On the Algorithmic Stability of SGD in Deep Learning

Abstract

Many existing deep learning generalization bounds do not seem to be informative and can even increase with the sample size, which has further motivated the study of algorithmic stability as a possible approach for overcoming these limitations. In this work, we present empirical evidence that uniform stability might not appear in practical deep learning settings with sufficient strength to explain generalization and, further, that a key requirement of existing arguments is not satisfied: for two datasets differing by a single point, the distance between the final learned parameters does not decrease with dataset size. However, deeper investigation reveals that these failures might not be as bleak as they appear: despite separation by a large distance, these parameters can still sometimes end up in the same basin of attraction. We use our insights to suggest promising directions for algorithmic stability as a tool for explaining generalization in deep learning.

1 Introduction

Despite the impressive empirical success of deep learning models, their ability to generalize well (on a significant set of data distributions) despite overparameterization has thus far largely eluded the research community [25, 23]. Various flavors of generalization bounds have been applied to neural networks, including various norm- and margin-based bounds [2, 22, 7, 17, 19], PAC-Bayes bounds [6, 24], and VC-dimension-based bounds [3]. However, many such bounds have been shown to be insufficiently-correlated with generalization as various model components are varied (e.g., number of parameters) [13]. Recently, Nagarajan and Kolter [20] demonstrated that some of these bounds can even increase with sample size in certain settings, underscoring the importance of *empirically* evaluating proposed bounds' behavior as a function of dataset size. Furthermore, Nagarajan and Kolter [20] suggest that all uniform-convergence-based approaches might inherently be unable to explain deep learning's generalization performance, even *after* uniform convergence is restricted to the smallest possible set of models determined by the implicit bias of the learning algorithm. If this is true, then what tools for proving deep learning generalization bounds remain? One such tool, as acknowledged by Nagarajan and Kolter [20], is *algorithmic stability*.

Algorithmic stability. Algorithmic stability typically refers to a sensitivity analysis of the algorithm itself; specifically, how much can swapping (or removing) one point in an m-item training set S change the output of an algorithm $\mathcal{A}(S)$? Bousquet and Elisseeff [4] formalized and proved generalization bounds under various different flavors of algorithmic stability; since then, additional variants of algorithmic stability have been developed [1, 9, 15, 18]. However, to this day, the main variant for obtaining bounds that hold with high probability over the random draw of the training set is *uniform stability*, the strictest of the requirements. Specifically, a learning algorithm $\mathcal{A}$ is called β-uniformly stable with respect to loss ℓ if:

$$\forall S \in \mathcal{Z}^m, \forall i \in \{1, \dots, m\}, \forall z \in \mathcal{Z} : |\ell(\mathcal{A}(S), z) - \ell(\mathcal{A}(S^{\setminus i}), z)| \leq \beta,$$

where $S^{\setminus i}$ is S with element i removed. Often, uniform stability is expressed with respect to the *swapping* of one point, instead of the *removal*:

$$\forall S, S' \in \mathcal{Z}^m, \forall z \in \mathcal{Z} : |\ell(\mathcal{A}(S), z) - \ell(\mathcal{A}(S'), z)| \leq \beta,$$

where S and S' only differ at one index.

Algorithmic stability of stochastic gradient descent (SGD). Various works thus far have studied whether the framework of algorithmic stability can be applied to the analysis of stochastic gradient descent (typically including at least some extension to nonconvex loss landscapes) [11, 8, 16]. However, each of these results has some *subset* of the following weaknesses when applied to practical deep learning:

- The bound is only in *expectation* with respect to the draw of the sample S. In general, we ultimately seek bounds that will hold with high probability over the draw of the sample $S \sim \mathcal{D}^m$, although such bounds are generally more difficult to prove theoretically.
- The stability parameter β relies on smoothness parameters of the loss landscape that might not be particularly favorable for neural networks.
- The result heavily relies on a learning rate of $\mathcal{O}(1/t)$, where t is the parameter update (vs. epoch). This ensures that, in expectation over the algorithm's randomness, the learning rate has decayed more for larger samples by the time the swapped point is encountered. In contrast, in deep learning, the learning rate typically stays constant for at least the first epoch.
- The proof relies on controlling the (expected) distance between $\mathcal{A}(S)$ and $\mathcal{A}(S')$ in parameter space, which seems unlikely to decrease sufficiently with dataset size in practice (without the aforementioned $1/t$ learning rate schedule). We explore this in more detail in Section 4.

Our work. Inspired by the growing literature empirically analyzing the shortcomings of current deep learning generalization bounds and the anticipated algorithmic stability weaknesses discussed above, in this work we initiate a study of the following question: Does SGD empirically satisfy uniform stability in *practical* deep learning settings, in a manner sufficient to yield generalization bounds that hold with high probability (over the draw of the dataset and the algorithm's randomness)? Unfortunately, analyzing uniform stability *empirically* is incredibly challenging due to the many suprema in the definition (i.e., $\forall S, S', z$), and we thus do not claim that any empirical analysis can *definitively* answer whether or not SGD in deep learning is uniformly stable. However, to our knowledge, this is the most *extensive* empirical examination of uniform stability in deep learning to date. Our contributions are as follows:

- Discussion of challenges in the empirical evaluation of uniform stability, with suggested methodology for overcoming them. Crucially, we validate our methodology in the simpler setting of logistic regression.
- Evidence that uniform stability (with respect to the cross-entropy loss) does not decrease sufficiently with dataset size to fully explain deep learning's generalization.
- Evidence that $\|\mathcal{A}(S) - \mathcal{A}(S')\|_2$ (when the output of $\mathcal{A}$ is treated as a single vector of concatenated parameters) does not sufficiently decrease with dataset size in practical deep learning settings; in some cases, it can even increase despite strong generalization. We suggest that, if there is a form of algorithmic stability at play in deep learning, it does not stem from parameter closeness. We argue that future theoretical attempts to prove stability of SGD in deep learning should proceed through a different key path.
- Discovery of settings with insufficient cross-entropy uniform stability to explain generalization but for which $\mathcal{A}(S)$ and $\mathcal{A}(S')$ are in the same basin of attraction (see Section 4 for a precise definition), suggesting that convex settings with large basins of attraction could also share these same failure modes and thus pave the way for more tractable analyses.

2 Methodology

Here, we describe the key aspects of our methodology for empirically evaluating uniform stability, with additional details in Appendix A. We first applied our methodology to logistic regression, which we used to help validate our methodology. We then applied our methodology to deep neural networks.

Throughout the paper, we use $\mathcal{A}(S)$ to denote to the output of the algorithm on dataset S. Although this object is really a function, we slightly abuse notation and treat it as a vector, i.e., with all of the model's parameters concatenated into a single vector. We occasionally use W_S instead to denote the parameters output by $\mathcal{A}$ on S, concatenated into a single vector.

Random seeds. Since we are seeking bounds that hold with high probability over the randomness of the algorithm, each plot we produce examines a *single* setting of the seed controlling initialization and the seed controlling SGD order. Thus, for each dataset/hyperparameter configuration, we present a single setting of the seeds in the main paper and defer our plots for other seeds to Appendix B.

Datasets: S and S'. We used MNIST for logistic regression (divided into two classes for binary classification: labels 0-4 and labels 5-9), and we used CIFAR-10 [14] and SVHN (Street View House Numbers) [21] for neural network training (10-class classification). In order to thoroughly study behavior (e.g., test/train error, various stability metrics, etc.) as a function of dataset size, we examined the following dataset sizes: $\{800, 1600, 3200, 6400, 12800\}$ for logistic regression and $\{15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000\}$ for neural networks. See Appendix A for more details regarding this choice of dataset sizes.

We emphasize that we intentionally do *not* use data augmentation; we want to precisely measure behavior as a function of dataset size, and to our knowledge, there is no widely-accepted approach for calculating the *effective* dataset size with data augmentation.

Multiple trials per dataset size. To study the quantifier $\forall S, S' \in \mathcal{Z}^m$ *empirically* as thoroughly as possible, we sampled multiple (S, S') pairs per dataset size m. Each (S, S') pair was sampled as follows: we first randomly drew a subset of size m from the relevant training dataset (uniformly at random without replacement) to form S, we then uniformly sampled a single element of the relevant test set (call this element z), and finally we uniformly sampled an index $i \in \{1, \ldots, m\}$ of S in which to swap in z, thus forming S'. We then trained two models in parallel, one on training dataset S and one on S'. This procedure was repeated 90 times per dataset size for logistic regression and 40 times per dataset size for each neural network configuration (due to the higher cost of each run).

Crucially, the only difference between training on S and S' was the appearance of z in S' in a single batch per epoch. All other data points were the same and were visited in the same order. Furthermore, we explicitly disabled all sources of GPU nondeterminism to ensure that we were fully isolating the effect of swapping in z.

Models and training. The logistic regression model is a 784-dimensional linear classifier plus a bias term, and the neural networks are residual networks, specifically ResNet-20 [12].

The logistic regression models were trained via stochastic gradient descent (SGD) with learning rate 0.1, batch size 128, and no momentum.

On SVHN, we trained a ResNet-20 via SGD with learning rate 0.01, batch size 32, and no momentum. On CIFAR-10, we explored two different hyperparameter configurations: one without momentum and one with momentum 0.9. The other hyperparameters were the same across both configurations: a decaying learning rate schedule (starting at 0.1 and dividing by 10 at iterations 32,000 and 48,000) and batch size 128 [12].

Stopping criterion. We train each model for 100,000 iterations (i.e., parameter updates). See Appendix A for a more detailed discussion of stopping criteria.

Uniform stability with respect to the cross-entropy loss. In the uniform stability definition, instead of a supremum over the domain, we calculate a max over the test set. A priori, it might not be clear how effective this would be, and we thus validate our methodology via logistic regression in Section 3 before proceeding to deep learning.

Plots and curve fitting. Many of the quantities examined in this paper take the form of $g(m) = \sup_{S \in \mathcal{Z}^m} f(S)$ or $g(m) = \sup_{S, S' \in \mathcal{Z}^m} f(S, S')$ for some function f, and we expect $g(m)$ to have the form $g(m) = am^b$ for some constants a, b. Thus, for these quantities, we use the following plotting motif: all trials per dataset size are displayed as blue dots, the maximum value per dataset

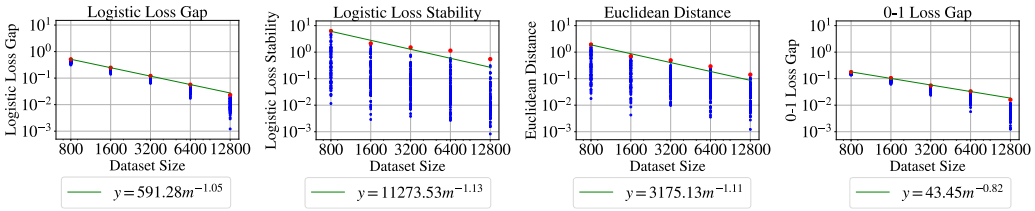


Figure 1: Generalization and stability curves. 90 samples per dataset size m. Each sample involves independently drawing $S \sim \mathcal{D}_{\text{train}}^m$, $z \sim \mathcal{D}_{\text{test}}$, $i \sim U([m])$. $S' := S^{i \leftarrow z}$.

size is a red dot, and a green curve of the form $g(m) = am^b$ is fit to the *red* dots (see Appendix A for curve fitting details). To emphasize b, the rate of decrease (or occasionally increase) with m, these plots display both the x- and y-axes in log scale.

3 Uniform Stability and Generalization

For many years, obtaining useful generalization bounds via uniform stability required $\beta = \mathcal{O}(1/m)$, but [8] (followed by [5]) recently derived tighter bounds of the form: with probability at least $1 - \delta$ over the choice of $S \sim \mathcal{D}^m$,

$$R_{\mathcal{D}}(\mathcal{A}(S)) \leq \widehat{R}_S(\mathcal{A}(S)) + c\left(\beta \log(m) \log(m/\delta) + \frac{\sqrt{\log(1/\delta)}}{\sqrt{m}}\right) \tag{1}$$

for some constant c. Here, $R_{\mathcal{D}}(\mathcal{A}(S))$ is the expected loss over the true distribution, and $\widehat{R}_S(\mathcal{A}(S))$ is the empirical loss evaluated on S. This bound suggests that $R_{\mathcal{D}}(\mathcal{A}(S)) - \widehat{R}_S(\mathcal{A}(S))$ is bounded by $\tilde{\mathcal{O}}(\max\{\beta, 1/\sqrt{m}\})$, hiding logarithmic dependencies inside the $\tilde{\mathcal{O}}$. Thus, if $R_{\mathcal{D}}(\mathcal{A}(S)) - \widehat{R}_S(\mathcal{A}(S))$ empirically decays more slowly than $1/\sqrt{m}$, providing empirical evidence that $R_{\mathcal{D}}(\mathcal{A}(S)) - \widehat{R}_S(\mathcal{A}(S))$ and β decay *similarly* with m would suggest that uniform stability has sufficient strength to explain generalization.

In this section, we present the results of our uniform stability experiments for both logistic regression and neural networks. In both sections, we also carefully estimate $R(\mathcal{A}(S)) - \widehat{R}_S(\mathcal{A}(S))$ as a function of dataset size, under both the 0-1 loss and the logistic or cross-entropy loss, to understand to what degree our uniform stability results are able to capture the strength of generalization. For convenience, we use the phrase "generalization gap" or "loss gap" to denote this difference in test and train loss.

3.1 Logistic regression

As there are obvious challenges in the empirical investigation of uniform stability with respect to the cross-entropy loss, we began by analyzing logistic regression, which presents many of the same challenges (e.g., the logistic loss, how to analyze the suprema over the domain, etc.) but provides a much simpler and better-understood testbed in which to explore our methodology.

Results. In Figure 1, we plot the logistic loss generalization gap, our empirical estimate of the logistic loss uniform stability, the Euclidean distance between the final parameters of $\mathcal{A}(S)$ and $\mathcal{A}(S')$, and the 0-1 loss generalization gap. We fit a curve to the maximum value per dataset size, as described in detail in Section 2 and Appendix A, and we compare the dependence on m of our curves. Among the first three plots, we see a very similar dependence on m, ranging from $m^{-1.05}$ to $m^{-1.13}$. The dependence on m in the 0-1 loss generalization gap plot is a bit weaker ($m^{-0.82}$), but we include this primarily for completeness and as a frame of reference; we are more interested in whether *logistic loss* stability can explain the strength (with respect to m) of generalization with respect to the *logistic loss*.

Conclusions. These plots demonstrate the potential of our methodology to capture, via uniform stability with a *finite* maximum over the test set, the dependence on m of the Euclidean distance between parameters and, most importantly, the logistic loss generalization gap. Thus, although there

ID	Model	Dataset	Learning rate	Batch size	Momentum
1a	ResNet-20	SVHN	0.01 (constant)	32	0.0
2a	ResNet-20	CIFAR-10	0.1, 0.01 at 32k, 0.001 at 48k	128	0.0
2b	ResNet-20	CIFAR-10	0.1, 0.01 at 32k, 0.001 at 48k	128	0.9

Table 1: Deep neural network settings studied.

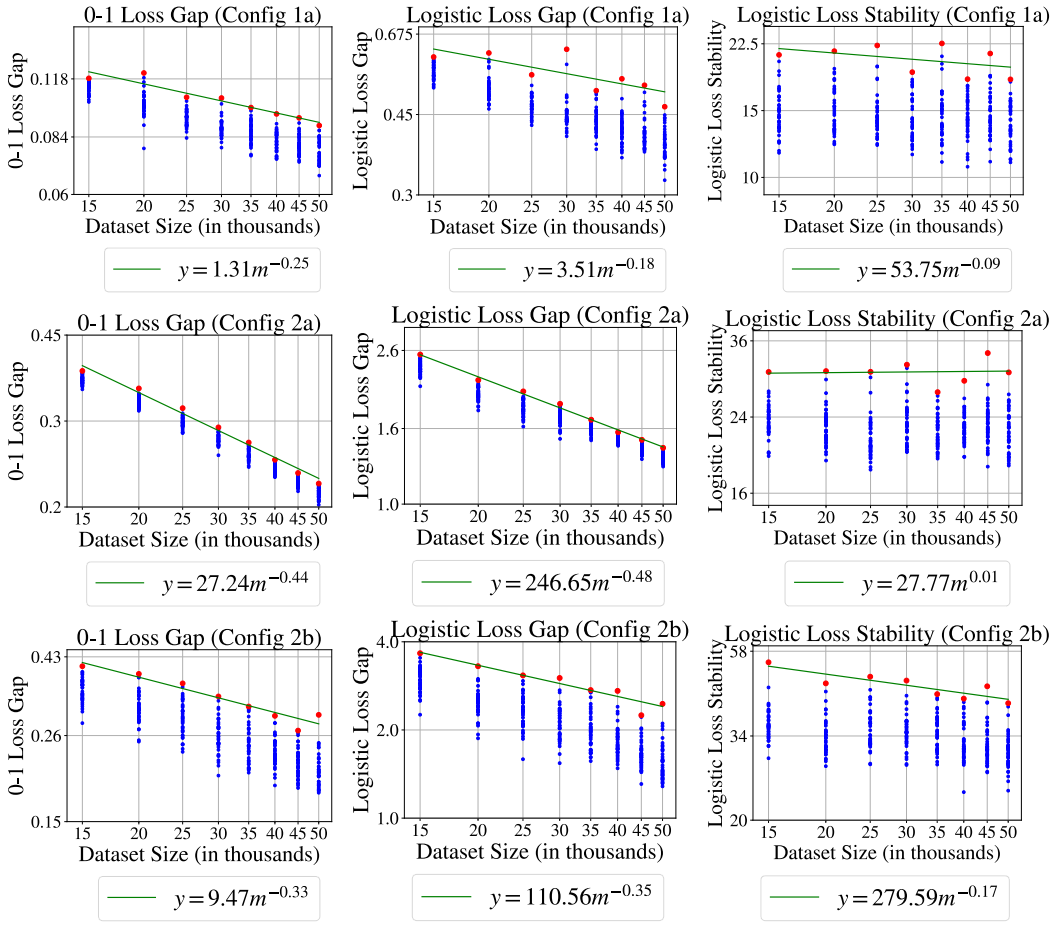


Figure 2: Generalization and stability curves. 40 samples per dataset size m. Each sample involves independently drawing $S \sim \mathcal{D}_{\text{train}}^m$, $z \sim \mathcal{D}_{\text{test}}$, $i \sim U([m])$. $S' := S^{i \leftarrow z}$. Note: We use "logistic loss" and "cross-entropy" loss interchangeably here; all models in this figure were trained and evaluated with the cross-entropy loss.

are obvious differences between the suprema in the definition of uniform stability and our empirical evaluation with finite maxima, our results suggest that there is nevertheless some promise of obtaining informative empirical results.

3.2 Deep learning

After validating our methodology in the simpler setting of logistic regression, we now extend our methodology to the three deep learning configurations described in Section 2.

Results. Figure 2 displays the generalization and stability results for our three neural network settings. In contrast with logistic regression, we postpone examining the parameters themselves until Section 4, in which we conduct an analysis more targeted to deep learning's nonconvex loss landscape.

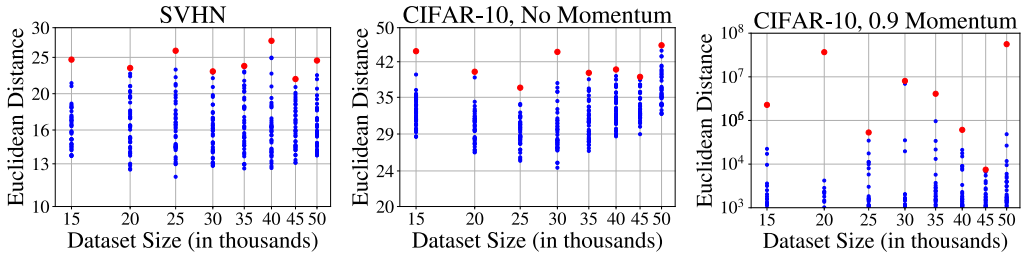


Figure 3: $\|\mathcal{A}(S) - \mathcal{A}(S')\|_2$, at $t = 100,000$, with 40 samples per dataset size m. Each sample involves independently drawing $S \sim \mathcal{D}_{\text{train}}^m$, $z \sim \mathcal{D}_{\text{test}}$, $i \sim U([m])$. $S' := S^{i \leftarrow z}$.

Most significantly, we compare the cross-entropy loss generalization gap to the uniform stability curve. For the ResNet-20 on SVHN, the stability curve displays a mild decrease with m (specifically, $m^{-0.09}$), compared to $m^{-0.18}$ for the cross-entropy loss generalization gap. For the ResNet-20 on CIFAR-10 *without* momentum, the stability curve does *not* decrease with m, despite the cross-entropy loss generalization gap having a dependence of $m^{-0.48}$. For the ResNet-20 on CIFAR-10 *with* momentum, the stability curve displays a mild decrease with m (specifically, $m^{-0.17}$), compared to $m^{-0.35}$ for the cross-entropy loss generalization gap.

To provide a frame of reference, we also compare the cross-entropy loss generalization gap to the 0-1 loss generalization gap and note that, at least for these particular configurations, attempting to explain the rate of decrease with m of the cross-entropy loss generalization gap does not leave us too far from the 0-1 generalization gap either.

Appendix B includes the same experiments repeated with more seeds (for initialization and SGD data order) and includes plots at other stopping points (other than 100,000 iterations).

Conclusions. Overall, in our deep learning experiments, uniform stability with respect to the cross-entropy loss does not appear with sufficient strength to explain observed generalization with respect to the cross-entropy loss.

4 Behavior of Parameters

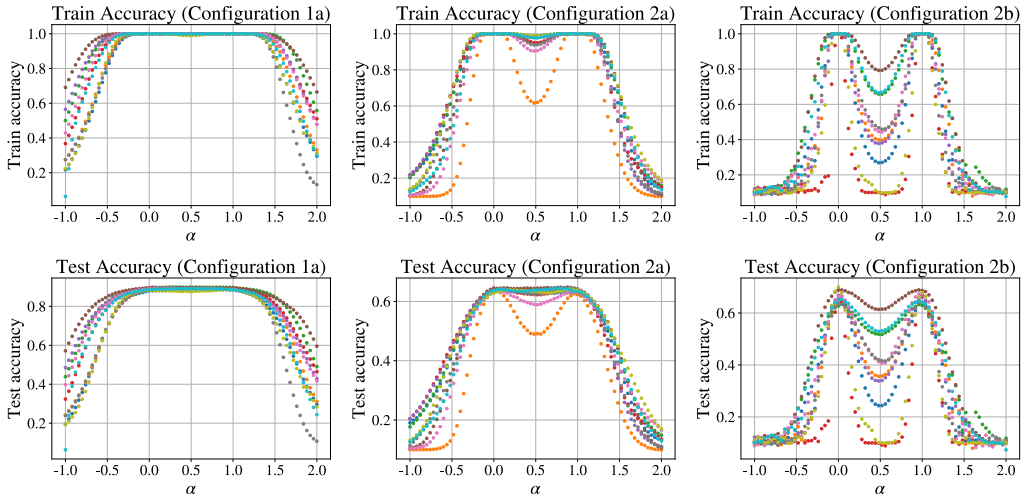
In this section, we analyze the behavior of the underlying parameters to try to disentangle the effect of the cross-entropy loss and the supremum over the domain (estimated via the max over the test set) from the learned models themselves in parameter space.

4.1 Euclidean Distance

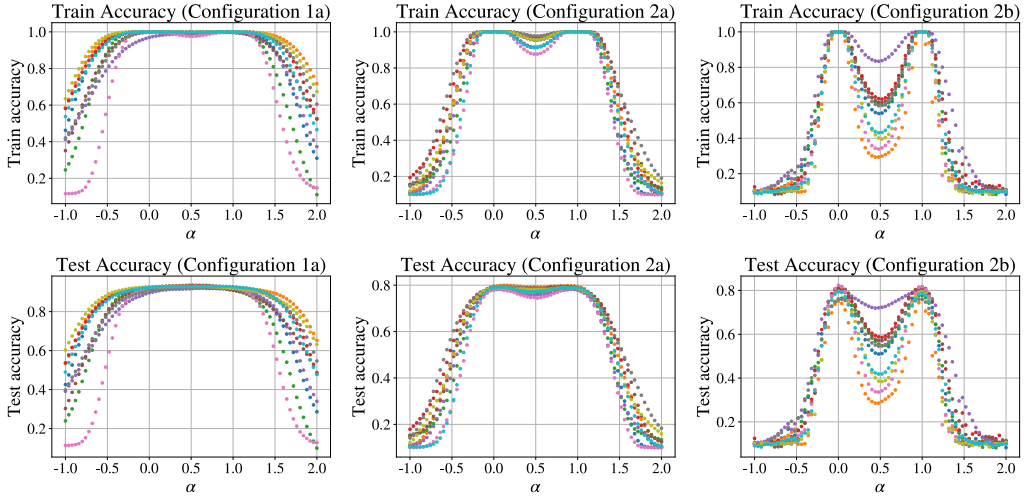
As mentioned in Section 1, we are further interested in studying the Euclidean distance between the final learned parameters to help understand whether the key proof strategy introduced by Hardt et al. [11] extends to practical deep learning settings. Since this paper, most proofs of the stability of SGD (even in nonconvex settings) proceed by bounding the Euclidean distance in parameter space between $\mathcal{A}(S)$ and $\mathcal{A}(S')$ and then appealing to the Lipschitzness of the loss. However, if the Euclidean distance between $\mathcal{A}(S)$ and $\mathcal{A}(S')$ does not decrease with dataset size in our trained models, this suggests that this proof strategy might not be sufficient for obtaining generalization bounds in practical deep learning settings that hold with high probability (over the random draw of the dataset and the random initialization and SGD data order of the algorithm).

Results. Figure 3 presents $\|\mathcal{A}(S) - \mathcal{A}(S')\|_2$ for our three neural network configurations. We see that, from $m = 15k$ to $m = 50k$, the distances do not decrease with dataset size at a sufficient rate to explain generalization and actually even increase in some dataset size ranges.

Conclusions. These results suggest that a decrease in Euclidean distance of the parameters with dataset size is likely not a viable path through which to prove stability in *practical* deep learning settings.



(a) $m = 15,000$.



(b) $m = 50,000$.

Figure 4: Interpolation at $t = 100,000$.

One might ask whether the nondecreasing Euclidean distance we observe here is caused by the norms in parameter space *themselves* growing with dataset size. We first emphasize that this question does not impact our conclusions, as the proofs to which we have referred invoke the raw Euclidean distance between the parameters. However, for completeness, we refer the interested reader to Appendix C for an extensive analysis of norms and normalized Euclidean distances.

4.2 Linear Mode Connectivity

We now ask the question: Is nonconvexity causing optimization on S and S' to diverge to different basins of attraction, thus thwarting efforts to extend analyses from convex settings to deep learning's nonconvex setting (as is done by [11] and follow-up works)? Here, we use *basin of attraction* to mean a convex set of solutions (in parameter space) all with comparable training and/or test loss.

To make this more precise, we invoke the *linear mode connectivity* framework of Frankle et al. [10] to study this question. Specifically, linear mode connectivity asks whether, at all networks along the linear path between two candidate networks (in parameter space), the training and/or test error does not increase. In our setting, W_S and $W_{S'}$ qualify as linearly connected modes if, for all $\alpha \in [0, 1]$,

the (test or train) accuracy of the model with parameters $\alpha W_S + (1-\alpha)W_{S'}$ is not significantly below that of W_S or $W_{S'}$ (roughly 2%, per [10]).

Results. We plot the train accuracy (on S) and test accuracy at $\alpha W_S + (1-\alpha)W_{S'}$ at 76 equally-spaced values of $\alpha \in [-1, 2]$. The learned parameters are at $\alpha = 0$ and $\alpha = 1$, but we include additional values of α on either end as a frame of reference. We randomly select 10 trials among the 40 trials described in Section 2 and, for each trial, we plot the train and test accuracy for each value of α. Figure 4a has results for $m = 15,000$ and Figure 4b has results for $m = 50,000$. The ResNet-20 on SVHN has nondecreasing accuracy when linearly interpolating between W_S and $W_{S'}$, the ResNet-20 on CIFAR-10 without momentum has slightly decreasing accuracy when linearly interpolating between W_S and $W_{S'}$, and the ResNet-20 on CIFAR-10 with momentum has significantly decreasing accuracy when interpolating.

Conclusions. A priori, it is not obvious what one should expect when linearly interpolating, and it is thus perhaps surprising that our three configurations largely span the space of possibilities. Thus, in order to further study the weaknesses of uniform stability in practical deep learning settings, we suggest that moving to a regime such as a ResNet-20 on CIFAR-10 *with* momentum, in which the solutions are *not* connected by a path of nondecreasing accuracy, might not be immediately necessary from a scientific standpoint. Perhaps the limitations of uniform stability can be explored and better understood with a configuration such as our ResNet-20 on SVHN (without momentum). Notably, the SVHN interpolation results suggest that nonconvexity might not be necessary at all to investigate the particular weaknesses of algorithmic stability experienced by deep learning; rather, *convex* settings with large enough basins of attraction (defined for our purposes as convex sets of parameters yielding approximately equal training and/or test loss) to host a reasonable degree of functional diversity might actually be subject to these same weaknesses. Thus, our findings might open the door to the study of more tractable, convex settings in which one can study the same limitations of algorithmic stability that appear in deep learning.

5 Conclusions and Future Work

In this work, we have initiated the challenging endeavor of empirically studying the uniform stability of deep learning. Although we freely admit that no reasonable empirical results could *definitively* rule out uniform stability (due to its formulation as several maxima over the domain), we believe that our results present compelling evidence that (a) uniform stability (with respect to the cross-entropy loss) might not be present in practical deep learning with sufficient strength to explain generalization, and (b) that typical theoretical approaches based on parameter distance decreasing with dataset size are likely not the driving force behind any form of algorithmic stability that nevertheless might exist in deep learning. Ultimately, if some form of algorithmic stability (perhaps weaker than uniform stability) is at play in deep learning, we suspect that it will stem from a function-space view that appropriately handles divergence to different basins of attraction after swapping one data point (as seen in Section 4.2, Configuration 2b). However, in the meantime, we present compelling evidence that many of the weaknesses of uniform stability can already be seen empirically in simpler, perhaps even convex, settings. We believe that formalizing and further investigating these more tractable settings presents an exciting direction for future work.

Broader Impact

Overall, we believe that this work has relatively minimal societal impact. Ultimately, in the long term, we do hope that this work will contribute to our collective understanding of how deep learning works, which is increasingly critical as deep learning is deployed in an ever-growing range of real-world applications. We see this as a potential positive benefit of our work. On the negative side, we do not envision any major *harm* from this work, other than the environmental cost of running these experiments. However, our hope is that, as we gain a better understanding of deep learning as a society, the need for so many large-scale scientific experiments will eventually subside, and we will be better equipped to predict the behavior of deep learning through theory (at least more than we are able to do at present).

References

[1] K. Abou-Moustafa and C. Szepesvári. An exponential efron-stein inequality for l_q stable learning rules. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, 2019.

[2] P. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1706.08498*, 2017.

[3] P. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20:1–17, 2019.

[4] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning*, 2: 499–526, 2002.

[5] O. Bousquet, Y. Klochkov, and N. Zhivotovskiy. Sharper bounds for uniformly stable algorithms. *arXiv preprint arXiv:1910.07833*, 2019.

[6] G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[7] G. F. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio. Large margin deep networks for classification. In *Proceedings of NeurIPS*, 2018.

[8] V. Feldman and J. Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Proceedings of COLT*, 2019.

[9] D. J. Foster, S. Greenberg, S. Kale, H. Luo, M. Mohri, and K. Sridharan. Hypothesis set stability and generalization. In *Proceedings of NeurIPS 2019*, pages 6726–6736, 2019.

[10] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[11] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[13] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio. Fantastic generalization measures and where to find them. In *Proceedings of ICLR*, 2020.

[14] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[15] S. Kutin and P. Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of UAI*, pages 275–282, 2002.

[16] I. Kuzborskij and C. H. Lampert. Data-dependent stability of stochastic gradient descent. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[17] T. Liang, T. Poggio, A. Rakhlin, and J. Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

[18] T. Liu, G. Lugosi, G. Neu, and D. Tao. Algorithmic stability and hypothesis complexity. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[19] P. M. Long and H. Sedghi. Generalization bounds for deep convolutional neural networks. In *Proceedings of ICLR*, 2020.

[20] V. Nagarajan and J. Z. Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Proceedings of NeurIPS*, 2019.

[21] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning.*, 2011.

[22] B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

[23] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[24] B. Neyshabur, S. Bhojanapalli, and N. Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *Proceedings of ICLR*, 2018.

[25] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *Proceedings of ICLR*, 2017.

A Further Methodology Details

Dataset splits. MNIST: From the 60,000 training examples, we randomly sampled subsets as specified in Section 2 for training. We used the full 10,000-element test set for evaluation (including computation of uniform stability, as specified in Section 2).

CIFAR-10: From the 50,000 training examples, we randomly sampled subsets as specified in Section 2 for training. We used the full 10,000-element test set for evaluation (including computation of uniform stability, as specified in Section 2).

SVHN: From the 73,257 training examples, we randomly sampled subsets as specified in Section 2 for training. To maintain consistency with MNIST and CIFAR-10, we randomly sampled 10,000 elements from the 26,032-element test set for evaluation. All datasets were normalized in the same manner, by dividing each coordinate by 255.

Selection of hyperparameters. The hyperparameters for ResNet-20 on CIFAR-10 are derived from [12]. The hyperparameters for logistic regression were chosen similarly, intentionally without momentum (since our primary goal was to study SGD) and without a decaying learning rate. The hyperparameters for Configuration 1a were intentionally chosen to vary from Configurations 2a and 2b, in order to create more diversity in our hyperparameter settings; specifically, we deemed it valuable to investigate a smaller batch size (i.e., 32) without momentum, and the corresponding learning rate of 0.01 worked fairly well with this batch size.

Stopping criterion details. We considered three different possible stopping criteria: parameter updates, epoch number, and average training loss. We performed preliminary analyses with all stopping criteria, but after careful consideration, we ultimately chose to focus our analysis on *parameter updates* for the following reasons: (1) parameter updates align with theoretical analyses of uniform stability, such as [11, 8], in which the stability parameter is expressed as a function of the number of parameter updates; and (2) parameter updates appear to give uniform stability the *best* chance at succeeding in explaining generalization, thus making our *negative* results more significant. Specifically, if instead we were to hold the number of epochs fixed across dataset sizes, this means that *larger* datasets would take more steps. This is true for average training loss as a stopping criterion as well, as it typically takes more steps for larger datasets to reach the same average training loss as smaller datasets. Thus, although these stopping criteria are perhaps truer to practice, we believe that they make it even *easier* for uniform stability to fail to explain the strength of generalization.

Dataset size range. We chose to limit our analysis to the ranges specified in Section 2 for the following reason. In order to ask the question *Can the strength of decrease with m in our generalization gap be explained by uniform stability?*, we wanted a rate of decrease with m that would be roughly constant in our dataset size range. Figure 5 shows a plot and curve fit on a normal-scale plot, followed by a log-log plot. Although the curve fit displays some room for improvement in the original plot, the log-log plot reveals different regions of decrease with m. Through this plot and additional such investigations, we noticed that the dataset size range 15,000-50,000 yielded the largest window with a roughly consistent rate of decrease with m. Thus, we chose to focus our analysis on this window in order to draw more meaningful conclusions. As deep learning models are typically trained in large-data regimes, this decision aligns with practical considerations as well.

Curve fitting details. We used `scipy`'s `optimize` package, specifically the `curve_fit` function. We fit parameters a and b in $y = am^b$, where m is the dataset size and y is the metric of interest.

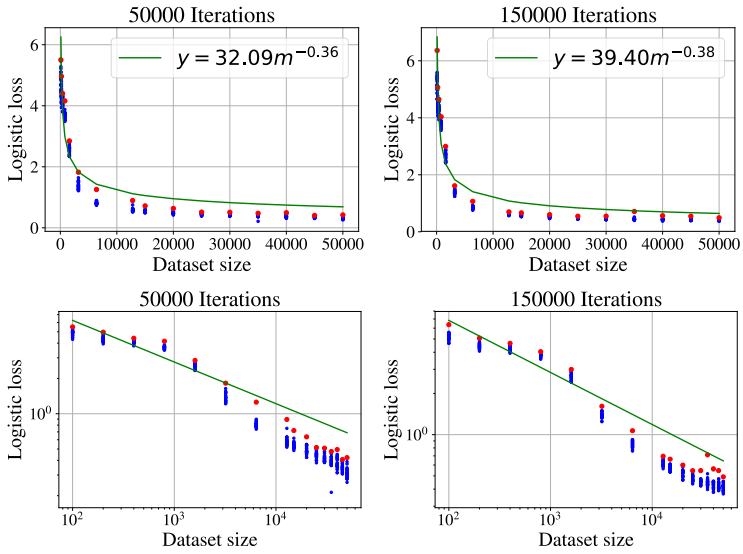


Figure 5: Configuration 1a, with 20 samples per dataset size m. Each sample involves independently drawing $S \sim \mathcal{D}_{\text{train}}^m$, $z \sim \mathcal{D}_{\text{test}}$, $i \sim U([m])$. $S' := S^{i \leftarrow z}$.

B Further Experiments for Section 3

In this section, we present stability and generalization results for two additional seeds (Trials 2 and 3) and compare them to the original seed presented in the main paper (Trial 1). Our figures are as follows:

- Figure 6 has all trials for Configuration 1a (SVHN).
- Figure 7 has all trials for Configuration 2a (CIFAR-10, no momentum).
- Figure 8 has all trials for Configuration 2b (CIFAR-10, 0.9 mometum).
- Figure 9 has generalization (cross-entropy only) and stability results for Iteration 50,000 for Configuration 1a.
- Figure 10 has generalization (cross-entropy only) and stability results for Iteration 50,000 for Configuration 2a.
- Figure 11 has generalization (cross-entropy only) and stability results for Iteration 50,000 for Configuration 2b.
- Figure 12 has generalization (cross-entropy only) and stability results for Iteration 150,000 for Configuration 1a.
- Figure 13 has generalization (cross-entropy only) and stability results for Iteration 150,000 for Configuration 2a.
- Figure 14 has generalization (cross-entropy only) and stability results for Iteration 150,000 for Configuration 2b.

The additional trials are roughly consistent with the trial highlighted in the main paper.

C Further Experiments for Section 4

In this section, we present further experiments regarding regarding the Euclidean distance between $\mathcal{A}(S)$ and $\mathcal{A}(S')$, parameter norms, and normalized Euclidean distances. Our figures are as follows:

- Figure 15 presents additional trials for $\|\mathcal{A}(S) - \mathcal{A}(S')\|_2$ at Iteration 100,000.
- Figure 16 has $\|\mathcal{A}(S)\|_2$ at Iteration 100,000.
- Figure 17 has normalized Euclidean distances, with further details in the caption.

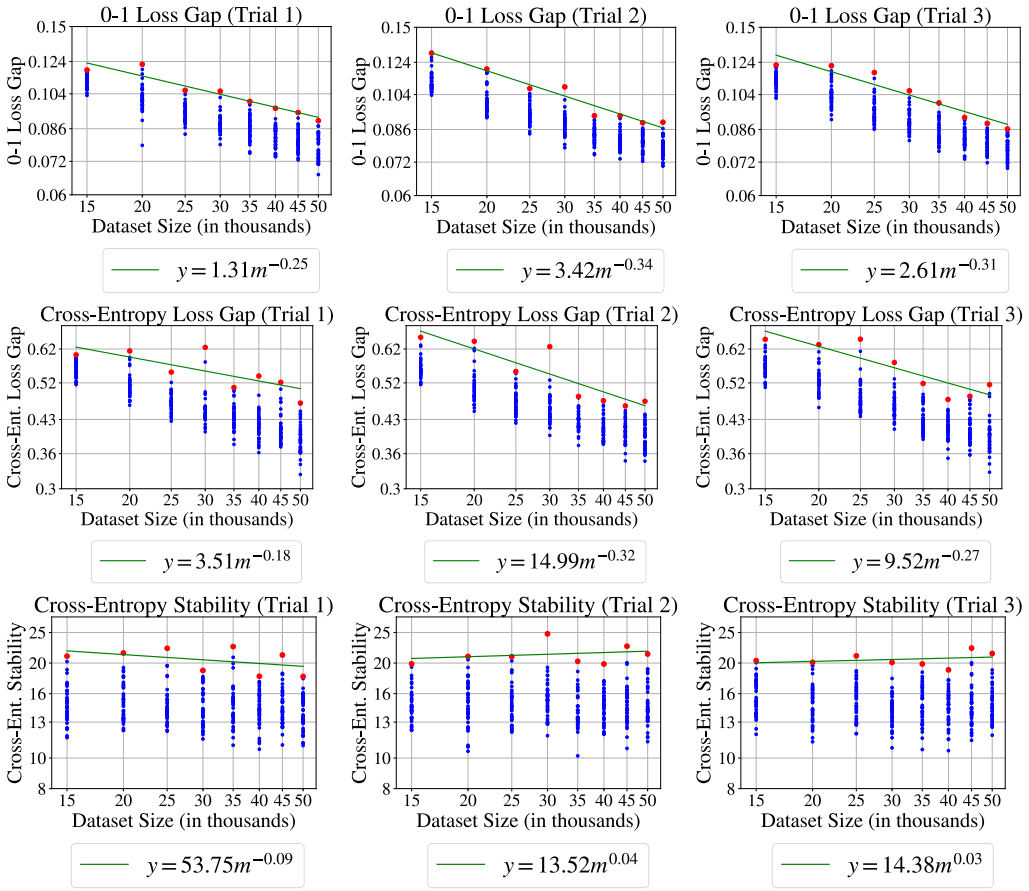


Figure 6: All trials for Configuration 1a (SVHN).

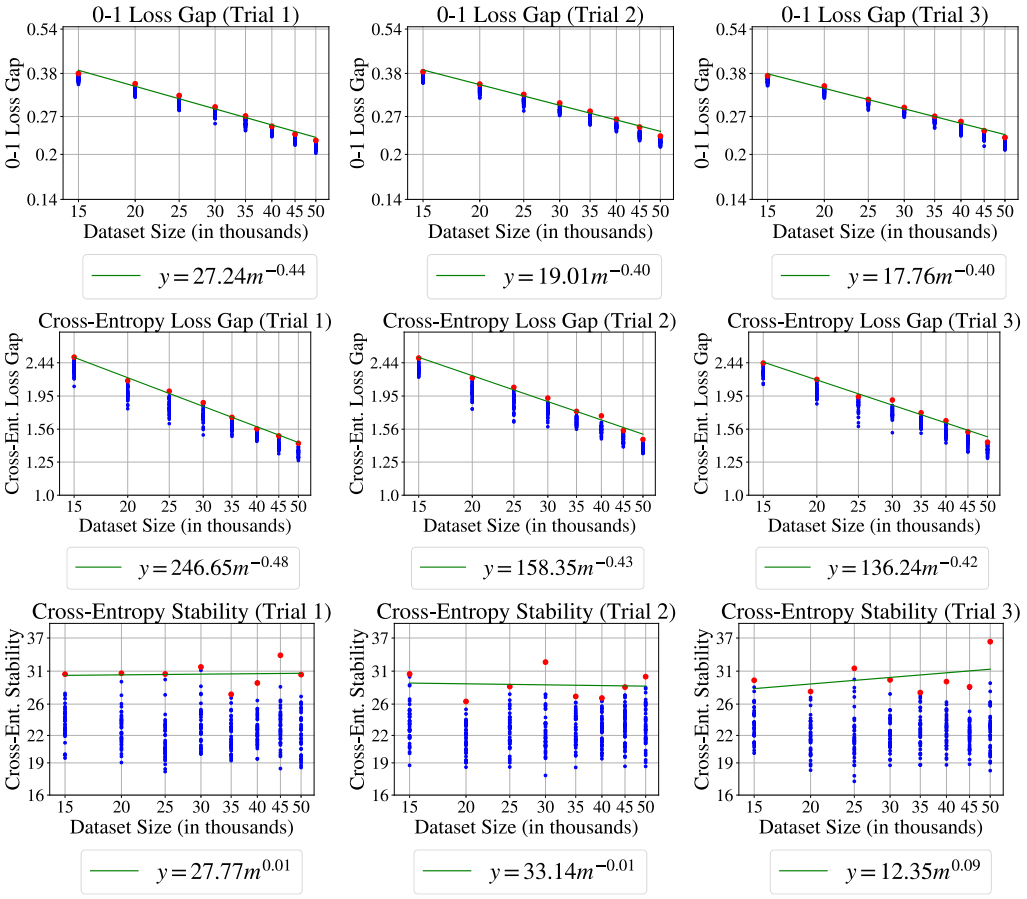


Figure 7: All trials for Configuration 2a (CIFAR-10, no momentum).

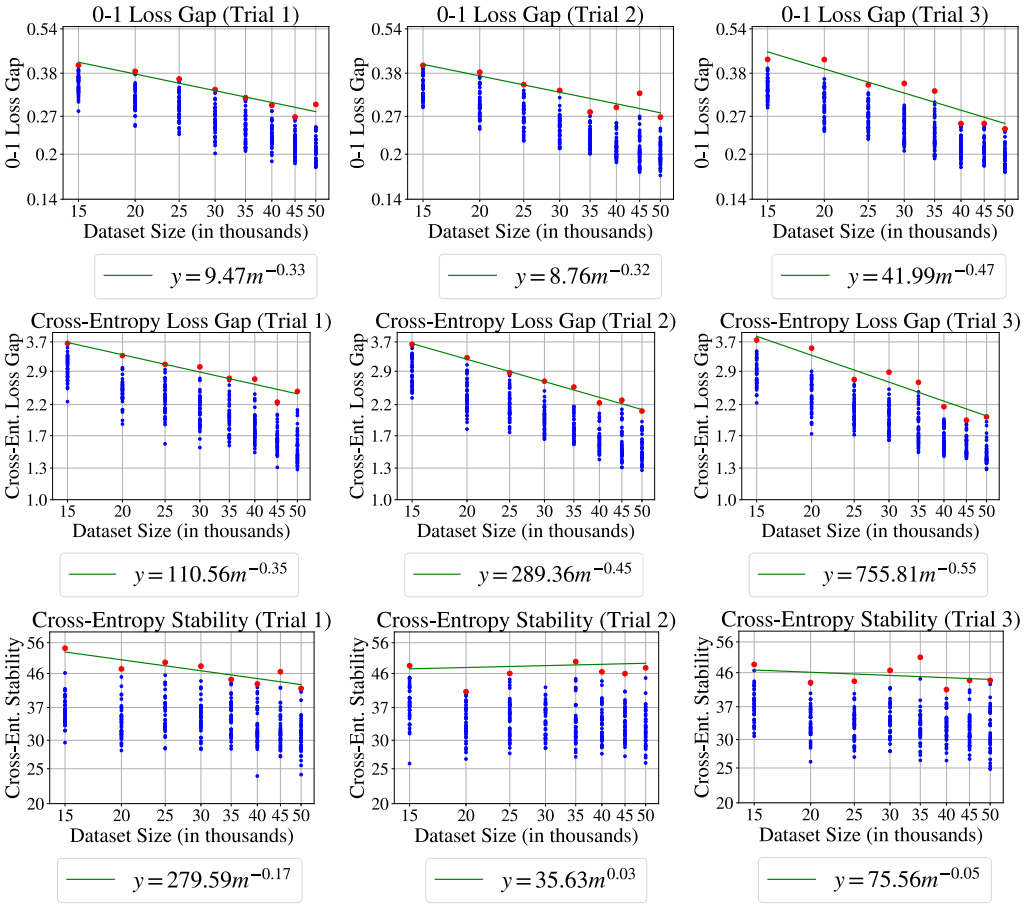


Figure 8: All trials for Configuration 2b (CIFAR-10, 0.9 momentum).

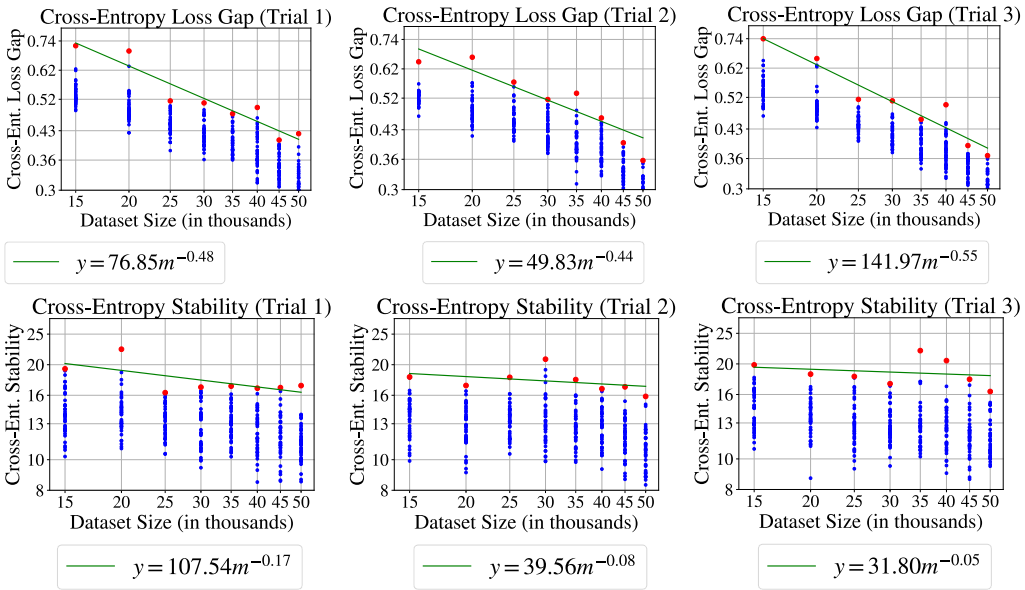


Figure 9: Iteration 50,000 for Configuration 1a.

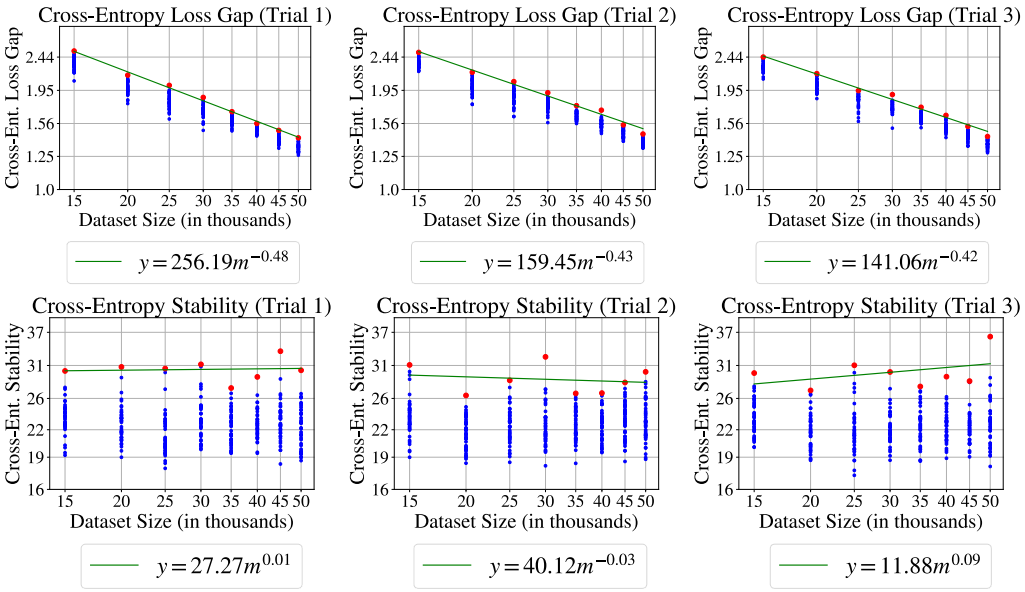


Figure 10: Iteration 50,000 for Configuration 2a.

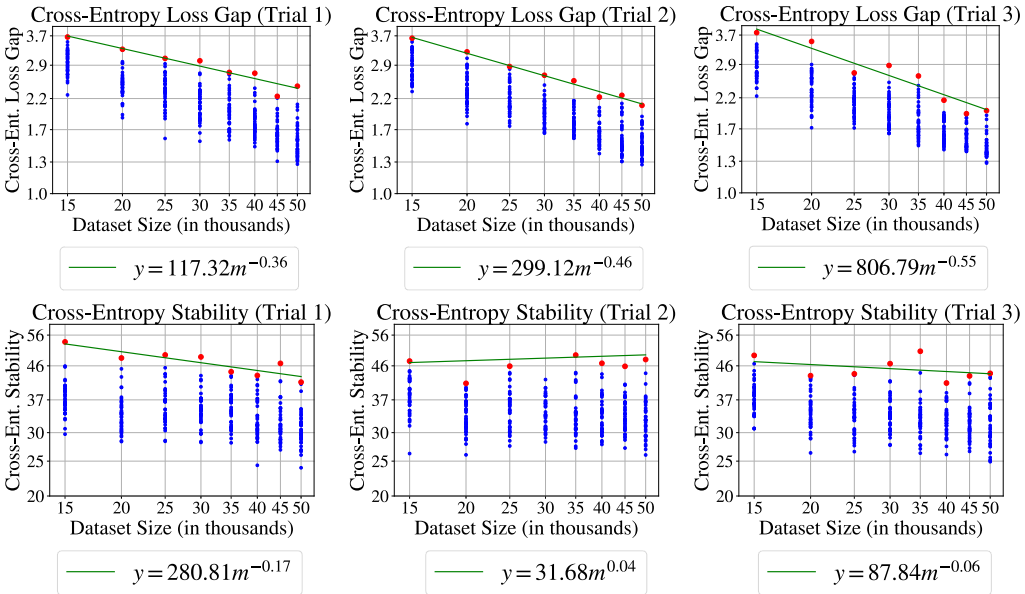


Figure 11: Iteration 50,000 for Configuration 2b.

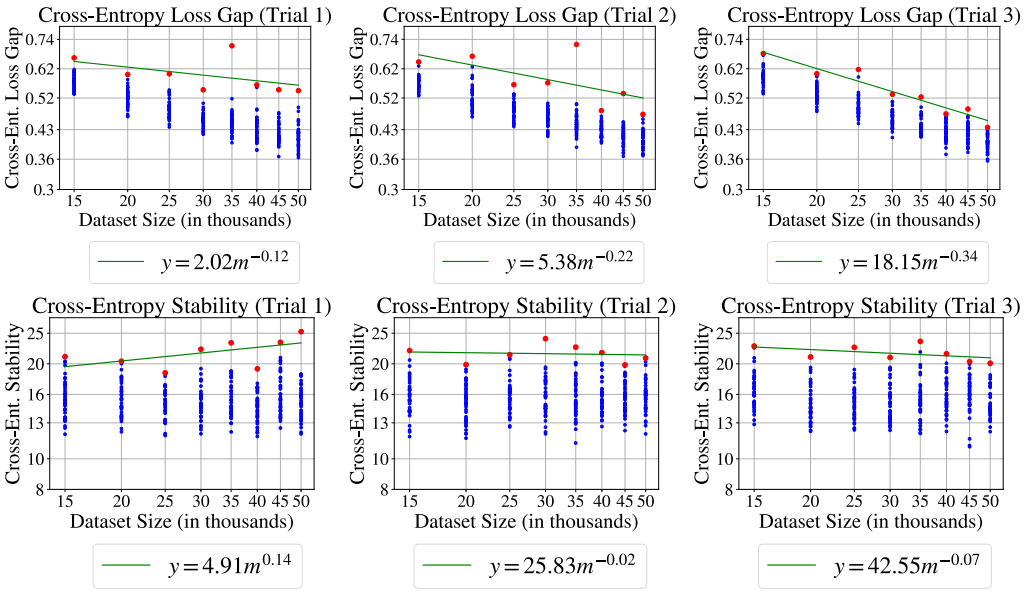


Figure 12: Iteration 150,000 for Configuration 1a.

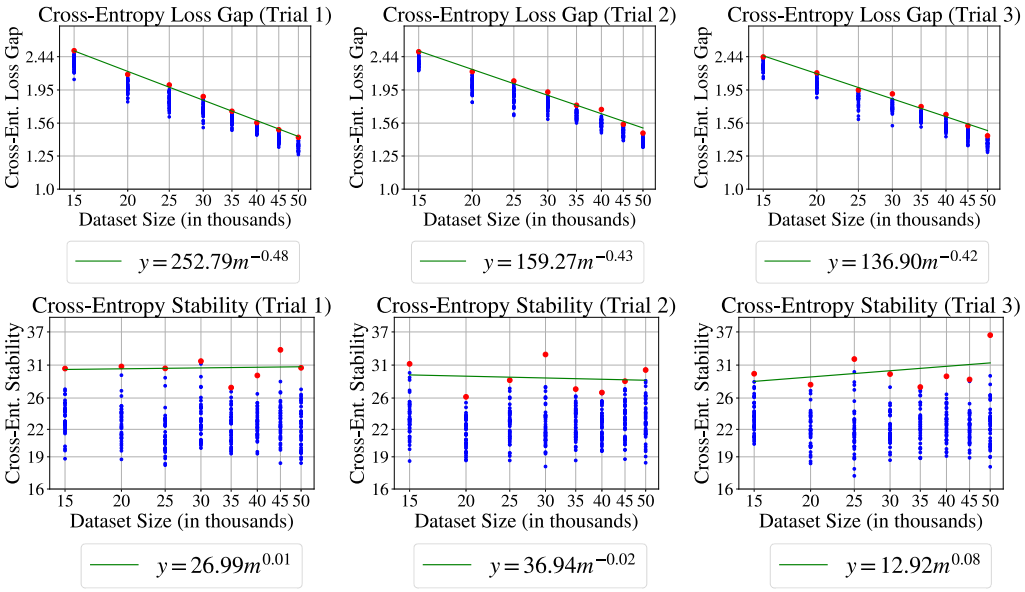


Figure 13: Iteration 150,000 for Configuration 2a.

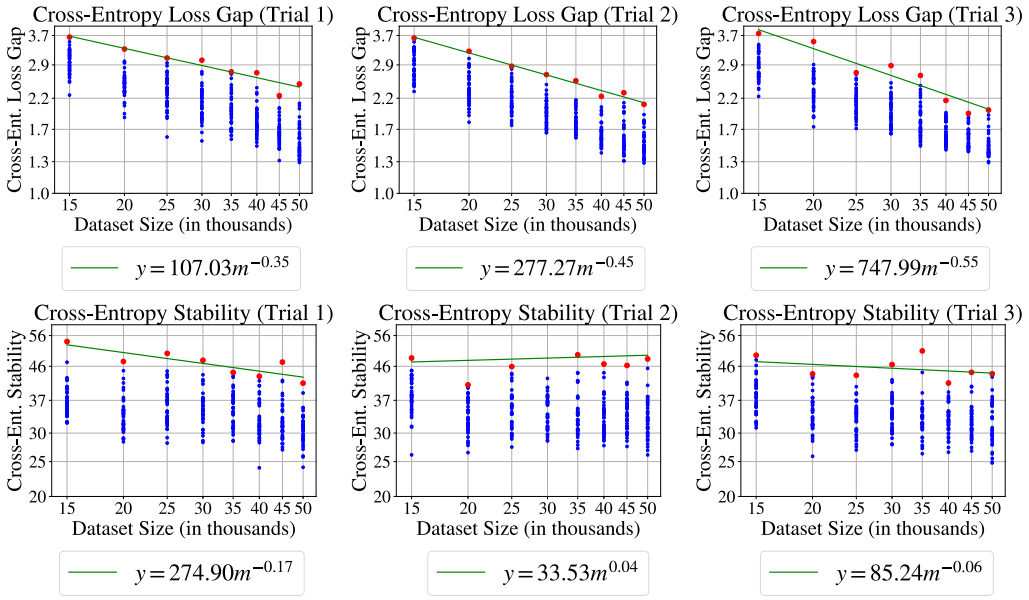


Figure 14: Iteration 150,000 for Configuration 2b.

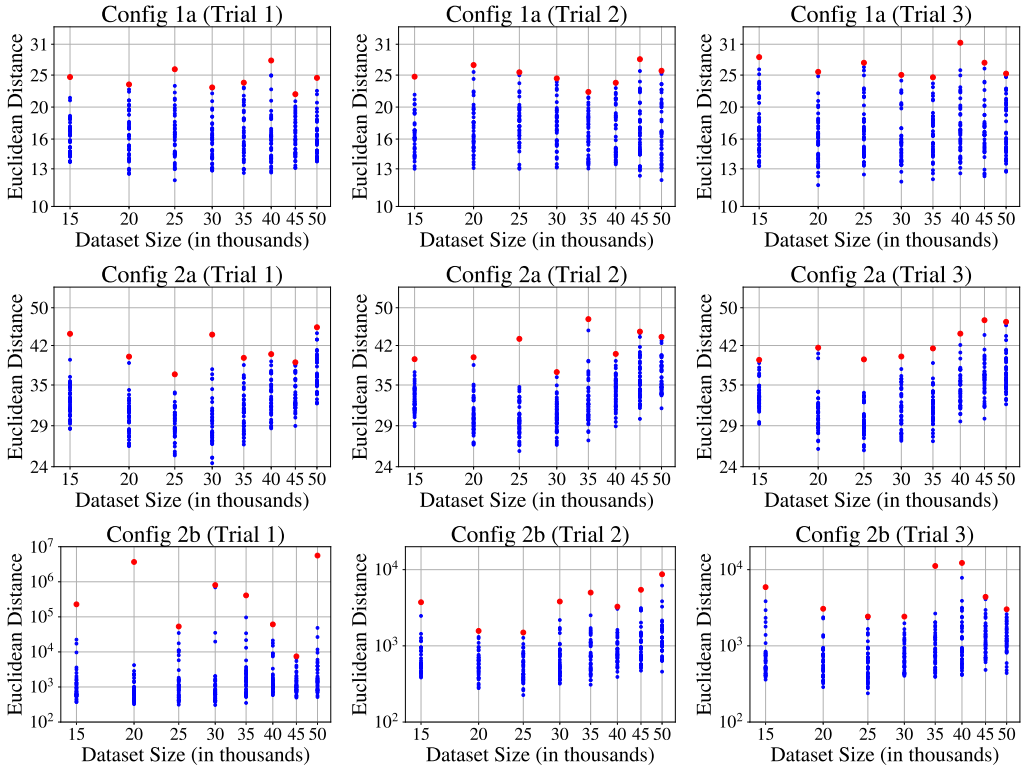


Figure 15: All trials for $\|\mathcal{A}(S) - \mathcal{A}(S')\|_2$ at $t = 100,000$.

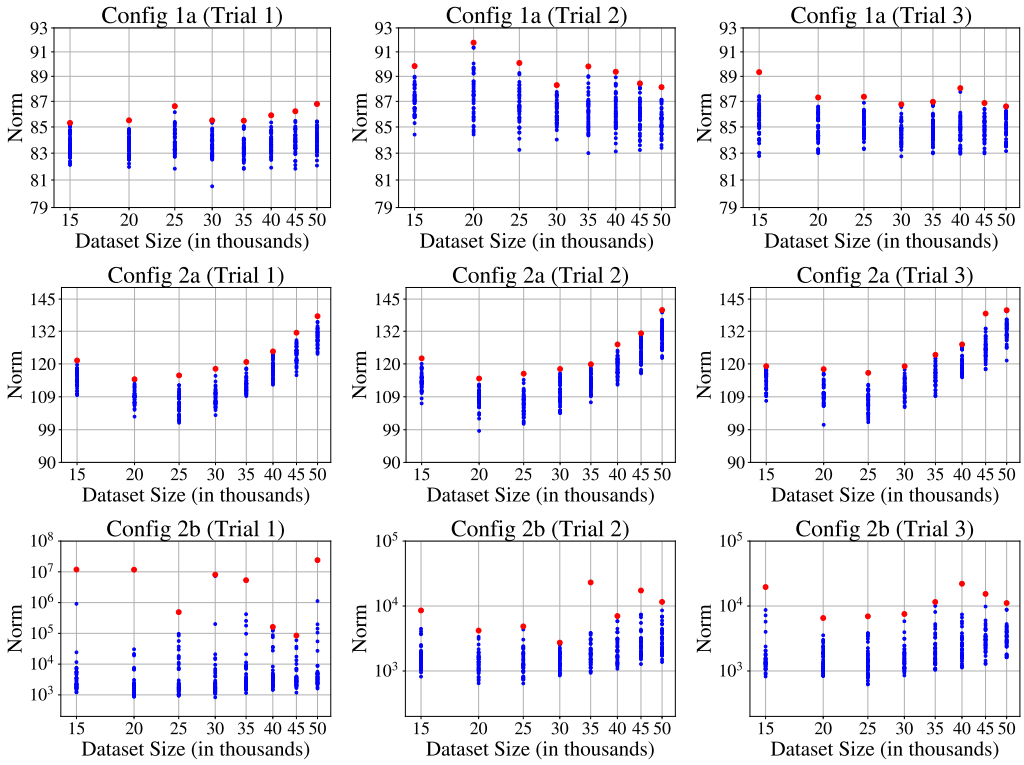


Figure 16: $\|\mathcal{A}(S)\|_2$ at $t = 100,000$.

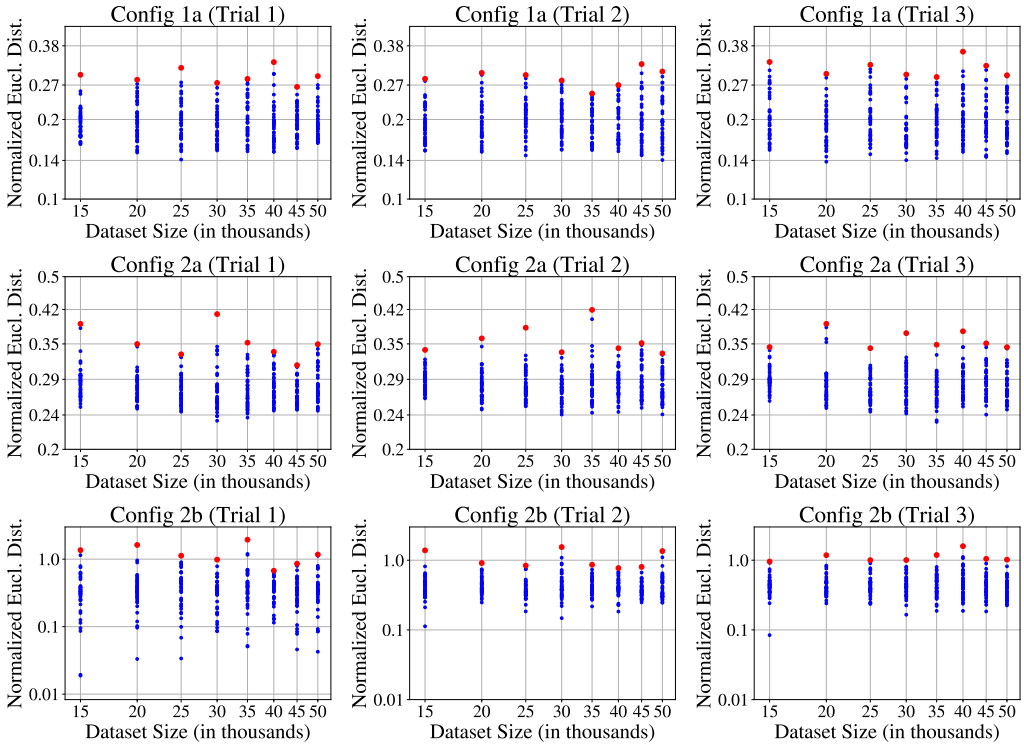


Figure 17: Normalized Euclidean distance. For each Euclidean distance $\|A(S) - A(S')\|$, we divide by $(\|A(S)\| + \|A(S')\|)/2$. The results suggest that normalizing largely mitigates the growth in Euclidean distance with dataset size; however, this does not appear to yield a significant *decrease* in Euclidean distance with dataset size.

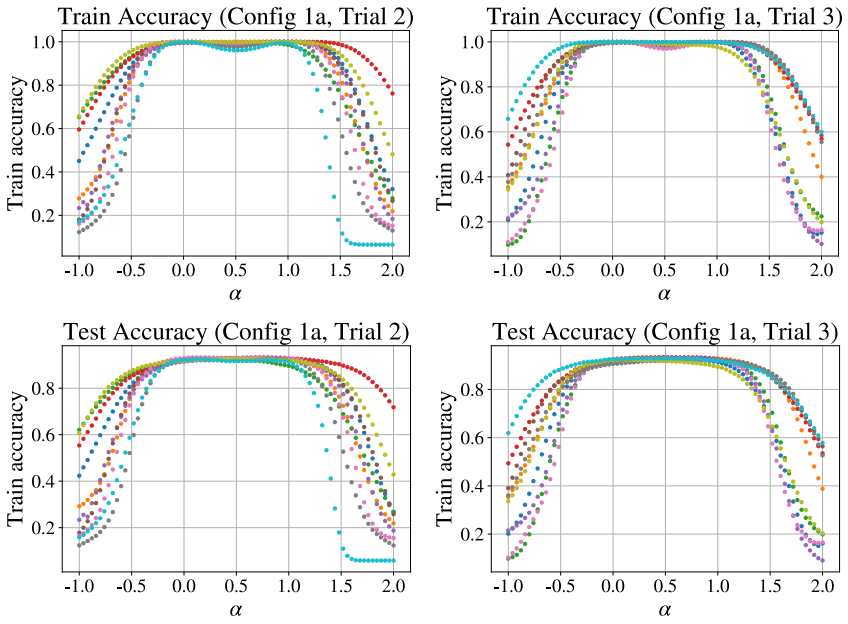


Figure 18: Additional interpolation trials: Configuration 1a (SVHN).

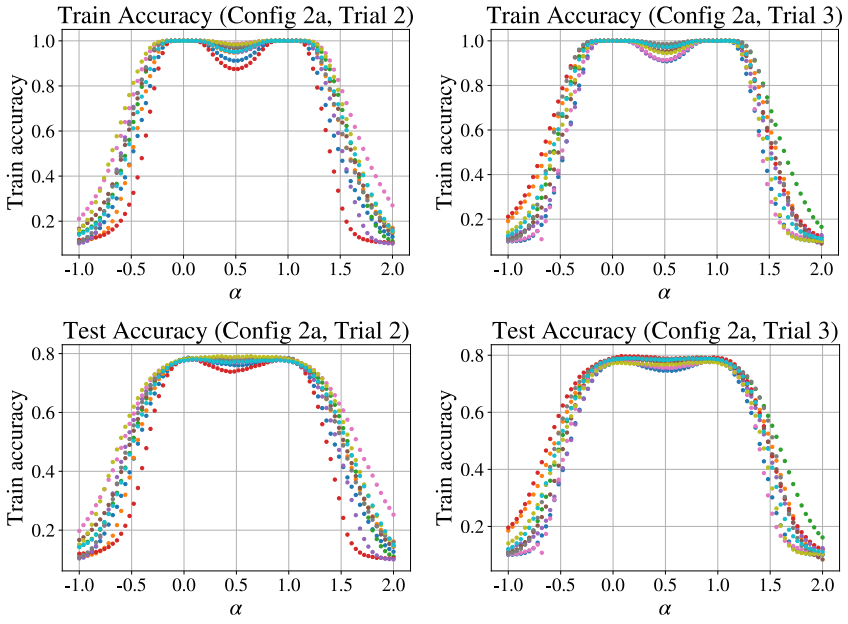


Figure 19: Additional interpolation trials: Configuration 2a (CIFAR-10, no momentum). Note: We omit Configuration 2b from additional trials because the lack of connectivity seen in the body of the paper is not our focus in these additional trials; rather, we are simply interested in confirming cases of linear mode connectivity.