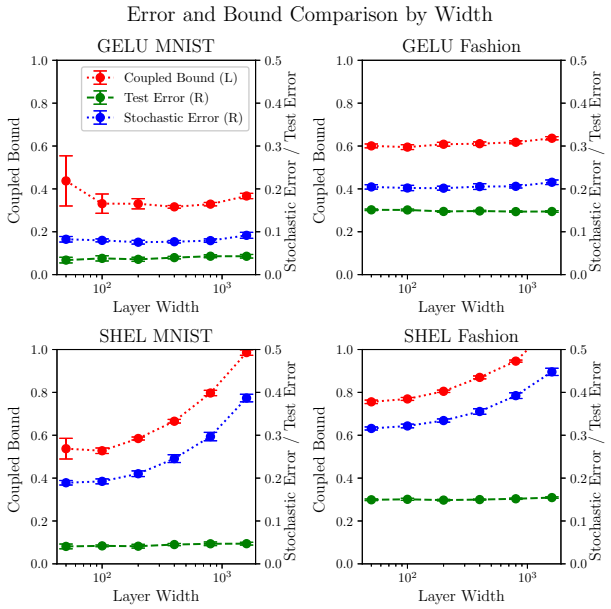

Figure 1. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{\mathrm{bnd}}(Q)$ on the **right** (R) axis versus width for SHEL and GELU networks trained with momentum SGD and learning rate 0.01 on Fashion-MNIST and MNIST. Error bars show 1 standard deviation from ten different random seeds. The different scales are chosen so the trade-off between $\hat{L}_{\mathrm{bnd}}(Q)$ and complexity terms can be seen more easily by neglecting the overall factor of 2, and the trends can be seen more clearly. We include an option in our code to generate these figures with a common scaling instead.

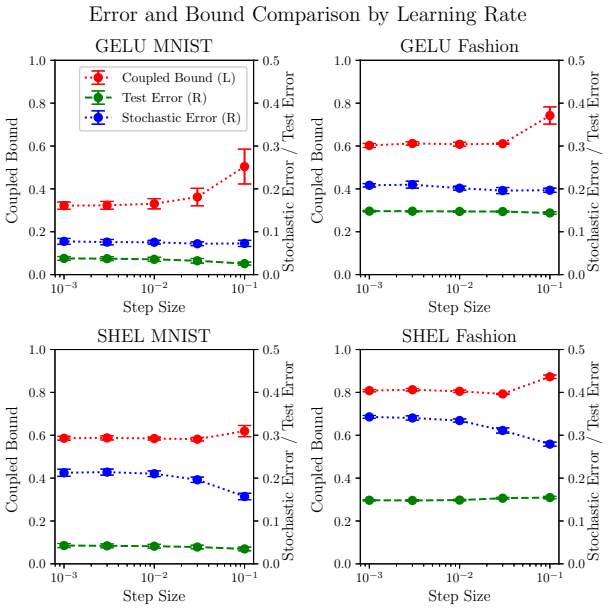

Figure 2. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{\mathrm{bnd}}(Q)$ on the **right** (R) axis versus learning rate for width 200 SHEL and GELU networks trained with momentum SGD on Fashion-MNIST and MNIST. Scales are as in Figure 1.

varies. This is especially true of multi-class SHEL networks as seen in Figure 1, perhaps since there is no easy way to set the stochastic error small by adjusting the variability of the Q hidden layer. This is in direct contrast to many works (Jiang et al., 2020b; Dziugaite et al., 2020) evaluating the predictive ability of PAC-Bayesian bounds for generalisation on hyper-parameter changes, which fix the weight variances as the largest leading to a bound on $\hat{L}(Q)$ of a fixed value, say 0.1. Our results show that this approach may be sub-optimal for predicting generalisation, if as in our results the optimal trade-off tends to fix the κ term and trade off the size of $\hat{L}(Q)$ instead of the reverse[1].

Width Comparison. For the width comparisons we note that it is difficult to discern the real trend in the out-of-sample error of our trained networks. The test sets only have 10000 examples and thus any test-set estimate of $L(F_{U,V})$ is subject to error; if the differences between test errors of two networks of different widths is smaller than about 0.02 (obtained through a Hoeffding bound) it is not possible to say if generalisation is better or worse. It is therefore possible that the pattern of weaker bounds for wider SHEL networks seen is a strong amplification of an existing trend, but it seems more likely it is an artefact of the bound shared with that of Biggs & Guedj (2022). Assuming the latter conclusion that the trained network true error really is relatively width-independent, the GELU bound does better matching this prediction (with this also being true in the momentum-free case, see appendix). The value of $\hat{L}_{\mathrm{bnd}}(Q)$ stays roughly constant as width increases, while we observe that the optimal bound σ_U tends to decrease with increasing width. We attribute to this the tighter bounds for wide GELU networks, since the SHEL network has no comparable way to reduce the randomness of the hidden layer in Q, as we discuss at the end of Section 4.2.

Lower-Variance Stochastic Predictions. Following from the above, we note that in general $\hat{L}_{\mathrm{bnd}}(Q)$ is smaller for comparably-trained GELU networks than the SHEL networks. We speculate that this arises from the increased randomness of the hidden layer of Q in Theorem 4.1: the sign activation is only $\{+1, -1\}$-valued and the amount of information coming through this layer is therefore more limited; and a $\{+1, -1\}$-valued random variable has maximum variance among $[+1, -1]$-bounded variables of given mean. In future work we will explore whether variance reduction techniques such as averaging multiple samples for each activation can improve the tightness of the bounds, but we also emphasise both that the bounds are still non-vacuous across a range of widths, and that the ability to adjust this variabil-

[1]The use of bi-criterion plots as suggested by Neyshabur et al. (2017) may therefore offer an better alternative when comparing vacuous bounds.

ity is a central advantage of our new GELU formulation.

Learning Rate Comparison and Stability. In the case of training with momentum SGD we see that a very large learning rate leads to weaker and higher-variance bounds, with significantly larger norm contribution in κ. We speculate this arises because of the reduced stability at such high rates: we found in general that small batch sizes (particularly under vanilla SGD) and fast learning rates caused the training trajectory of U^n, V^n to diverge more greatly from that of U, V.

Improving Prior Coupling. With the instability of high learning rates and the empirical observation that in many cases $\hat{L}(Q)$ was very close to $L(Q)$ (as estimated from the test set), we see that there is a degree of slackness in the bound arising from the κ term. We speculate that it may be possible to make more efficient use of the sample S in constructing U^n, V^n to reduce this term further. This might be possible through an improved coupling scheme, or through extra side-channel information from S^{bnd} which can be compressed (as per Zhou et al., 2019) or is utilised in a differentially-private manner (as by Dziugaite & Roy, 2018).

Majority Votes. In our results we rely on the novel idea of randomised single-hidden-layer neural networks as the expectation or majority vote of randomised predictors for de-randomisation of our PAC-Bayes bound. For the multi-class bounds we rely on an additional assumption, so a first step in future work could be providing further conditions under which this assumption can be justified without relying on a test set. Next, we found empirically (similarly to many PAC-Bayesian works) that $L(Q) > L(F_{U,V})$, in other words the derandomised predictor was better than the stochastic version on the test set. By de-randomising through the first order bound, we introduce a factor of 2 which cannot be tight in such cases. Removal of this term would lead to considerably tighter bounds and even non-vacuous bounds for CIFAR-10 (Krizhevsky, 2009), based on preliminary experiments, where the training error for one-hidden-layer networks on CIFAR-10 was greater than 0.5 so such bounds could not be non-vacuous, but the final bounds were only around $1.1-1.2$. Improved bounds for the majority vote have been the focus of a wide variety of PAC-Bayesian works (Lacasse et al., 2006; Masegosa et al., 2020), and can theoretically give tighter results for $L(\mathrm{MV}_Q)$ than $L(Q)$, but these are not yet competitive. They universally led to inferior or vacuous results in preliminary experiments. However, there is still much scope for exploration here: alternative formulations of the oracle C-bound lead to different empirical bounds, and improvement of the KL term (which appears more times in an empirical C-bound than Theorem 2.1) may improve these bounds more than the first order one. We also hope that offering this new perspective on one-hidden-layer networks as majority votes can lead to better understanding of their properties, and perhaps even of closely-related Gaussian processes (Neal, 1996).

Deeper networks and convolutions. An extremely interesting question whether this approach will generalise to convolutions or deeper networks. For convolutions, the parameter sharing is not a problem as separate samples can be taken for each convolution kernel position (although potentially at a large KL divergence cost that might be mitigated through the use of symmetry). For deeper networks the answer is less clear, but the empirically-observed stability of most trained networks to weight perturbation would suggest that the mode of a Bayesian neural network may at least be a close approximation to its majority vote, a connection that could lead to further results.

Summary. We have provided non-vacuous generalisation bounds for shallow neural networks through novel methods that make a promising new link to majority votes. Although some aspects of our approach have recently appeared in the PAC-Bayesian literature on neural networks, we note that all previous results obtaining non-vacuous generalisation bounds only apply to randomised versions of neural networks. This often leads to degraded test set performance versus a deterministic predictor. By providing bounds directly on the deterministic networks we provide a setting through which the impact of robustness, flat-minima and stability on generalisation can be explored directly, without making potentially sub-optimal trade-offs or invoking stringent assumptions.

In future work we intend to address two main potential sources of improvement: through progress in majority votes to tighten the step from stochastic to deterministic predictor; and through development of the prior (perhaps thorough improved utilisation of data), a strand running parallel to much PAC-Bayesian research on neural networks.

Acknowledgements

F.B. acknowledges the support of the EPSRC grant EP/S021566/1. B.G. acknowledges partial support by the U.S. Army Research Laboratory, U.S. Army Research Office, U.K. Ministry of Defence and the U.K. Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/R013616/1; B.G. also acknowledges partial support from the French National Agency for Research, grants ANR-18-CE40-0016-01 and ANR-18-CE23-0015-02.

References

Alquier, P. User-friendly introduction to PAC-Bayes bounds. 2021. URL https://www.arxiv.org/abs/2110.11216.

Ambroladze, A., Parrado-Hernández, E., and Shawe-Taylor, J. Tighter pac-bayes bounds. In Schölkopf, B., Platt, J. C., and Hofmann, T. (eds.), *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pp. 9–16. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 254–263. PMLR, 2018. URL http://proceedings.mlr.press/v80/arora18b.html.

Ba, L. J. and Frey, B. J. Adaptive dropout for training deep neural networks. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 3084–3092, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/7b5b23f4aadf9513306bcd59afb6e4c9-Abstract.html.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6240–6249, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/b22b257ad0519d4500539da3c8bcf4dd-Abstract.html.

Biggs, F. and Guedj, B. Differentiable PAC-Bayes objectives with partially aggregated neural networks. *Entropy*, 23(10):1280, 2021. doi: 10.3390/e23101280. URL https://doi.org/10.3390/e23101280.

Biggs, F. and Guedj, B. On margins and derandomisation in pac-bayes. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I. (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pp. 3709–3731. PMLR, 2022. URL https://proceedings.mlr.press/v151/biggs22a.html.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006. ISBN 0471241954.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Conference on Uncertainty in Artificial Intelligence 33.*, 2017.

Dziugaite, G. K. and Roy, D. M. Data-dependent PAC-Bayes priors via differential privacy. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8440–8450, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/9a0ee0a9e7a42d2d69b8f86b3a0756b1-Abstract.html.

Dziugaite, G. K., Drouin, A., Neal, B., Rajkumar, N., Caballero, E., Wang, L., Mitliagkas, I., and Roy, D. M. In search of robust measures of generalization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/86d7c8a08b4aaa1bc7c599473f5dddda-Abstract.html.

Dziugaite, G. K., Hsu, K., Gharbieh, W., Arpino, G., and Roy, D. On the role of data in pac-bayes. In Banerjee, A. and Fukumizu, K. (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 604–612. PMLR, 2021. URL http://proceedings.mlr.press/v130/karolina-dziugaite21a.html.

Germain, P., Lacasse, A., Laviolette, F., and Marchand, M. PAC-Bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pp. 1–8, Montreal, Quebec,

Canada, 2009. ACM Press. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553419.

Guedj, B. A Primer on PAC-Bayesian Learning. In *Proceedings of the second congress of the French Mathematical Society*, 2019. URL https://arxiv.org/abs/1901.05353.

Hellström, F. and Durisi, G. Fast-rate loss bounds via conditional information measures with applications to neural networks. In *IEEE International Symposium on Information Theory, ISIT 2021, Melbourne, Australia, July 12-20, 2021*, pp. 952–957. IEEE, 2021. doi: 10.1109/ISIT45174.2021.9517731. URL https://doi.org/10.1109/ISIT45174.2021.9517731.

Hendrycks, D. and Gimpel, K. Bridging nonlinearities and stochastic regularizers with Gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL https://arxiv.org/abs/1606.08415.

Hinton, G. E. and van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In Pitt, L. (ed.), *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory, COLT 1993, Santa Cruz, CA, USA, July 26-28, 1993*, pp. 5–13. ACM, 1993. doi: 10.1145/168304.168306. URL https://doi.org/10.1145/168304.168306.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Comput.*, 9(1):1–42, 1997. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

Jiang, Y., Foret, P., Yak, S., Roy, D. M., Mobahi, H., Dziugaite, G. K., Bengio, S., Gunasekar, S., Guyon, I., and Neyshabur, B. NeurIPS 2020 competition: Predicting generalization in deep learning. *CoRR*, abs/2012.07976, 2020a. URL https://arxiv.org/abs/2012.07976.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b. URL https://openreview.net/forum?id=SJgIPJBFvH.

Kawaguchi, K., Kaelbling, L. P., and Bengio, Y. Generalization in deep learning. To appear in Mathematics of Deep Learning, Cambridge University Press, 2020. URL https://www.arxiv.org/abs/1710.05468.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Lacasse, A., Laviolette, F., Marchand, M., Germain, P., and Usunier, N. PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In Schölkopf, B., Platt, J. C., and Hofmann, T. (eds.), *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pp. 769–776. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/779efbd24d5a7e37ce8dc93e7c04d572-Abstract.html.

Langford, J. and Seeger, M. Bounds for averaging classifiers. 2001. URL http://www.cs.cmu.edu/~jcl/papers/averaging/averaging_tech.pdf.

Langford, J. and Shawe-Taylor, J. PAC-Bayes & margins. In *Advances in Neural Information Processing Systems*, pp. 439–446, 2003.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Letarte, G., Germain, P., Guedj, B., and Laviolette, F. Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 6872–6882. Curran Associates, Inc., 2019.

Masegosa, A. R., Lorenzen, S. S., Igel, C., and Seldin, Y. Second order PAC-Bayesian bounds for the weighted majority vote. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/386854131f58a556343e056f03626e00-Abstract.html.

Maurer, A. A note on the PAC-Bayesian theorem. *CoRR*, cs.LG/0411099, 2004. URL https://arxiv.org/abs/cs.LG/0411099.

McAllester, D. A. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational Learning Theory*, pp. 230–234. ACM, 1998.

McAllester, D. A. PAC-Bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational Learning Theory*, pp. 164–170. ACM, 1999.

Nagarajan, V. and Kolter, J. Z. Deterministic PAC-Bayesian generalization bounds for deep networks via generalizing noise-resilience. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a. URL https://openreview.net/forum?id=Hygn2o0qKX.

Nagarajan, V. and Kolter, J. Z. Uniform convergence may be unable to explain generalization in deep learning. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 11611–11622, 2019b. URL https://proceedings.neurips.cc/paper/2019/hash/05e97c207235d63ceb1db43c60db7bbb-Abstract.html.

Neal, R. M. *Priors for Infinite Networks*, pp. 29–53. Springer New York, New York, NY, 1996. ISBN 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0_2. URL https://doi.org/10.1007/978-1-4612-0745-0_2.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. In Grünwald, P., Hazan, E., and Kale, S. (eds.), *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pp. 1376–1401. JMLR.org, 2015. URL http://proceedings.mlr.press/v40/Neyshabur15.html.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5947–5956, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/10ce03a1ed01077e3e289f3e53c72813-Abstract.html.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Skz_WfbCZ.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. The role of over-parametrization in generalization of neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BygfghAcYX.

Parrado-Hernández, E., Ambroladze, A., Shawe-Taylor, J., and Sun, S. PAC-Bayes bounds with data dependent priors. *J. Mach. Learn. Res.*, 13: 3507–3531, 2012. URL http://dl.acm.org/citation.cfm?id=2503353.

Perez-Ortiz, M., Rivasplata, O., Guedj, B., Gleeson, M., Zhang, J., Shawe-Taylor, J., Bober, M., and Kittler, J. Learning PAC-Bayes priors for probabilistic neural networks. 2021a. URL https://arxiv.org/abs/2109.10304.

Perez-Ortiz, M., Rivasplata, O., Shawe-Taylor, J., and Szepesvari, C. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227):1–40, 2021b. URL http://jmlr.org/papers/v22/20-879.html.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hkuq2EkPf.

Rivasplata, O., Szepesvári, C., Shawe-Taylor, J., Parrado-Hernández, E., and Sun, S. PAC-Bayes bounds for stable algorithms with instance-dependent priors. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 9234–9244, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/386854131f58a556343e056f03626e00-Abstract.html.

Tinsi, L. and Dalalyan, A. S. Risk bounds for aggregated shallow neural networks using gaussian priors. 2021. URL https://arxiv.org/abs/2112.11086.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL https://arxiv.org/abs/1708.07747.

Zantedeschi, V., Viallard, P., Morvant, E., Emonet, R., Habrard, A., Germain, P., and Guedj, B. Learning stochastic majority votes by minimizing a PAC-Bayes generalization bound. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS*, 2021. URL https://arxiv.org/abs/2106.12535.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. Non-vacuous generalization bounds at the ImageNet scale: A PAC-Bayesian compression approach. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJgqqsAct7.

A. Proofs

Proof of Theorem 3.2. We consider randomised functions $f(x) = \frac{1}{T}\sum_{t=1}^{T} \operatorname{sign}(w_t \cdot x)$ with $w_1, \ldots, w_T \sim Q^{\otimes T}$ identically and independently distributed. Here Q is a mixture of Gaussians distribution with $2K$ components; we denote by $Q_k = \operatorname{Categ}(q)$ the distribution over the choice of component, and by Q^k the corresponding component. We choose the mixture component weights

$$q = \frac{1}{\|v\|_1}[\max(0, v_1), \ldots, \max(0, v_K), \max(0, -v_1), \ldots, \max(0, -v_K)],$$

and component distributions $Q^k = N(u_k, \frac{1}{2}\beta^{-2}I)$ for $k \in 1, \ldots, K$, and $Q^k = N(-u_k, \frac{1}{2}\beta^{-2}I)$ for $k \in K+1, \ldots, 2K$. Here u_k are the rows of U. This dimension-doubling trick accommodates the use of negative final-layer weights.

A PAC-Bayes bound on the above relates to the SHEL network through the following. Firstly, it is easy to show that $\mathbb{E}_{f\sim Q^{\otimes T}} f(x) = \frac{1}{\|v\|_1} F(x)$, where F is the SHEL network with parameters U, v as given above. This follows using the expectation of a mixture followed by using the aggregation of a sign function under a Gaussian weight given in Equation (2), which gives

$$\mathbb{E}_{f\sim Q^{\otimes T}} f(x) = \sum_{k=1}^{K} q_k \operatorname{erf}\left(\beta \frac{u_k \cdot x}{\|x\|_2}\right) + \sum_{k=K+1}^{2K} q_k \operatorname{erf}\left(\beta \frac{-u_k \cdot x}{\|x\|_2}\right) = \frac{F(x)}{\|v\|_1}$$

The predictions of this SHEL network, $\operatorname{sign} F(x)$, are equivalent to a majority vote of $f(x)$, since $\operatorname{MV}(x) = \operatorname{sign}(\mathbb{E}\operatorname{sign}(f(x)))$ is 1 if $\mathbb{E}f(x) \propto F(x) \geq 0$ and vice-versa for -1. Therefore the first order bound can be used to see that $\ell(F(x), y) \leq 2\mathbb{E}_{Q^{\otimes T}} \ell(f(x), y)$.

To obtain a PAC-Bayes bound in full, we choose a set of prior weights U^0, v^0 to define a prior P that takes the same structure as Q. The index distribution $P^k = \operatorname{Categ}(p)$ with

$$p = \frac{1}{2\|v^0\|}[|v_1^0|, \ldots, |v_K^0|, |v_1^0|, \ldots, |v_K^0|],$$

and component distributions defined as per Q^k but with weights u_k^0 instead.

Then, using the chain rule for KL divergence (Cover & Thomas, 2006) twice,

$$\operatorname{KL}(Q, P) \leq \operatorname{KL}(Q_{w,k}, P_{w,k}) \leq \operatorname{KL}(Q_{w|k}, P_{w|k}) + \operatorname{KL}(Q_k, P_k) \tag{6}$$

where $Q_{w,k}$ and $Q_{w|k}$ are the joint and conditional distributions on w and mixture index k (and analogously for P), as opposed to Q, which is a marginal on w.

Using the definitions of the KL divergence for categorical and Gaussian distributions in the above, $\operatorname{KL}(Q, P)$ is bounded by

$$\sum_{k=1}^{K} q_k \beta \|u_k - u_k^0\|_2^2 + \sum_{k=1}^{K} q_k \log \frac{q_k}{p_k} = \kappa.$$

Combining Theorem 2.1 with the fact that $\operatorname{KL}(Q^{\otimes T}, P^{\otimes T}) = T \operatorname{KL}(Q, P)$ since the T copies are i.i.d., the following holds with probability $\geq 1 - \delta$

$$L(F_{U,v}) \leq 2 \operatorname{kl}^{-1}\left(\hat{L}(Q^{\otimes T}),\ \frac{T\kappa + \log \frac{2\sqrt{m}}{\delta}}{m}\right).$$

To complete the result we also note the closed form for $\hat{L}(Q^{\otimes T})$ given through the following. The average misclassification

loss

$$\begin{aligned}
\mathbb{E}_{Q^{\otimes T}} \ell(f(x), y) &= \mathbb{P}_{Q^{\otimes T}}\left(yf(x) \le 0\right) \\
&= \mathbb{P}_{Q^{\otimes T}}\left(\sum_{t=1}^{T} y \operatorname{sign}(w^t \cdot x) \le 0\right) \\
&= \mathbb{P}_{Q^{\otimes T}}\left(\sum_{t=1}^{T} \frac{1}{2}(y \operatorname{sign}(w^t \cdot x) + 1) \le \frac{1}{2}T\right) \\
&= \mathbb{P}_{Q^{\otimes T}}\left(\sum_{t=1}^{T} \mathbf{1}_{y=\operatorname{sign}(w^t \cdot x)} \le \frac{1}{2}T\right) \\
&= \operatorname{Bin}\left(\frac{T}{2}; T, \mathbb{P}_Q(y = \operatorname{sign}(w^t \cdot x))\right) \\
&= \operatorname{Bin}\left(\frac{T}{2}; T, \frac{1}{2}\left(1 + \frac{yF(x)}{\|v\|_1}\right)\right)
\end{aligned}$$

where we have interchanged $\mathbf{1}_{y=\operatorname{sign}(w \cdot x)} = \frac{1}{2}(y \operatorname{sign}(w \cdot x) + 1)$.

All of the above can be readily extended to the data-dependent prior case, replacing $U^0 \to U^n$, $v^0 \to v^n$, $m \to m - n$, and $\hat{L} \to \hat{L}_{\text{bnd}}$. □

Proof of Theorem 4.4. We are considering a distribution on functions of the form $\sum_k w^k h^k(x)$ where for each index $k \in [K]$ we have $w_k \sim N(\frac{1}{\sigma_V} v_k, I)$ and $h_k \sim Q_k$. This slightly different formulation can take advantage of the scaling-invariance of the final layer to the misclassification loss when $V^0 = 0$, so we can then choose $\sigma_V > 0$ arbitrarily. The expectation of this takes the form given in Equation (4) scaled by $1/\sigma_V$ and leads to the empirical loss above.

Given another distribution P taking a similar form with $w_k \sim N(\frac{1}{\sigma_V} v_k^0, I)$ and components P_k, the KL divergence can be expressed (using the chain rule for KL divergence) as

$$\operatorname{KL}(Q, P) \le \sum_{k=1}^{K} \operatorname{KL}(Q^k, P^k) + \frac{\|V - V^0\|_F^2}{2\sigma_V^2}.$$

We prove the overall bound by combining Theorem 2.1 with the assumption ($\star$). □

Proof of Theorem 4.1. Apply the bound from Theorem 4.4 with the individual units as $h_k(x) = \operatorname{sign}(w_k \cdot x)$ and $w_k \sim N(u_k, \frac{1}{2}\beta^{-2} I)$ alongside Theorem 4.4. The aggregated form of the sign activation function is given in (2). The prior takes the same form as the posterior with weight means U^0, V^0 and the same variances, leading to the form of KL divergence for Gaussian weights given in κ. □

Proof of Theorem 4.3. The proof takes the same form as that of Theorem 4.1. We note that the expectation under the given probability distributions of $\mathbb{E}[W_2(\mathbf{1}_{W_1 x} \otimes (W_1' x))] = \|x\|_2 F_{U,V}^{\text{GELU}}(x)$, but since the misclassification loss is scaling-invariant this gives equivalent results. Choosing appropriate prior forms as in Theorem 4.1 gives the KL divergence which we substitute into Theorem 4.4. □

Proof of Theorem 5.1. Define $\xi = \sum_{i=1}^{m} \sum_{t=1}^{T} \frac{1}{mT} \ell(h^{i,t}(x_i), y_i)$ which has expectation $\mathbb{E}_Q \xi = \hat{L}(Q)$. Since this quantity is a sum of mT independent random variables in $\{0, 1/mT\}$, application of Hoeffding's inequality gives the result. □

B. Additional Results and Code

We provide all of our results and code to reproduce them along with the figures (including with the option of using the same scaling for the bound and errors, as described in Figure 1) in the supplementary material. We also note here that the "erf" function is included in a wide variety of common deep learning libraries.

Here we also provide Figures 3 and 4 similar to Figures 1 and 2 for GELU and SHEL networks trained without momentum and with a batch size of 1000, as described in Section 5. We then also provide further similar plots for networks trained with momentum and a batch size of 200 as in Section 5 with different learning rates and widths, to show the similar behaviour across a variety of regimes.

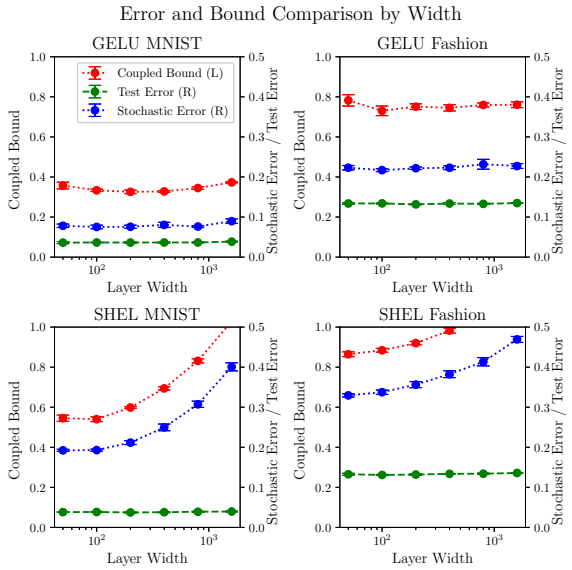

Figure 3. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\mathrm{bnd}}}(Q)$ on the **right** (R) axis versus width for SHEL and GELU networks trained with vanilla SGD and learning rate 0.01 on Fashion-MNIST and MNIST. Scales are as in Figure 1.

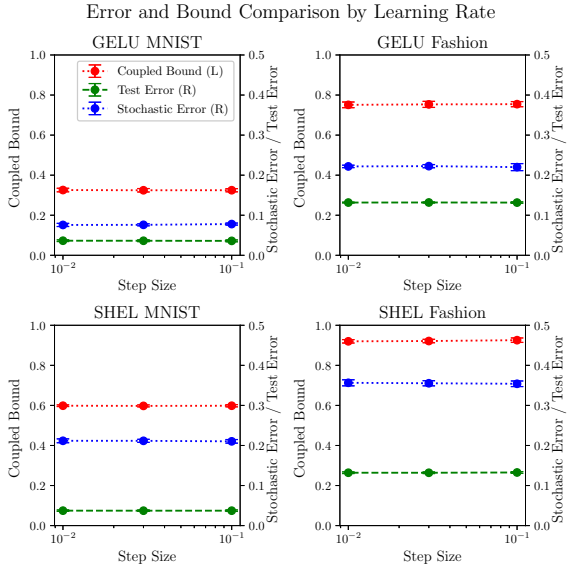

Figure 4. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\mathrm{bnd}}}(Q)$ on the **right** (R) axis versus learning rate for width 200 SHEL and GELU networks trained with vanilla SGD on Fashion-MNIST and MNIST. Scales are as in Figure 1.

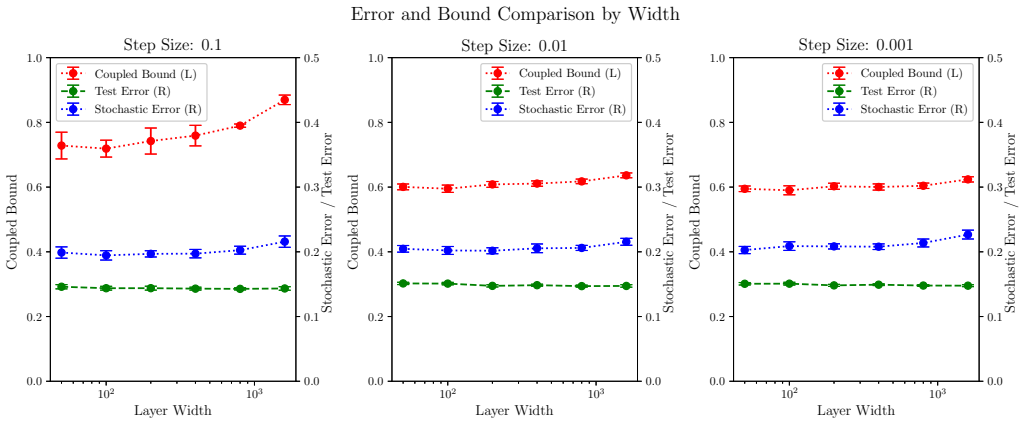

Figure 5. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{\mathcal{S}\mathrm{bnd}}(Q)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a GELU network trained with momentum on Fashion-MNIST.

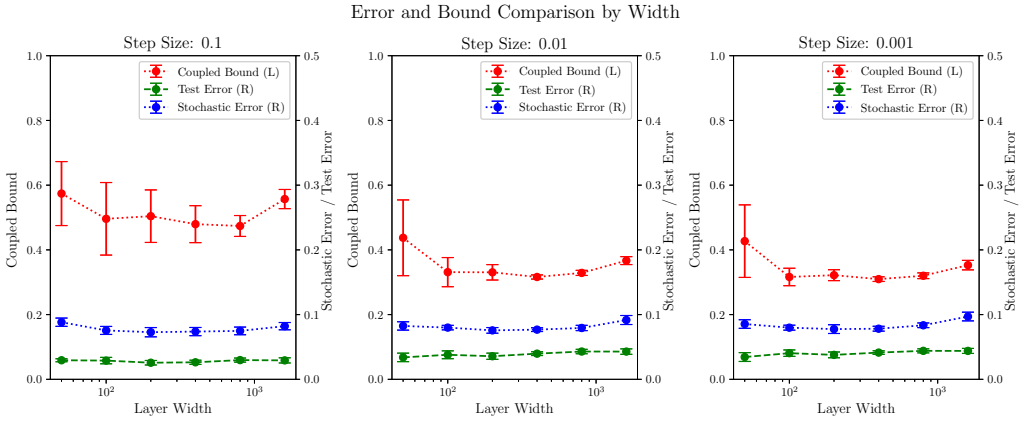

Figure 6. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{\mathcal{S}\mathrm{bnd}}(Q)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a GELU network trained with momentum on MNIST.

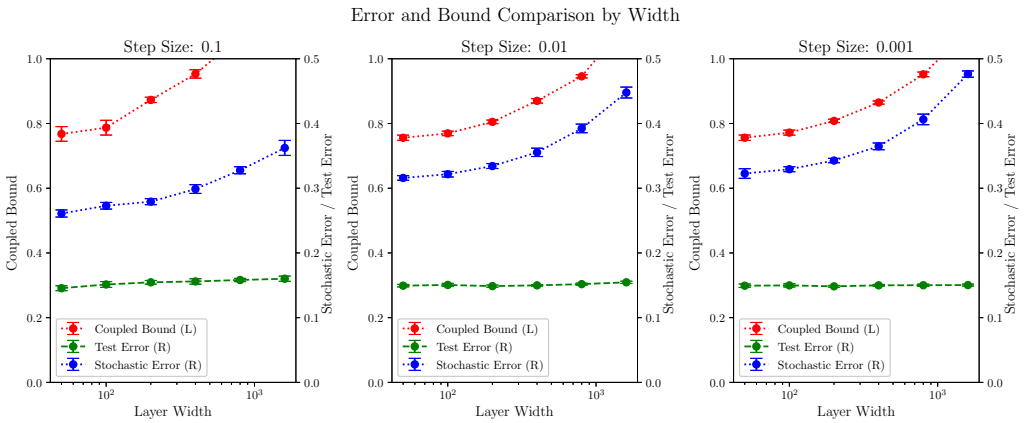

Figure 7. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{\mathcal{S}\mathrm{bnd}}(Q)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a SHEL network trained with momentum on Fashion-MNIST.

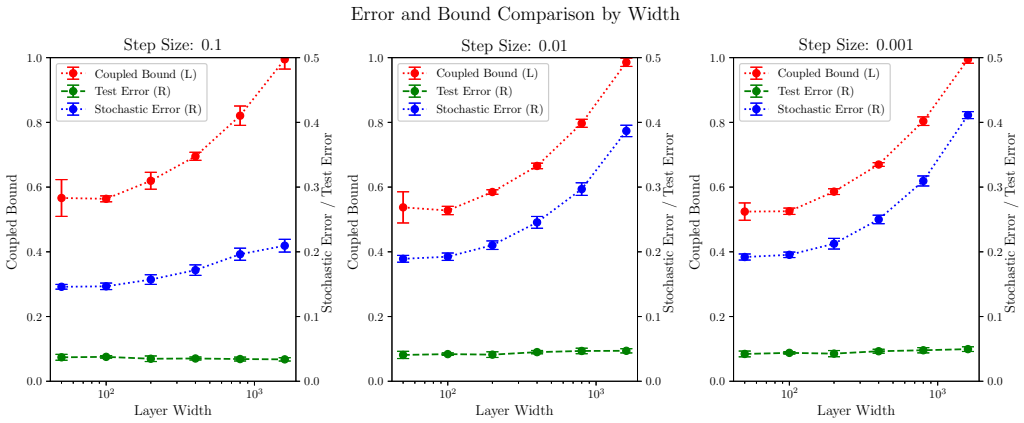

Figure 8. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\mathrm{bnd}}}(Q)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a SHEL network trained with momentum on MNIST.

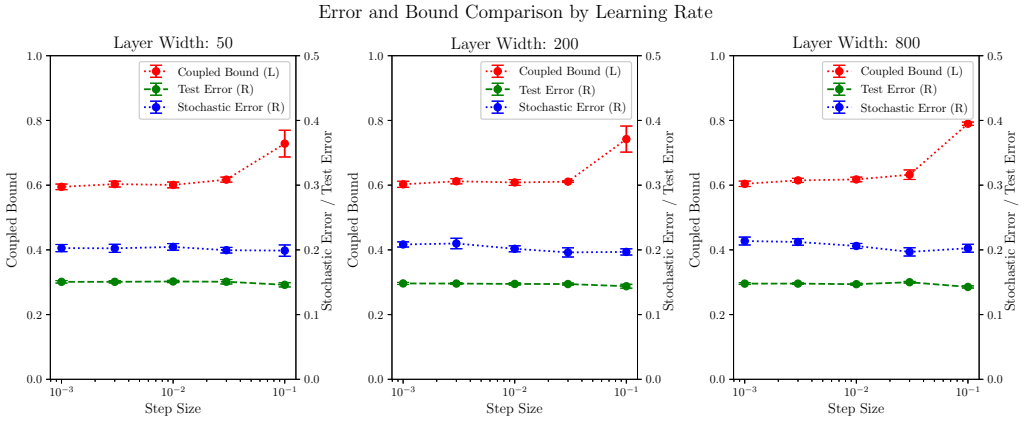

Figure 9. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\mathrm{bnd}}}(Q)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a GELU network trained with momentum on Fashion-MNIST.

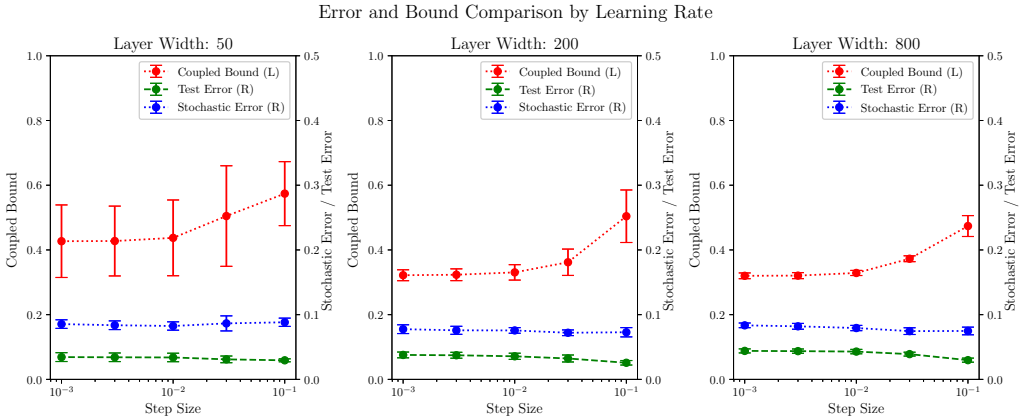

Figure 10. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\mathrm{bnd}}}(Q)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a GELU network trained with momentum on MNIST.

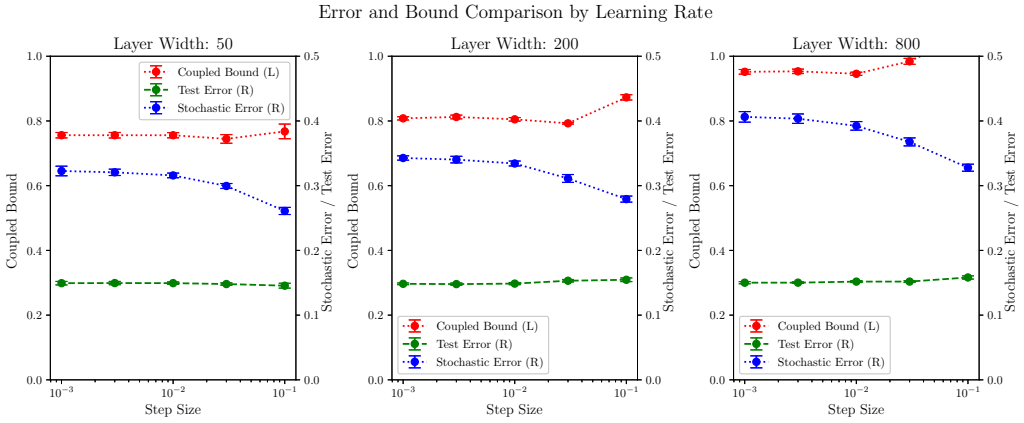

Figure 11. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\mathrm{bnd}}}(Q)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a SHEL network trained with momentum on Fashion-MNIST.

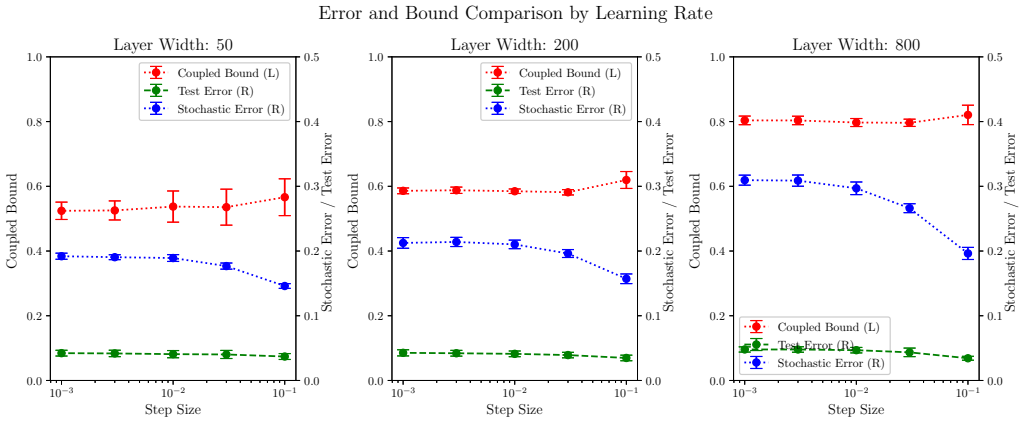

Figure 12. Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\mathrm{bnd}}}(Q)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a SHEL network trained with momentum on MNIST.