Learning from setbacks: the impact of adversarial initialization on generalization performance

Kavya Ravichandran*
TTIC
kavya@ttic.edu

Yatin Dandi
EPFL
yatin.dandi@epfl.ch

Stefani Karp
CMU & Google Research
shkarp@cs.cmu.edu

Francesca Mignacco
Princeton University & CUNY
fmignacco@princeton.edu

Abstract

The loss landscape of state-of-the-art neural networks is far from simple. Understanding how optimization algorithms initialized differently navigate such high-dimensional non-convex profiles is a key problem in machine learning. Liu et al. (2020) use pre-training on random labels to produce adversarial initializations that lead stochastic gradient descent into global minima with poor generalization. This result contrasts with other literature arguing that pre-training on random labels produces positive effects (see, e.g., Maennel et al. (2020)). We ask under which conditions this initialization results in solutions that generalize poorly. Our goal is to build a theoretical understanding of the properties of good solutions by isolating this phenomenon in some minimal models. To this end, we posit and study several hypotheses for why the phenomenon might arise in models of varying levels of simplicity, including representation quality and complex structure in data.

1 Introduction

Overparametrized models that can fit even random data (Zhang et al., 2021) are able to find generalizing solutions despite searching a high-dimensional loss landscape with multiple "bad" minima (Keskar et al., 2016). Even more surprisingly, this success can be achieved by relatively simple gradient-based algorithms initialized at random (Chizat and Bach, 2018; Du et al., 2019). Despite the increasing theoretical effort to understand the performance of overparametrized models, the knowledge of the underlying mechanisms is still sparse. A successful and influential research direction to address this puzzle is to study the implicit bias that leads training algorithms to pick good solutions (see, e.g., Neyshabur et al. (2014); Soudry et al. (2018); Gunasekar et al. (2018)).

In this paper, we adopt a complementary perspective by considering learning failures: we investigate the properties of "bad" solutions in order to identify the key missing ingredients that are necessary for generalization. We explore the conditions under which gradient-based algorithms converge to bad minima, even when substantial changes in model parameters occur. One method to identify such bad minima is through initializations that empirically lead to poorly generalizing solutions: we refer to these as *adversarial* initializations. For example the adversarial initialization method of Huang et al. (2020) relies on training the model until it perfectly fits not only the training data but also additional data with randomized labels. Liu et al. (2020) have proposed a different technique to find these adversarial initializations leading SGD to poorly-generalizing global minima. In particular, they consider two consecutive training phases:

*Corresponding author

Mathematics of Modern Machine Learning Workshop at NeurIPS 2023.

Phase 1: train the network using *random* labels until 100% training accuracy.

Phase 2: train on the original task, but initialized at the solution found in phase 1.

They find that the *final* solution starting from this adversarial initialization generalizes poorly on the original task. This detrimental effect of initialization is removed when adding explicit regularization. We exploit the same procedure to further investigate what about these minima *makes* them bad and how this relates to pre-training on random labels. Indeed, there is no consensus on this point and other works suggest that pre-training on random labels could help training (Maennel et al., 2020). Recent empirical work (Chiang et al., 2022) further suggests that the volume of bad minima could be significantly smaller compared to the volume of good minima in deep neural networks trained on real datasets. Understanding why adversarial initializations still cause convergence to bad minima despite their small volume would therefore shed light into why typical initializations don't. Crucially, we are not interested in asymptotic-time results; instead we consider timescales that can be reached in practice by numerical experiments.

2 Experimental Framework and Proposed Hypotheses

We study a supervised learning task with training dataset $\boldsymbol{X}_{\text{train}} \in \mathbb{R}^{n \times d}$ and true labels $\boldsymbol{y}_{\text{train}} \in \mathbb{R}^n$, where n denotes the number of training samples and d the input dimension. We also consider a randomized version of the labels, that we call $\boldsymbol{y}_{\text{rand}}$. Liu et al. (2020) train different neural networks architectures (VGG16, ResNet18, ResNet50, and DenseNet40) with SGD on CIFAR, CINIC10, and ImageNet datasets and randomized labels until a solution is found. We call this solution $\boldsymbol{\Theta}_{\text{adv}}$. They observe that retraining from $\boldsymbol{\Theta}_{\text{adv}}$ on $(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ leads to poorer generalization error compared to training on $(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ from random initialization. We hypothesize several possible explanations for this phenomenon and design a series of experiments to discern which of these may hold. We aim to isolate the phenomena observed by Liu et al. (2020) in minimal models that are amenable to theoretical analyses. In this section, we describe each of these hypotheses and the experiments to address them. In Sec. 3, we discuss how our findings relate to the posited explanations. Finally, in Sec. 4 we present the next steps to further our investigation. We conjecture that poor generalization from adversarial initialization could be caused by:

1. *Norm of the solution:* the large norm of Θ_{adv} leads to poor generalization.
2. *Quality of representation:* adversarial initialization produces bad representations that are hard to escape. We ask which layer is more responsible for escaping bad minima. To this end, we freeze the first layer weights during *phase 2* and compare the performance gaps.
3. *Complex structure in data:* Given that we are pre-training with the same examples (only randomizing labels), something about the data structure should be memorized by the model. What is the impact of this "memorization" bias on *phase 2*?

As a simple check to rule out the first hypothesis, we scale up the weights of a randomly-initialized VGG16 model so that the norm of the overall weights matches that of the adversarial initialization. The test accuracy is higher than in the adversarial initialization case but still does not match the baseline, suggesting norm accounts for part of the problem but the shape of the initialization matters, too. For further investigation, we develop the other two hypotheses. Our findings are detailed in Sec. 3.

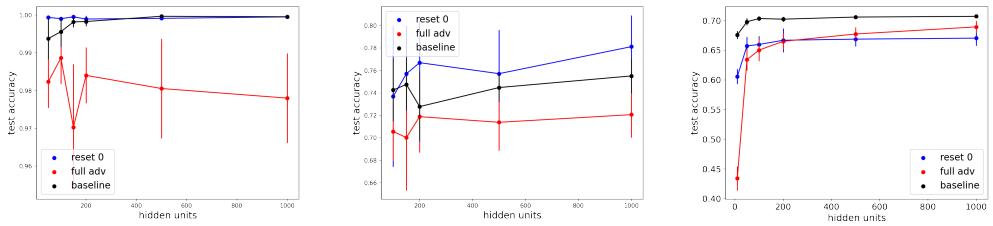
3 Classification Experiments to Study Representation Quality

We consider simple architectures to assess the generality of this behaviour. We use a fully-connected network with one-hidden-layer of p hidden units and ReLU activation, which we call $\mathcal{N}_p$ for succinctness, trained on $n = 100$ samples using gradient descent with step size 0.01 for the experiments described below (except for MNIST, where we use SGD with batch size 32 and step size 0.1).

3.1 Experiments

Experiment 1: We train $\mathcal{N}_p$ on $(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ until zero training error is achieved. Note that in the classification setting, zero training error does not correspond to zero training loss, since logistic loss is a proxy for the $0-1$ loss and only achieves zero asymptotically.

Experiment 2: In phase 1, we train $\mathcal{N}_p$ on $(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{rand}})$ until zero training error is achieved at a solution $\boldsymbol{\Theta}_{\text{adv}}$. In phase 2, we train $\mathcal{N}_p$ starting from $\boldsymbol{\Theta}_{\text{adv}}$ on $(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ until zero training error.



(a) Easy Gaussians (b) Spirals $t \leq 10$ (c) Vectorized MNIST

Figure 1: **Test accuracy for different initializations.** We plot the test accuracy of $\mathcal{N}_p$ trained on linearly-separable Gaussians (a), multiple turns of the spirals ($t \in [0, 10]$)(b), and vectorized MNIST (c) under each of the experimental configurations described in Sec. 3.1, as a function of the network width p. The black curve represents the baseline of training from random initialization on the true labels (Experiment 1), the red curve represents following the phase 1 – phase 2 training proposed by Liu et al. (2020) (Experiment 2), and the blue curve represents completing phase 1, freezing the first layer weights, re-initializing the second layer weights to zero, and then completing phase 2 (Experiment 3). Error bars represent one standard deviation across different random initializations. In (a) and (b) we take $d = 2$, $n = 100$, while in (c) we have $d = 784, n = 128$. For the spirals, plotting mean and standard deviation shows intersecting error bars; however, in the Appendix we show plots of the median and interquartile range, which suggests that some of the error bar intersection actually results from outliers.

Experiment 3: In phase 1, we train $\mathcal{N}_p$ on $(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{rand}})$ until zero training error. In phase 2, we freeze the first layer weights (i.e., feature representations), reset the second layer weights to zero, and train *only the second layer* on $(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ until zero training error is achieved.

Note that any dataset where the test accuracy of Experiment 2 is worse than the test accuracy of Experiment 1 exhibits the phenomenon of interest.

3.2 Datasets

Mixture of Gaussians: This dataset consists of points of binary label $y \in \{0, 1\}$, drawn from a Gaussian with mean $(2y - 1) \cdot \mu$ and variance Σ. In (a), $\mu = [2.5, 2.5]$ and Σ is the identity matrix. The task is depicted in Fig. 2a.

Concentric Circles: This dataset consists of two concentric circles where the region inside the innermost circle is labeled $y = 0$ and the outside ring is labeled $y = 1$. More precisely, for each point, we pick a radius at random over $[0, 10]$, add or subtract 5 according to the class, then add some noise to it. We then randomly pick an angle at which to place it. The task is depicted in Fig. 2b.

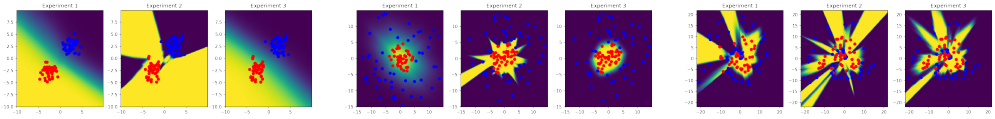
Spirals: This dataset consists of points in $\mathbb{R}^2$ of binary label $y \in \{0, 1\}$, where the first component is given by $(y + 1) \cdot t \cdot \cos(t)$, and the second component is given by $(y + 1) \cdot t \cdot \sin(t)$. Both components are each jittered by some uniform random noise scaled by $0.9 \cdot t$.

Vectorized MNIST: We take each of 128 MNIST images and flatten them from 28×28 to 784×1. Labels are the standard ones and cross entropy loss is used. For this setting, due to computational constraints, all experiments consider the same 128 samples, and each random label set is used for each width studied. This is substantially less randomness than in the other experiments, which may explain why the error bars are much smaller.

3.3 Results and Discussion

In this section, we comment the results from our numerical experiments. Figure 1 displays the performance as a function of the width for different tasks. In all cases, we observe statistically-significant gaps between the output from adversarial initialization (Experiment 2, red curve) and the baseline (Experiment 1, black curve). Below, we investigate the hypothesis in Sec. 2.

The role of representation quality — We investigate whether the main issue is the quality of representations induced by adversarial initialization. To this end, we first run Experiment 3, freezing the



(a) Linearly-separable Gaussians. (b) Concentric circles. (c) Spirals.

Figure 2: **The learned functions.** Decision boundary at the end of training in Experiments 1, 2, and 3 (from left to right) for linearly-separable Gaussians (a), concentric circles (b), and spirals (c). The plots are for $p = 500$ but reflect similar trends as other widths.

first layer to the adversarial solution, on a dataset of linearly-separable Gaussians. In the linearly-separable case (a), this performs as well as the baseline, and all three experiments achieve very close to perfect generalization[2]. While this certainly indicates that Hypothesis 2 is not satisfactory, it also suggests that this model is extremely simple. To test Hypothesis 2 in the more complex end, we studied vectorized MNIST, finding almost the opposite phenomenon: i.e., adversarial features perform much more comparably to fully adversarial initialization, with some variation depending on width (Fig. 1c). Thus, since it appears that it is possible in more complex settings for the adversarial initialization procedure to produce representations that do not suffice to separate the data as well as learned ones, the specific effect cannot be isolated just from the simple model. Viewing Figure 1a and Figure 1c, then, suggests a nontrivial interplay of adversarial initialization with data structure (Hypothesis 3).

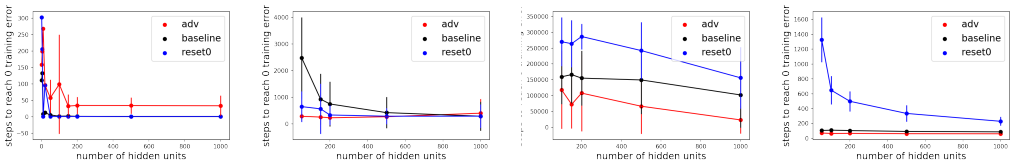
The role of data structure — To that end, we design and study a dataset which requires a non-linear separator (unlike simple mixture of Gaussians) but can still be expressed analytically (unlike MNIST), that we refer to as *spirals* dataset. The results from the spirals dataset (Figure 1b) indicate it might even *help* to maintain the features from this initialization procedure, cf Maennel et al. (2020). This result suggests that memorization could help in certain tasks. More subtly, in vectorized MNIST, for small widths, we see that Experiment 3 outperforms Experiment 2, indicating that when there are only a few features, they are not inherently bad, but for larger widths, more of the problematic performance could be related to bad features (since the features do no better than the fully adversarial case). Put together, these results lead us to conclude that the role of data structure and representation quality are tightly related.

Note that sources of randomness in each experiment of easy Gaussians and spirals are: (1) the draw of the data, (2) the draw of the random labels used in the adversarial initialization, and (3) the random weights that the network is initialized with. If we fix the data and only vary the other two factors (see appendix for examples), the trends look different to the averaged trend over different draws of data. This implies that even for a given distribution, the different loss landscapes induced by differing draws seem to be traversed differently during adversarial initialization. We will study this further in an attempt to characterize which factors arising from the weight distribution and which ones arising from the specific realization of the task affect the adversarial initialization performance.

Impact on decision boundary — Plotting the learned functions for the three experiments (Fig. 2) shows that adversarial initialization leads to jagged decision boundaries, while resetting the readout weights smoothens them. This may indicate that more of the piece-wise linear components are active with fully adversarial initialization (Experiment 2) as compared to the setting where we reset the readout weights (Experiment 3).

Impact on training time — An interesting question is how the three training schemes affect optimization speed (Fig. 2). For linearly-separable Gaussians, the training time following adversarial initialization (i.e., the number of steps required to reach zero training error) is larger than for random initialization or resetting at 0, for sufficiently large number of hidden units. On the other hand, for concentric circles, spirals, and vectorized MNIST, the training time from adversarial initialization is smaller, suggesting that the landscape near the adversarial initialization provides many easy-to-find minimizers of the training error. In the spirals and MNIST cases, the adversarial features with 0 readout setting takes much longer to achieve a solution, suggesting there needs to be substantial drift

[2]The concentric circles dataset also shows similar behavior, albeit less starkly. We defer the plot to the appendix.



(a) Easy Gaussians (b) Concentric circles (c) Spirals up to 10 (d) Vectorized MNIST

Figure 3: **The training time.** Number of steps to converge to zero training error for different datasets as a function of width. Different colors represent different initialization/training schemes. For the simplest task (a), the adversarial initialization takes longer to converge, whereas in the more complex ones (c-d), it converges relatively quickly. Dataset (b) behaves in a more inconclusive way.

in the second layer to find a solution.[3] This suggests that spirals may be a good model for further inquiry into what differentiates between overly simple models and theoretically-intractable ones.

Conclusions Hypothesis 1 does not suffice. Hypothesis 2 holds in certain regimes but importantly does not hold in many regimes. Hypothesis 3 bears further investigation; next steps in the representation quality line of work will need to consider the relationship between the quality of the representations and the necessary complexity of representations for a given data distribution.

4 Pathways for Future Investigation

While our current suite of experiments and datasets presents an intriguing picture of the phenomenon we are studying, there remain many open questions. In the following, we list a series of theoretical and computational investigations that we are currently exploring to shed light into our findings.

Assess the "badness" of the solution — We can evaluate more systematically the properties of the minima found from adversarial initialization using well-understood proxies for the performance, such as flatness and the quality of the decision boundary (see, e.g. Keskar et al. (2016); Wu et al. (2017); Guan and Loew (2020); Fawzi et al. (2018)).

Characterize the adversarial initialization —A related promising direction is to exploit theoretical results from the literature on implicit bias, as in Soudry et al. (2018). Analyzing the effect of the implicit bias for training on random labels starting from random initialization for one-hidden-layer networks Chizat and Bach (2020) might allow us to characterize the predictor achieved in phase 1 of Experiment 2, which in turn would inform us about the outcome of phase 2.

Characterize adversarial representations — We propose to study a random features model as done by Mei et al. (2022), in a teacher-student framework where labels are generated by a set of "teacher" features that establish a benchmark for "good" representations. We could then analyze how random features compare to adversarial ones. Adversarial features could be modeled as orthogonal to the teacher ones, or even estimated from implicit bias as mentioned in the above section. Furthermore, we could tune the the teacher features to model different task structures.

Characterize the solution space — Finally, we can attempt to quantify the volume of solutions found from adversarial initialization using the replica method from statistical physics as done, e.g., in Annesi et al. (2023); Baldassi et al. (2023), or enumerating the number of minima using the Kac-Rice method (see, e.g., Maillard et al. (2020)). Understanding the structure of the space of solutions and their generalization properties could shed light into the behaviour of the convergence time in relation to the performance.

Acknowledgments and Disclosure of Funding

This work had its genesis in open problem sessions during the 2022 Les Houches summer school in Statistical Physics and Machine Learning, organized by Lenka Zdeborová and Florent Krzakala. We

[3]These experiments suggest that a higher learning rate would be better for the spirals dataset, but we only report the smaller one since it is well-known that large step sizes fail to enter narrow minima and wanted to be fair in this evaluation.

are thankful to them for this opportunity. We also thank Damien Barbier and Kamesh Krishnamurthy for discussions in the early phases of this project.

This work was supported in part by the National Science Foundation under grants CCF-1815011 and CCF-2212968, by the NSF-Simons Funded Collaboration on the Mathematics of Deep Learning, and by the Defense Advanced Research Projects Agency under cooperative agreement HR00112020003. The views expressed in this work do not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred. FM was supported by a grant from the Simons Foundation (ID: 1141576).

References

Annesi, B. L., Lauditi, C., Lucibello, C., Malatesta, E. M., Perugini, G., Pittorino, F., and Saglietti, L. (2023). The star-shaped space of solutions of the spherical negative perceptron. *arXiv preprint arXiv:2305.10623*.

Baldassi, C., Malatesta, E. M., Perugini, G., and Zecchina, R. (2023). Typical and atypical solutions in non-convex neural networks with discrete and continuous weights. *arXiv preprint arXiv:2304.13871*.

Chiang, P.-y., Ni, R., Miller, D. Y., Bansal, A., Geiping, J., Goldblum, M., and Goldstein, T. (2022). Loss landscapes are all you need: Neural network generalization can be explained without the implicit bias of gradient descent. In *The Eleventh International Conference on Learning Representations*.

Chizat, L. and Bach, F. (2018). On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31.

Chizat, L. and Bach, F. (2020). Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In Abernethy, J. and Agarwal, S., editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1305–1338. PMLR.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR.

Fawzi, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Soatto, S. (2018). Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770.

Guan, S. and Loew, M. (2020). Analysis of generalizability of deep neural networks based on the complexity of decision boundary. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 101–106. IEEE.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. (2018). Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR.

Huang, W. R., Emam, Z., Goldblum, M., Fowl, L., Terry, J. K., Huang, F., and Goldstein, T. (2020). Understanding generalization through visualizations.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*.

Liu, S., Papailiopoulos, D., and Achlioptas, D. (2020). Bad global minima exist and sgd can reach them. *Advances in Neural Information Processing Systems*, 33:8543–8552.

Maennel, H., Alabdulmohsin, I. M., Tolstikhin, I. O., Baldock, R., Bousquet, O., Gelly, S., and Keysers, D. (2020). What do neural networks learn when trained with random labels? *Advances in Neural Information Processing Systems*, 33:19693–19704.

Maillard, A., Arous, G. B., and Biroli, G. (2020). Landscape complexity for the empirical risk of generalized linear models. In *Mathematical and Scientific Machine Learning*, pages 287–327. PMLR.

Mei, S., Misiakiewicz, T., and Montanari, A. (2022). Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84.

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

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. (2018). The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878.

Wu, L., Zhu, Z., et al. (2017). Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

A Some Additional Plots

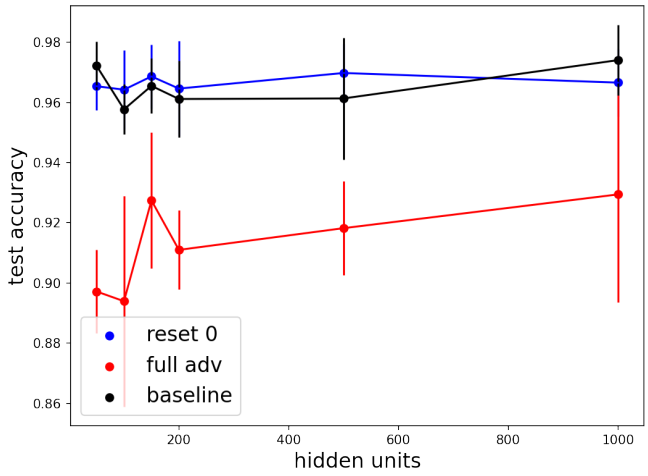


Figure 4: Concentric Circles shows similar phenomenon to easy Gaussians (Fig.1a).

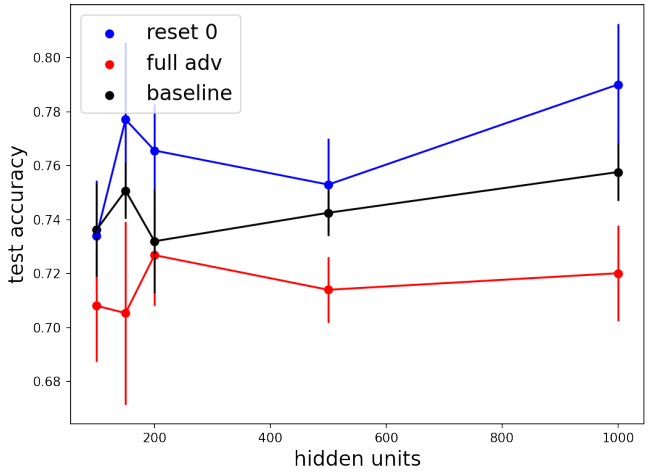
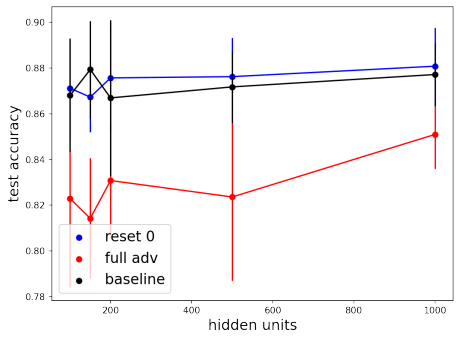
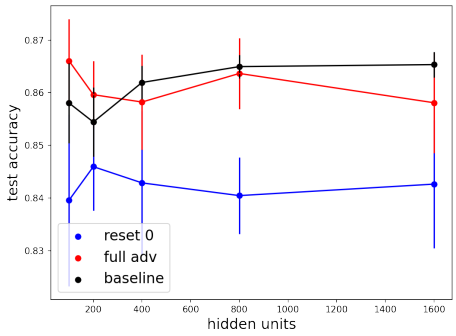


Figure 5: Spirals where $t \leq 10$. Plot shows median and Inter Quartile Range.



(a) Spirals where $t \leq 2\pi$ and averaged over many X, y_{train}



(b) Spirals where $t \leq 2\pi$ and the same X, y_{train} are used for all experiments. (Additionally, each y_{rand} is studied for each of the widths.)

Figure 6: Note that the trends are different between the two plots, one that is averaged over different draws of X, y_{train} , and the one where X, y_{train} is fixed. Thus, different draws have different properties that need to be studied.