Perspective: New Insights From Loss Function Landscapes of Neural Networks

Sathya R. Chitturi[1], Philipp C. Verpoort[2], Alpha A. Lee[2], and David J. Wales[1]

1 Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, UK and

2 Department of Physics, University of Cambridge, Cambridge CB3 0HE, UK

We investigate the structure of the loss function landscape for neural networks subject to dataset mislabelling, increased training set diversity, and reduced node connectivity, using various techniques developed for energy landscape exploration. The benchmarking models are classification problems for atomic geometry optimisation and hand-written digit prediction. We consider the effect of varying the size of the atomic configuration space used to generate initial geometries and find that the number of stationary points increases rapidly with the size of the training configuration space. We introduce a measure of node locality to limit network connectivity and perturb permutational weight symmetry, and examine how this parameter affects the resulting landscapes. We find that highly-reduced systems have low capacity and exhibit landscapes with very few minima. On the other hand, small amounts of reduced-connectivity can enhance network expressibility and can yield more complex landscapes. Investigating the effect of deliberate classification errors in the training data, we find that the variance in testing AUC, computed over a sample of minima, grows significantly with the training error, providing new insight into the role of the variance-bias trade-off when training under noise. Finally, we illustrate how the number of local minima for networks with two and three hidden layers, but a comparable number of variable edge weights, increases significantly with the number of layers, and as the number of training data decreases. This work helps shed further light on neural network loss landscapes and provides guidance for future work on neural network training and optimisation.

I. INTRODUCTION

In this report we analyse the structure of the loss function landscape for neural networks. Here, the landscape refers to the loss as a function of the trainable parameters (node weights), and we exploit computational tools developed for exploration of energy landscapes in molecular science.[1] The principal focus is on the organisation of local minima of the loss function, which correspond to the isomers of a molecule. This organisation is defined by the pathways between local minima mediated by transition states, which are stationary points of Hessian index one, with precisely one negative Hessian eigenvalue.[2] The connection between a molecular energy landscape (EL) and a loss function landscape has been developed in previous work, as summarised below. We have previously referred to the loss function landscape (LFL) as a machine learning landscape (MLL), and we will employ these descriptions interchangeably in the present contribution.

Unfortunately, direct analysis of the loss landscape is challenging due to issues of computational complexity.[3] The high dimensionality employed in deep learning representations produces poorly conditioned problems for optimisation, and leads to slow convergence. Furthermore, the number of stationary points grows exponentially with the dimensionality of the problem,[4] as in molecular science.[5,6] Nevertheless, the power and utility of recent machine learning techniques is remarkable, and part of the motivation for the present work is to understand these advances in terms of the underlying loss function landscape.

Choromanska et al. have previously considered the performance of various local minima for neural networks.[7] A good performance, by their metric, corresponds to high accuracy for both an independent training and test set. They show that theoretically, subject to a number of assumptions of independence, neural network optimisation reduces to minimising the energy of the spin-glass Hamiltonian from statistical physics.[7] Based on the spin-glass model, bounds can be derived, suggesting that there exists a tight band of local minima, bounded above the global minimum, characterized by low training and testing errors. Furthermore, in this model it is exponentially less likely to find a minimum with relatively high testing error as the dimensionality of the neural network grows.[7] These results suggest that almost any local minimum that is found via standard optimisation techniques should perform comparably to any other local minimum on an unseen test set.[7]

Wu et al. agree with the conclusion that the majority of local minima solutions of the loss

landscape tend to have properties similar to that of the global minimum.[8] This work suggests that neural networks may generalize well because they yield simple solutions (minima) with small Hessian norm. A theoretical analysis of two-layer networks indicates that these simple solutions occur because the volumes of the basins of attraction for minima with high test error are exponentially dominated by the volumes of the basins of attraction for minima with low test errors. In other words, good solutions lie in large, flat regions of parameter space and bad solutions lie in small, sharp regions.[8]

Li et al. proposed a filter-wise normalisation scheme to preserve scale invariant properties of neural networks, which allows for comparison between different architectures and landscapes.[9] Low-dimensional 2D contour plots were created to investigate the loss function along random directions near chosen minima. By studying a variety of different architectures on the CIFAR-10 dataset, Li et al. suggest that flat minima tend to generalize better than sharp minima. Furthermore, shallow, wide neural networks have contour surfaces with a convex appearance, which might make them more generalizable. Nguyen et al. agree with this description, and showed that if a network has a pyramid-like structure following a very wide layer, then local minima are very close the global minimum and the surface is much easier to navigate.[10]

Some of the assumptions made in the above theoretical models are quite restrictive, and may not hold for examples of practical interest. Furthermore, low-dimensional representations of the landscape can misrepresent the underlying non-convexity present in higher dimensions. In addition, it is possible to manipulate the dataset and optimisation problem to create solutions with very high training accuracies but with arbitrarily low testing accuracies. This scenario can be achieved by adding a tunable attacking term to the cost function and deliberately misassigning labels during training.[8] Furthermore, it is possible to create datasets in which specific initialisation schemes will either not converge or converge to high-lying loss solutions.[11]

To avoid the problems of low-dimensional projection and restrictive theoretical assumptions, the present work builds on previous considerations of the LFL as an energy landscape.[12–15] Energy landscapes (EL) in molecular science,[12,13,16–18] are defined in terms of the potential energy (PE), with minima corresponding to physically stable structures which can interconvert via transition states. Minima are defined geometrically, as stationary points with non-negative Hessian eigenvalues. Transition states are defined as stationary

points with exactly one negative Hessian eigenvalue (index one saddle points);[2] the Murrell-Laidler theorem guarantees that the pathway with the lowest barrier between two minima involves only transition states, and not higher index saddles.[1,2] By investigating the correspondence between a potential energy surface (PES) and the neural network loss function, where the atomic configuration space becomes the neural network parameter space, many of the tools developed in EL research can be used to study neural network landscapes.[12,13]

We have recently compared the landscapes for neural networks with one, two and three hidden layers for a similar number of fitting parameters.[18] In principle, a single hidden layer with enough nodes is sufficient to fit a well-behaved function,[19] although the required number of hidden nodes scales exponentially with the number of parameters.[20] In the present contribution we report new results for the properties of such networks to investigate the structure of the underlying LFL. In particular, we consider the effect of systematically removing certain edges from the network to reduce the connectivity, and the effects of training set mislabelling. In addition we present some results for neural networks with multiple hidden layers for comparison.

Our goal in this research is to understand the behavior of relatively small neural networks, where the underlying solution landscape can be properly characterised. We hope that the resulting insight will carry over to large networks, where there may be too many parameters to locate even a single local minimum.

II. DEFINING THE NETWORK

We begin with a standard single hidden-layer neural network architecture[21] containing input, output, and hidden nodes, plus a bias added to the sum of edge weights used as input to the activation function for each hidden node, w_j^{bh}, and each output node, w_i^{bo}. For the classification problem described in §IV the inputs correspond to interatomic distances for starting point geometries in a triatomic cluster, and there are $N_{\mathrm{out}} = 4$ possible outputs, corresponding to the four local minima that the cluster can adopt, as in previous work.[13,14,22] Each training or test data item α comprises N_{in} inputs written as $\mathbf{x}^\alpha = \{x_1^\alpha, \ldots, x_{N_{\mathrm{in}}}^\alpha\}$, and a set of N_{data} input data is written as $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^{N_{\mathrm{data}}}\}$.

The outputs, y_i, were calculated as

$$y_i(\mathbf{W};\mathbf{x}^\alpha) = w_i^{\mathrm{bo}} + \sum_{j=1}^{N_{\mathrm{hidden}}} w_{ij}^{(1)} \tanh\left[w_j^{\mathrm{bh}} + \sum_{k=1}^{N_{\mathrm{in}}} w_{jk}^{(2)} x_k^\alpha\right], \tag{1}$$

for a given input data item $\mathbf{x}^\alpha$, and weights $w_{ij}^{(1)}$ between hidden node j and output i, $w_{jk}^{(2)}$ between input k and hidden node j, and bias weights w_j^{bh} and w_i^{bo} collected into the vector $\mathbf{W}$. Softmax probabilities, $p_c(\mathbf{W};\mathbf{x}^\alpha)$ were obtained from the outputs to reduce the effect of outliers

$$p_c(\mathbf{W};\mathbf{x}^\alpha) = e^{y_c(\mathbf{W};\mathbf{x}^\alpha)} \Big/ \left(\sum_i^{N_{\mathrm{out}}} e^{y_i(\mathbf{W};\mathbf{x}^\alpha)}\right). \tag{2}$$

The loss function, which defines local minima and transition states of the machine learning landscape, was written as the sum of a cross-entropy, and an **L2** regularisation term with coefficient $\lambda > 0$:

$$E(\mathbf{W};\mathbf{X}) = -\frac{1}{N_{\mathrm{data}}} \sum_{\alpha=1}^{N_{\mathrm{data}}} \ln p_{c(\alpha)}(\mathbf{W};\mathbf{x}^\alpha) + \lambda \mathbf{W}^2, \tag{3}$$

where $c(\alpha)$ is the known outcome for input data item α in the training set. The regularisation term biases against large values for the weights and shifts any zero eigenvalues of the Hessian (second derivative) matrix, which would otherwise complicate transition state searches.[15,23] To accelerate computation of the potential, a GPU version[24] of the loss function and gradient was also implemented and is available in the public domain `GMIN` and `OPTIM` programs.[25–27]

III. CHARACTERISATION OF THE LOSS FUNCTION LANDSCAPE

To train each network we minimise the loss function, $E(\mathbf{W};\mathbf{X})$, with respect to the variables $w_{ij}^{(1)}$, $w_{jk}^{(2)}$, w_j^{bh} and w_i^{bo}, written collectively as a vector of weights $\mathbf{W}$. Basin-hopping global optimisation was used[28–30] to search for the global minimum, and all the distinct minima obtained during these searches were saved for later comparison. In this approach we take steps between local minima of the loss function, accepting or rejecting moves according to a simple Metropolis criterion[31] based upon the change in loss function, scaled by a parameter that plays the role of temperature. Downhill moves are always accepted, and the probability of accepting an uphill move depends on the fictitious temperature.[28–30] For the machine learning landscapes considered in the present work locating the global minimum is usually straightforward, and the choice of basin-hopping parameters is not critical. A customised LBFGS optimisation routine was employed for local minimisation, based on the limited memory version[32,33] of the quasi-Newton Broyden,[34] Fletcher,[35] Goldfarb,[36] Shanno,[37] BFGS procedure.

Transition state candidates were determined using the doubly-nudged[38,39] elastic band[40,41] (DNEB) approach, which involves optimising a series of intermediate atomic configurations (images) connected by a harmonic potential. The transition state candidates were then refined using hybrid eigenvector-following,[42–44] which involves systematic energy maximisation along just one Hessian eigenvector. Having determined a candidate transition state, the connected minima are located by minimisation following small displacements along the eigenvector corresponding to the unique negative eigenvalue. This method can be employed to create databases of connected local minima,[45] which are analogous to kinetic transition networks.[46–49] Visualization of the landscape was performed using disconnectivity graphs.[50–52] This approach segregates the energy landscape into disjoint sets of minima that can interconvert within themselves below each energy threshold. Using this topological method, an undirected tree is constructed.[53] For the machine learning analysis, the vertical axis represents the neural network training loss. The branches of the graph correspond to the minima of the loss function. More specifically, each branch represents the vector of parameters containing the node-connectivity weights for the neural network, and terminates at a height on the vertical axis corresponding to the training loss function. The branches join together at regularly spaced intervals on the vertical axis when they can interconvert via

pathways mediated by index one saddle points (transition states). At the highest threshold all the minima lie in the same group because there are no infinite barriers on the landscape, and only one vertical branch remains in the graph.

Analytic first and second derivatives were programmed for $E(\mathbf{W};\mathbf{X})$ in the public domain `GMIN` and `OPTIM` codes for exploration of the corresponding loss function landscapes.[25–27] Further details are provided elsewhere, including a review of the energy landscapes perspective in the context of machine learning.[12] Performance of the neural networks was measured using standard area under curve (AUC) metrics. The AUC metric ranges from 0 to 1, with an AUC = 0.5 signifying random performance. If the AUC value is < 0.5, the model performs worse than than a random guess and if the AUC is > 0.5, it performs better. The AUC is calculated by determining the true positive and false positive statistics for the machine learning problem as a function of the threshold probability, P, for predicting convergence to one of the outcomes. Details are provided in the Supplementary Information (SI).

For the single-layered architectures, we use a short-hand [A,B,C,D,E], to refer to the number of inputs, hidden nodes, outputs, training data and regularization constant, respectively. For example, in the geometry optimisation classification problem the [2,10,4,1000,0.0001] architecture corresponds to 74 optimisable parameters (two input bond lengths, 10 hidden nodes and 4 output classes with 1000 training points and a regularization constant of 0.0001), and for the MNIST dataset the [784,10,10,1000,0.1] architecture corresponds to 7960 optimisable parameters.

IV. APPLICATION TO PREDICTION OF GEOMETRY OPTIMISATION OUTCOMES FOR AN ATOMIC CLUSTER

The first classification problem that we consider involves predicting the outcome of local minimisation for a triatomic cluster, as in previous reports.[13,14,22] Here we emphasise that we are not using machine learning to perform the optimisation,[54–56] but instead to predict the outcome from a given starting configuration.

The potential energy surface for the cluster is defined by a two-body Lennard-Jones[57] potential and a three-body Axilrod–Teller[58] term, weighted by a coefficient Z. The total

potential energy for this $LJAT_3$ cluster for particle positions $\mathbf{r}_i$, and separations r_{ij} is then

$$V = 4\varepsilon \sum_{i<j} \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] + Z \sum_{i<j<k} \left[\frac{1 + 3\cos\theta_1 \cos\theta_2 \cos\theta_3}{(r_{ij} r_{ik} r_{jk})^3} \right], \tag{4}$$

where the internal angles of the triangle defined by atoms i, j and k are θ_1, θ_2 and θ_3. We choose $Z = 2$, for which the molecular potential energy surface has an equilateral triangle minimum and three linear minima with each of the three atoms in the centre position. In all the AUC calculations for this problem we chose to refer the threshold probability P to the outcome corresponding to the equilateral triangle minimum.

This cluster, denoted $LJAT_3$, defines a multinomial logistic regression problem. Local minimisation for any starting configuration will terminate in one of the four minima, and we seek to predict this outcome given inputs corresponding to the initial geometry. The configuration is uniquely specified by the three interatomic distances r_{12}, r_{13}, and r_{23}, and given sufficient training data in this three-dimensional space a large enough neural network can make accurate predictions by learning the basins of attraction[1,59] of the four minima. Here, however, we limit the inputs to two of the three distances, namely r_{12} and r_{13}. The basins of attraction of the equilateral triangle and the linear minimum with atom 1 in the middle overlap in the space defined by the missing coordinate, r_{23}. Hence the best predictions possible should correspond to networks that learn the marginal probabilities for the different outcomes in the lower dimensional space.

We note that this sort of classification problem is not only a convenient benchmark, but also has practical applications. Knowledge of the relative configuration volumes for the catchment basins of different minima can be used to calculate thermodynamic properties,[60] and predicting outcomes without running local minimisation to convergence would provide a way to save computational resources.[61]

Two databases (D1 and D2) of initial configurations and outcomes were considered, as in previous work.[18] Starting geometries were generated by randomly distributing the three atoms in a cube of side length L. The datasets involved 200,000 minimisations for cube lengths of $L = 2\sqrt{3}\,\sigma$ (D1) and $L = 1.385\,\sigma$ (D2). A third dataset, D3, with $L = 2\sqrt{2}\,\sigma$ (D3) was also created; results for this dataset are reported in the Supplementary Information (SI). In each case the data was divided into two halves for training and testing purposes. All the local minimisations, which define the outcome and classification label for each data item, consisting of the initial r_{12} and r_{13} values, were performed using the customised LBFGS

algorithm[32,33] described above. The convergence condition on the root mean square gradient was $10^{-10}\varepsilon/\sigma$.

A. Landscapes Subject to Dataset Mislabelling

Many real datasets of interest have significant label noise,[62,63] arising from difficulties in the data cleaning and acquisition processes, or simply from ambiguous class differentiation criteria. Additionally, to reduce acquisition loss, many practitioners prefer to obtain large amounts of low-quality data, rather than small amounts of high-quality data. While this scenario allows for the creation of a much larger labelled training set, it also has the potential to greatly deteriorate the quality of the dataset.[62–65] In light of the positive advantages of acquiring cheap data, much effort has been dedicated to improving the robustness of training neural networks under noise. Previously, it has been demonstrated that neural networks can perform well under uniform label noise,[62,64] even retaining predictive capability in regimes where the the ratio of noisy data to clean data exceeds 100 to 1. One possible explanation is that this phenomenon is a result of a filtering effect due to favourable gradient cancellation.[62] On the other hand, it is known that neural networks perform poorly for more sophisticated noise models, including both stochastic,[64] and adversarial type noise.[66]

Here we have analysed the uniform mislabelling case to see if the landscape approach can provide insight into how neural networks learn under noise. To study this problem, we permuted a fixed percentage of training outcomes for 1000 input data items. In this scenario, an outcome i would be mapped to any other outcome j with probability $\frac{1}{N-1}$, where N is the number of output classes. Specifically, for the D1 and D2 datasets (four outputs), class i could be mislabelled to that of any class $j \neq i$ with equal probabilities of $\frac{1}{3}$. Similarly, for the MNIST dataset,[67] each of the ten output classes has nine possible options for mislabelling with corresponding probabilities of $\frac{1}{9}$. It is important to note that the mislabelling procedure was applied only to the training data set in order to study relevant properties on a clean unseen testing set.

Previous work by Rolnick et al. used a fixed amount of correct training data rather than a total error percentage (fixed total training data). In our analysis, we opt for an error percentage formulation, as the number of stationary points decreases with the amount of training data,[18] complicating the interpretation of our disconnectivity graph analysis. Here,

we note that the distribution of outcomes varies with the size of the configuration space. For example, the more compact dataset (D2) contains a larger number of equilateral triangle minima (class 0). Since we cannot uncouple the outcome distribution from the choice of configuration space in an unbiased manner, we studied all the relevant properties using the size of the configuration space as an extra variable parameter.

For the D1 and D2 datasets, we were able to perform a (near) exhaustive search of the low-lying minima for landscapes with fixed error percentages of 0, 10, 50 and 100% for the [2,10,4,1000,0.0001] neural network architecture. Results for the D3 dataset are presented in the SI.

Training							**Testing**	
Error (%)	Min	Ts	Gmin AUC, Loss	$\overline{AUC}$, σ(AUC)	Incorrect $\overline{AUC}$, σ(AUC)	Correct $\overline{AUC}$, σ(AUC)	Gmin AUC, Loss	$\overline{AUC}$, σ(AUC)
0	122	592	0.749, 0.850	0.746, 0.0035	-	0.746, 0.0035	0.732, 0.891	0.733, 0.0025
10	266	960	0.727, 1.000	0.724, 0.0036	0.509, 0.015	0.747, 0.0034	0.720, 0.726	0.726, 0.0043
50	394	1474	0.639, 1.291	0.638, 0.0029	0.539, 0.0079	0.760, 0.0072	0.706, 1.131	0.699, 0.0083
100	490	1395	0.589, 1.321	0.591, 0.0061	0.591, 0.0061	-	0.336, 1.918	0.340, 0.013

TABLE I: Summary of results for the D1 dataset. Min and Ts refer to the number of minima and transition states, while Gmin refers to the minimum with the lowest loss value.

Training							**Testing**	
Error (%)	Min	Ts	Gmin AUC, Loss	$\overline{AUC}$, σ(AUC)	Incorrect $\overline{AUC}$, σ(AUC)	Correct $\overline{AUC}$, σ(AUC)	Gmin AUC, Loss	$\overline{AUC}$, σ(AUC)
0	6	20	0.810, 0.519	0.810, 0.00033	-	0.810, 0.00033	0.797, 0.552	0.796, 0.00031
10	13	66	0.730, 0.791	0.728, 0.0021	0.190, 0.019	0.809, 0.0023	0.791, 0.622	0.791, 0.0020
50	26	155	0.604, 1.285	0.602, 0.0018	0.398, 0.010	0.779, 0.0054	0.741, 0.994	0.739, 0.0030
100	20	148	0.772, 1.236	0.768, 0.0047	0.768, 0.0047	-	0.242, 2.771	0.245, 0.0043

TABLE II: Summary of results for the D2 dataset. Min and Ts refer to the number of minima and transition states, while Gmin refers to the minimum with the lowest loss value.

The number of minima and transition states (Min and Ts), as well as the loss associated

with the training global minimum (Gmin Loss), are shown in Tables I-II. All training set distributions are available in the SI.

We found that, on average, the number of local minima and transition states increased with the percentage of mislabelled data for both datasets (Tables I-II). This observation suggests that a larger number of local minima reflect many competing values for the parameters of the model, and thus produce higher uncertainty in the statistical fit. Based on this reasoning, it is unsurprising that noisier datasets lead to greater uncertainty in fitting the training data. The loss value of the global minimum also increased with the percentage of mislabelled data (Tables I-II).

In addition, we observed that the larger the molecular configuration space (D1 > D2), the greater the number of minima and transition states; this trend also holds for D3 (SI). This result is expected as there should be greater uncertainty in predicting final outcomes from more diffuse initial molecular configurations. In other words, the diversity of the dataset depends on the size of the configuration space. This interpretation is further supported by the observation that the loss of the global minimum increases with configuration space size.

To study generalization, we used the AUC value corresponding to the training global minimum (Gmin AUC) as a metric to characterise the performance of the neural network on the D1 and D2 datasets (Tables I-II). In both geometry optimisation datasets (D1 and D2), as the percentage of mislabelled data increased, the training and testing AUC for global training minimum decreased (Tables I-II). This trend is consistent with expectations, as randomising labels should increase the generalization error.[65] For 0% error, we observe relatively high AUC values for both training and testing; in particular, for the D1 and D2 datasets, the training AUCs outperform the corresponding testing AUCs, as expected. Interestingly, however, for 10% and 50% error, the testing AUCs outperform the training AUCs for the global training minimum (Tables I-II). This result implies that the neural network learns the structure of the correct data and filters out the noise.[62] Thus, since the training AUC is calculated on the mislabelled dataset, the neural networks perform poorly (since they have actually learned the correct structure). However, since the testing AUC is calculated on a correctly labelled dataset, the neural networks perform significantly better. Note that when the error rate is increased to 100%, the training error is relatively low,[65] as the network overfits to noise. However, the testing AUC decreases precipitously. This decrease is unsurprising as the neural network is fitted to noise, and thus cannot possibly

generalize to an unseen dataset.

In addition to studying the properties of the training global minimum, we also calculate the average ($\overline{AUC}$) and standard deviation (σ(AUC)) of the AUC values computed over all the training local minima in our database (Tables I-II). For the case of 0% error, we observe a tight band of low-lying local minima with high testing accuracies, which agrees with previous work.[7,8] For increasing error percentages, we also observe the same trends for the average AUC values as those obtained using the training global minima, suggesting a general filtering mechanism for single-layered perceptrons under uniform label noise (Tables I-II). We also find that the variance of the testing AUC increases significantly with the percentage of training error (Tables I-II).

To further analyse these effects, we investigated the performance of the network on the mislabelled and correctly labelled entries of the (mislabelled) training dataset (Tables I-II). For both datasets (D1 and D2), the training AUC values for the correctly labelled components exceeded the corresponding testing AUC values (Tables I-II). From these results, it is clear that, even at high training errors, the network can distinguish clean data from noisy data.

To study the structure of the loss function landscapes for single-layered perceptrons under uniform noise, we produced the corresponding disconnectivity graphs,[50,51] coloured by both training and testing AUC values, for the D1 and D2 datasets (Figures 1-4).

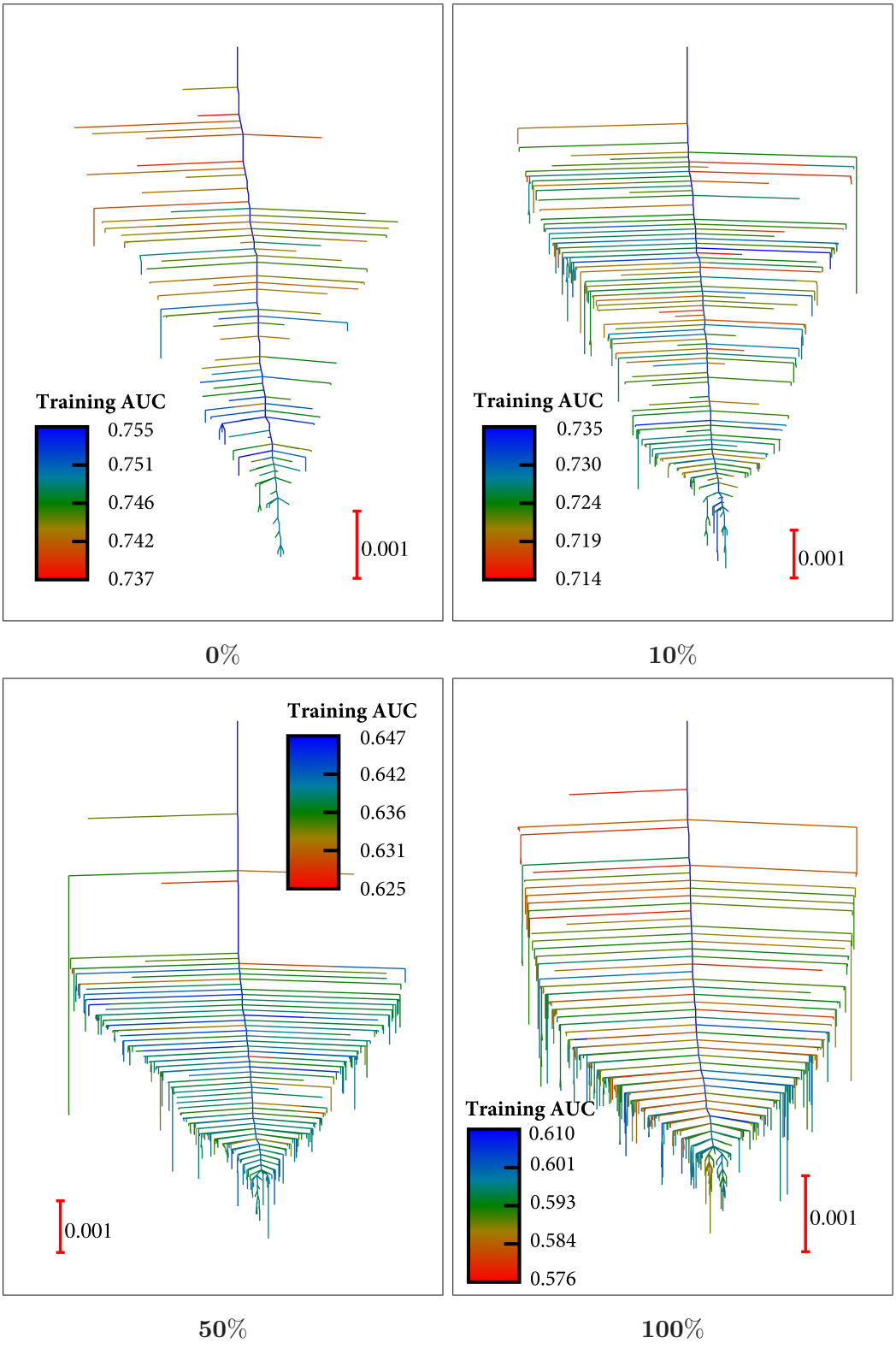
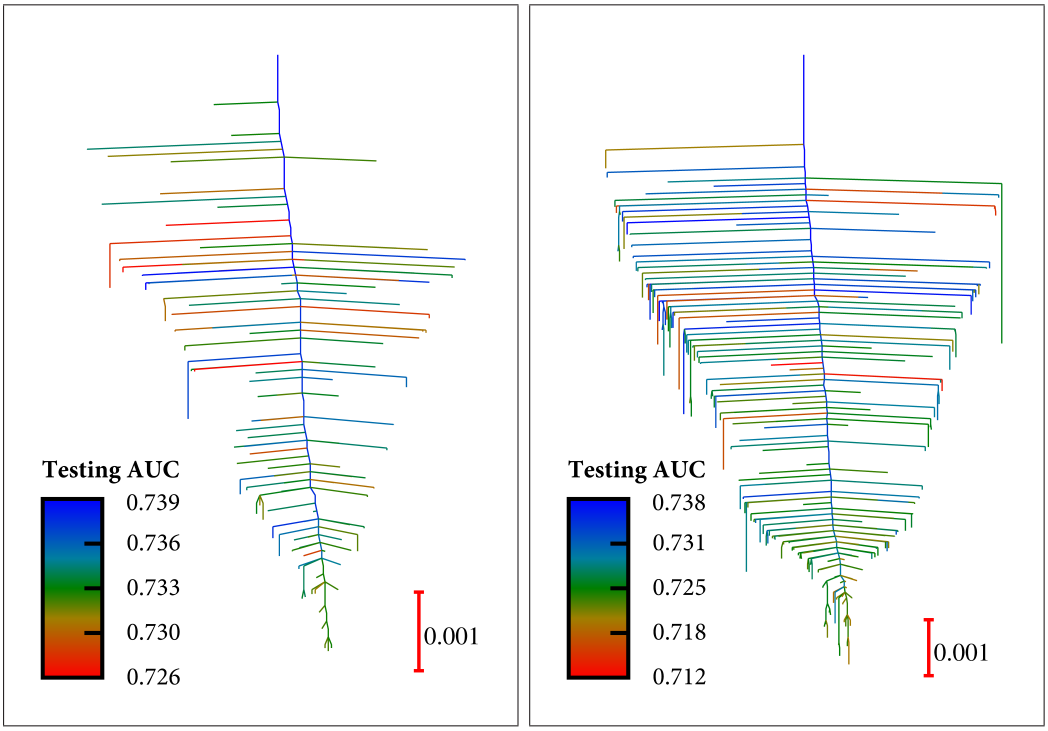
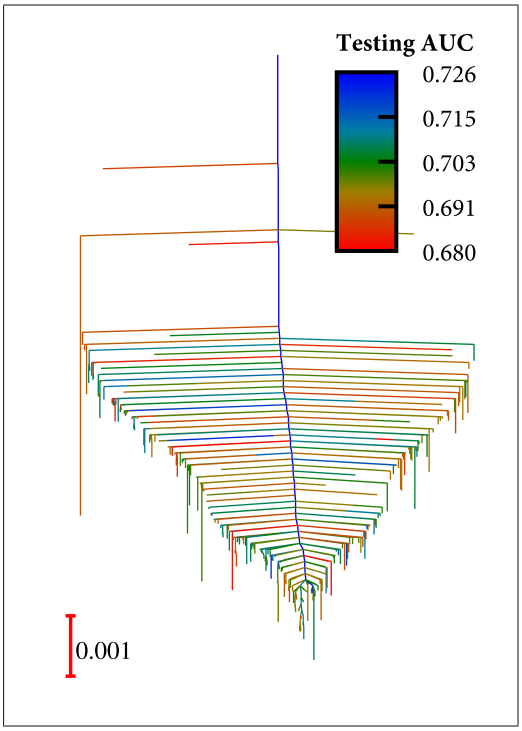


FIG. 1: Disconnectivity graphs for dataset D1, 1000 training points, $\lambda = 0.0001$, coloured by training AUC as a function of % label errors, as marked.

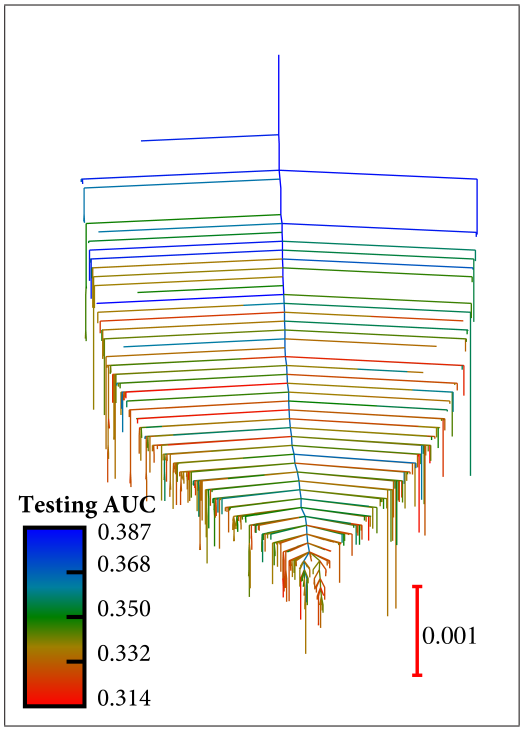


0%

10%



50%



100%

FIG. 2: Disconnectivity graphs for dataset D1, 1000 training points, $\lambda = 0.0001$, coloured by testing AUC as a function of % label errors, as marked.

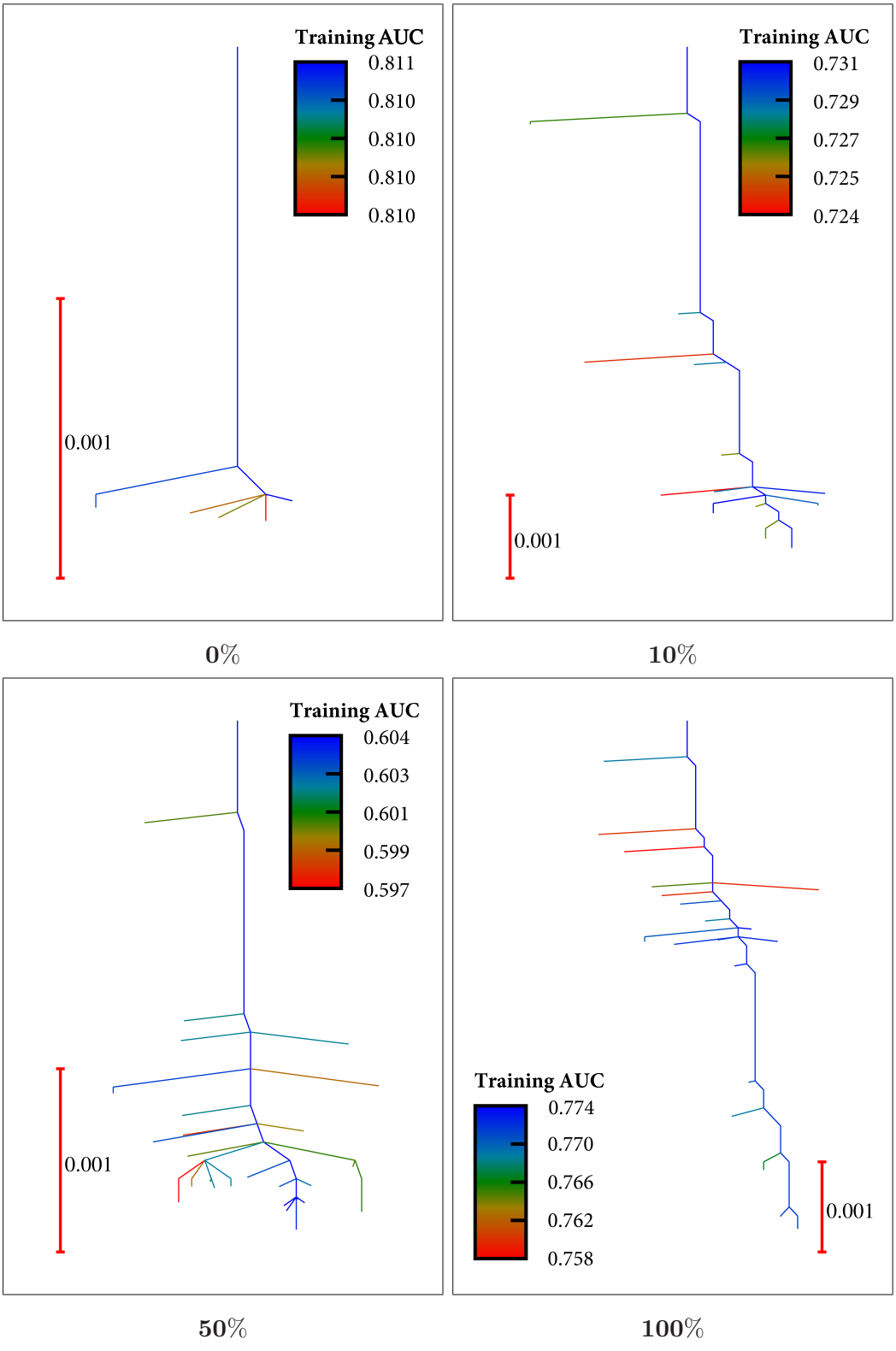


FIG. 3: Disconnectivity graphs for dataset D2, 1000 training points, $\lambda = 0.0001$, coloured by training AUC as a function of % label errors, as marked.

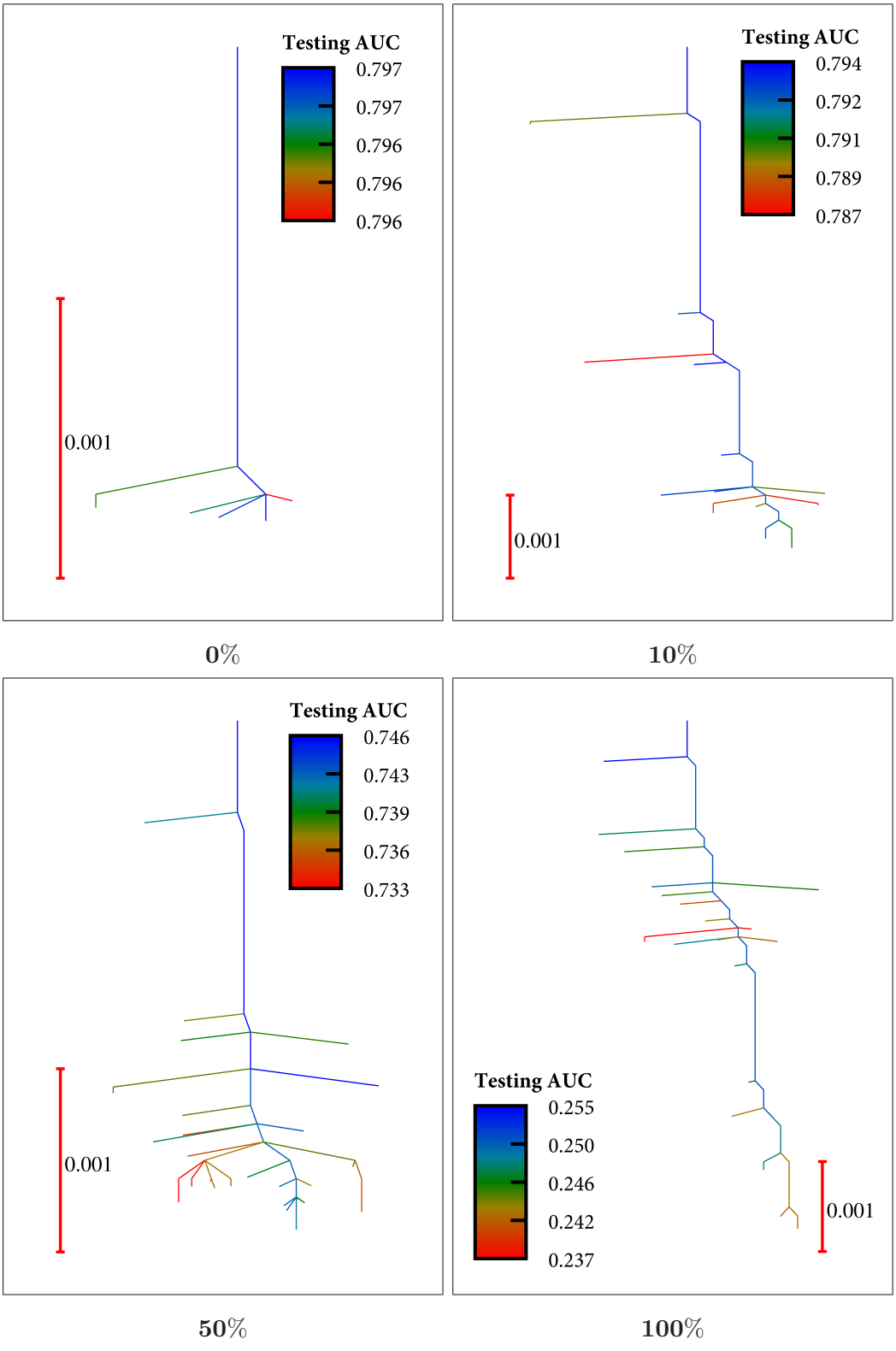


FIG. 4: Disconnectivity graphs for dataset D2, 1000 training points, $\lambda = 0.0001$, coloured by testing AUC as a function of % label errors, as marked.

Interestingly, single-funnelled energy landscapes are observed in each case. Since even the graphs at 100% error have a funnelled appearance, the structure likely arises due to the single-layered feed-forward architecture, not the input data. These results are consistent with previous work on the appearance of single-layered neural network landscapes,[8,12,13] as well as previous suggestions that noisy landscapes are no harder to train than clean landscapes.[65]

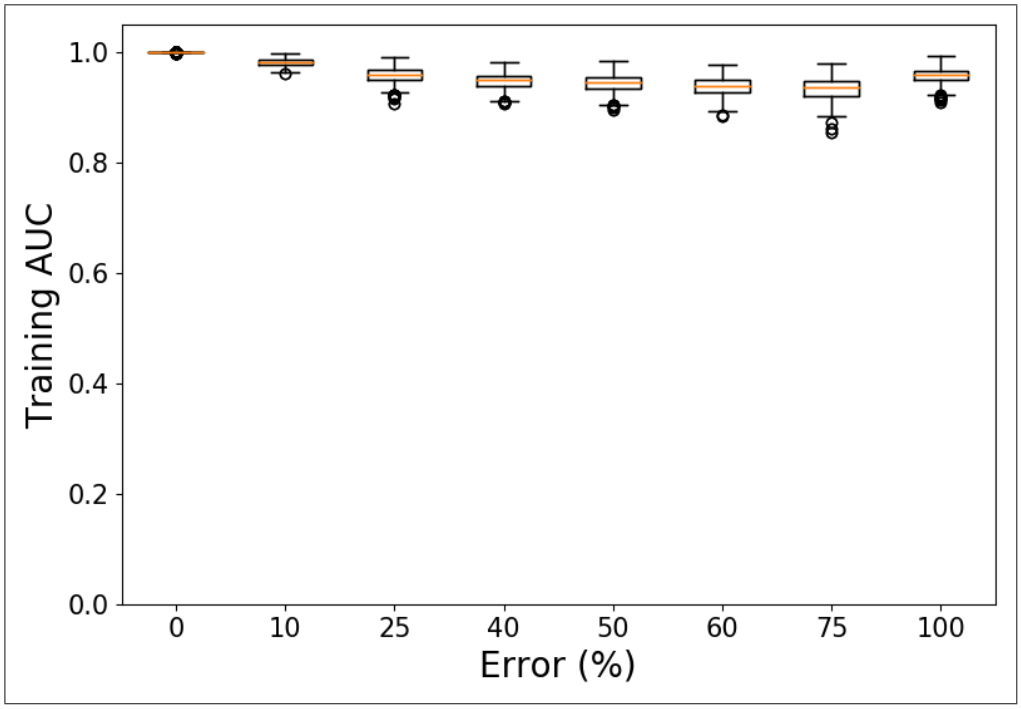
As expected, for all error thresholds, low-lying minima correspond to high training AUCs. Furthermore, the training and testing AUC values are reasonably correlated for 0% error. This result is also unsurprising, as the premise of neural network training is that low-lying minima generalize well to unseen training sets. Interestingly, as the mislabelling percentage increases, the better testing AUC minima (in the graphs, green-blue) are found at higher loss values, and the low-lying minima can have relatively low testing AUC values. This result highlights the bias-variance trade-off between over-fitting and generalization. Some low loss minima overfit to noise, leading to high training AUCs and low testing AUCs. However, some high loss training can filter the noise more effectively and thus generalise well (i.e. higher AUC values for testing). These results are consistent with the hypothesis that it can sometimes be better to converge to local minima, rather than the global minimum, to prevent overfitting.[7] Together, these results help explain why the testing variance for the AUC increases with the percentage of mislabelled training data.

For MNIST data, a similar pictures emerges after mislabelling various fixed percentages of the training data. Since the architecture used here, [784,10,10,1000,0.1], has nearly 8000 optimisable parameters, our results are based on samples of low-lying minima (i.e. not exhaustive searching). These calculations were much more computationally expensive, and we used a GPU accelerated implementation for basin-hopping global optimisation.[24] Unlike the D1 and D2 datasets, we do not obtain higher testing accuracies relative to training accuracies as the error threshold is increased (Table III). However, analysis of neural network performance, average over database minima on the the correct and incorrect portions of the mislabelled dataset, shows that the networks still perform significantly better on the clean segment of the training data, even with large amounts of noise (Table III).

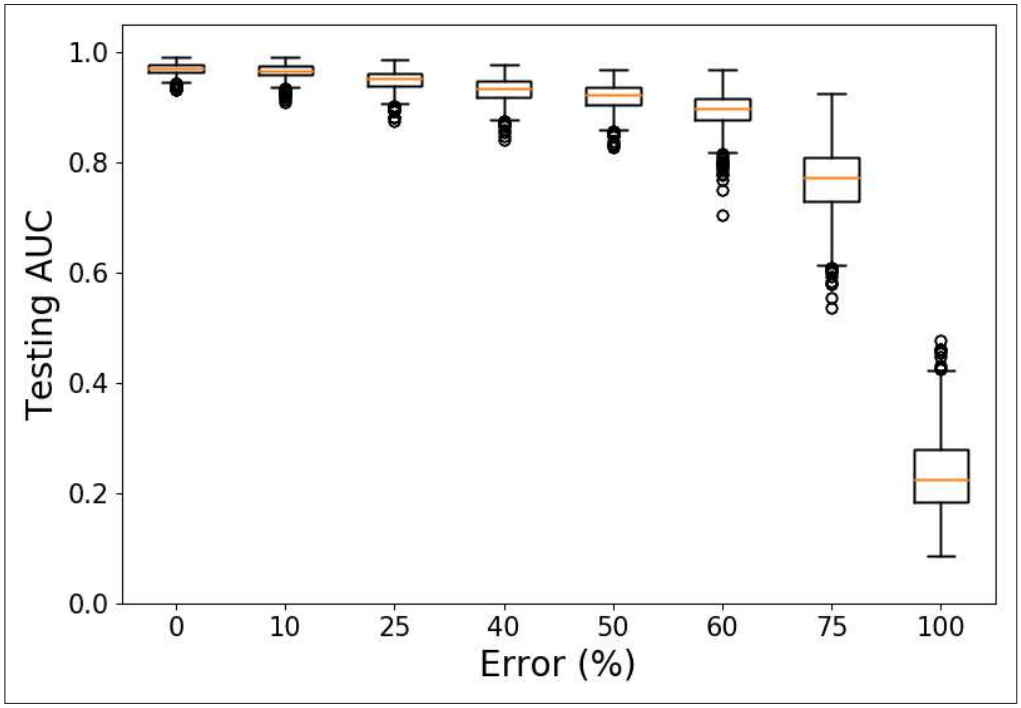
Training				**Testing**
Error	$\overline{AUC}$, σ(AUC)	Incorrect $\overline{AUC}$, σ(AUC)	Correct $\overline{AUC}$, σ(AUC)	$\overline{AUC}$, σ(AUC)
0	0.9996, 0.0027	-	0.9996, 0.0027	0.9687, 0.010
10	0.9783, 0.0070	0.7747, 0.050	0.9997, 0.0014	0.9645, 0.012
25	0.9545, 0.013	0.8011, 0.044	0.9991, 0.0025	0.9472, 0.018
40	0.9429, 0.015	0.8440, 0.032	0.9976, 0.0042	0.9304, 0.022
50	0.9390, 0.016	0.8707, 0.029	0.9950, 0.0062	0.9197, 0.024
60	0.9310, 0.017	0.8893, 0.026	0.9891, 0.010	0.8940, 0.031
75	0.9281, 0.020	0.9164, 0.024	0.9729, 0.019	0.7716, 0.061
100	0.9509, 0.015	0.9509, 0.015	-	0.2333, 0.072

TABLE III: Summary statistics for MNIST dataset.

It is again worth highlighting that, similar to the D1 and D2 case, there is a systematic trend towards increased testing AUC variance with the increase in dataset error, which indicates a change in the structure of the underlying landscape. Thus, while we do find good minima,[62,64] we also find many bad minima (Figure 5).



(a) **Training AUC for various percentages of mislabelled data.**



(b) **Testing AUC for various percentages of mislabelled data.**

FIG. 5: Box-plots for training (a) and testing (b) AUC values for various error percentages on the MNIST dataset. The box extends from the first (Q_1) to third (Q_3) quartiles (25th to 75th percentiles, range $Q_3 - Q_1 = IQR$, the interquartile range) with a band at the median.

Based on these numerical results it appears that the relatively tight band of local minima above the global minimum[7,8] no longer exists for the mislabelled case. Furthermore, in almost every example, the variance of the average testing AUC is greater than the variance of the average training AUC. Thus, while it is possible to obtain high testing accuracies under uniform random error,[62,64] the landscape perspective indicates that the probability of finding such solutions diminishes as the error percentage increases. These results indicate that it might be valuable to further analyze the properties of the subset of minima that perform well under high training error. This approach might be particularly helpful in designing new optimisers to preferentially find good solutions when training under noise.

V. LANDSCAPES WITH REDUCED CONNECTIVITY

To investigate the effect of reduced connectivity between the layers of a network we defined locality via a simple distance metric. The nodes in each of the three layers were mapped onto a unit line at positions 0, $1/(N_\beta - 1)$, $2/(N_\beta - 1)$, $\ldots$, $(N_\beta - 2)/(N_\beta - 1)$, 1, defining N_β sites separated at intervals of $1/(N_\beta - 1)$. The distance between hidden node h and an input node i or output node o was then defined as

$$\left| \frac{h-1}{N_{\text{hidden}}-1} - \frac{i-1}{N_{\text{in}}-1} \right|, \qquad \text{or} \qquad \left| \frac{h-1}{N_{\text{hidden}}-1} - \frac{o-1}{N_{\text{out}}-1} \right|, \tag{5}$$

for $1 \leq h \leq N_{\text{hidden}}$, $1 \leq i \leq N_{\text{in}}$ and $1 \leq o \leq N_{\text{out}}$. The distances were sorted and the weights $w_{ij}^{(1)}$ and $w_{jk}^{(2)}$ corresponding to a specified number of nearest neighbours were retained. Weights corresponding to connections outside the neighbour cutoff were frozen at zero, with all bias weights retained. When it was necessary to choose between neighbours at the same distance we simply selected the input or output node with the lowest index i or o.

This scheme is related to the dropout procedure, where nodes are randomly removed during training.[68,69] Dropout helps to prevent overfitting in large networks, and also reduces the problem of local regions of the network coadapting, which can degrade the predictive capabilities in testing. The present formulation is closer to the DropConnect procedure, which removes connections rather than nodes.[70] However, unlike both DropOut and DropConnect, the architecture remains fixed during training in our analysis. Furthermore, the connections are not removed at random, but instead are omitted to define a locality in the network. This

construction is similar to previous work by LeCun et al. who systematically reduced network connectivity using a weight saliency metric based on second derivative information.[71] The present analysis is designed to test whether the global connections between adjacent layers of the network are responsible for the single-funnelled appearance of the MLL, which has been observed in previous studies.[13,14,17] The potential energy landscapes of atomistic systems generally exhibit more local minima and transition states for short-range forces.[72–75] Introducing reduced connectivity based on locality might have a systematic effect on machine learning landscapes, and we wish to investigate this possibility for the present setup. Our formulation also gives an indication of how reducing the capacity of a neural network is manifested in the underlying landscape.

The potential defined in terms of neighbourhood connectivity described above was used to generate databases of minima and transition states for the D1 dataset with the [2,10,4,1000,0.0001] and [2,5,4,1000,0.00001] single-layered architectures; this dataset was chosen because it has a relatively large number of minima. Landscapes for 1, 2, 3 nearest-neighbours, and the fully-connected [2,10,4,1000,0.0001] model, corresponding to 40, 20, 10 and 0 frozen weights, were created and visualized using disconnectivity graphs. To study generalisability, all the minima obtained were coloured by testing AUC (Figure 6).

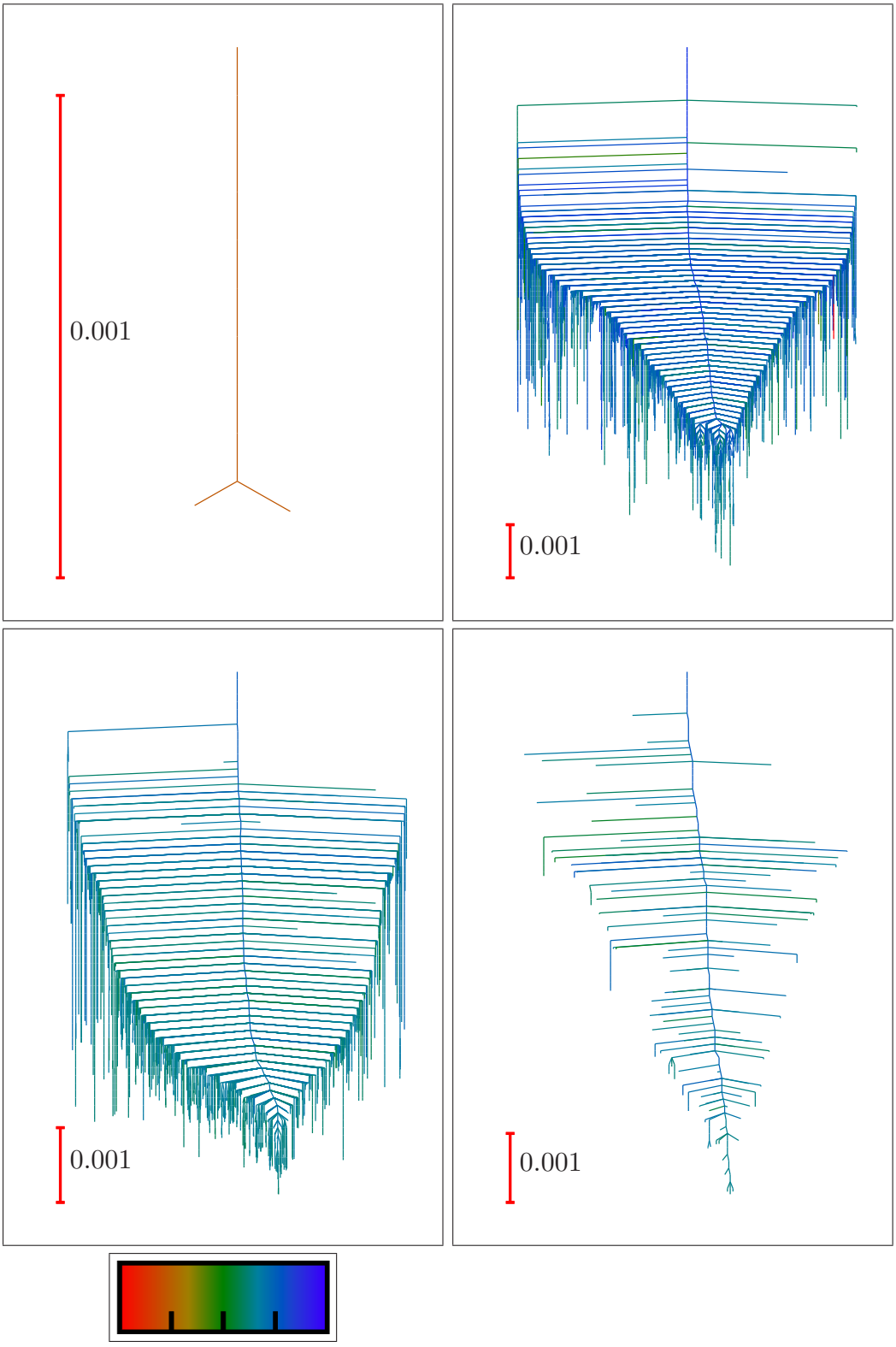


FIG. 6: Disconnectivity graphs for 1 (top left), 2 (top right) and 3 (bottom left) nearest-neighbours for the D1 dataset, compared to the fully-connected architecture (bottom right). The colouring runs from red (low testing AUC) to blue (high testing AUC).

For two and three nearest-neighbours, the number of stationary points increased significantly from the fully connected reference. This situation is consistent with previous results for interatomic potentials with short-range forces in molecular systems.[72–75] The present analysis also suggests that strong locality can induce more complex machine learning landscapes. This conjecture is supported by recent results for two- and three-layered perceptrons, which can have more locality than the single-layered perceptrons, and exhibit more local minima for a similar number of edge weight variables.[18]

Local minima for the two- and three-neighbour networks performed reasonably well on an unseen testing set, with two nearest-neighbours even outperforming the fully-connected model (Figure 6). One possible reason for this phenomenon is the DropOut argument; i.e. the reduced neural network minimises the problem of local regions of network coadaptation, and instead produces a small number of connections, which are independently good at predicting the correct class.[69,71] Another possibility is that the new network has broken symmetry and therefore no longer has highly degenerate solutions arising from parameter permutation, which may facilitate expression of more complex fitting functions.[76] This perspective is at least partially substantiated by the observation of much more complicated landscapes for reduced connectivity (Figure 6). Interestingly, however, only two poorly performing minima were found for the one nearest-neighbour model. This observation likely reflects the fact that the architecture has significantly reduced capacity, since more than half the trainable weights are zero. Overall, our results suggest that in terms of the landscape, optimal architectures may balance sparsity and expressiveness to perform well on unseen testing sets.

Although the reduced-connectivity landscapes obtained for the [2,10,4,1000,0.0001] architecture were significantly more frustrated than the fully-connected model, they were still relatively single-funnelled (Figure 6). To determine whether we could obtain glassy or multi-funnelled landscapes, we visualized disconnectivity graphs for two and three nearest-neighbours and significantly reduced regularization (10-fold) architecture [2,5,4,1000,0.00001] (Figure 7).

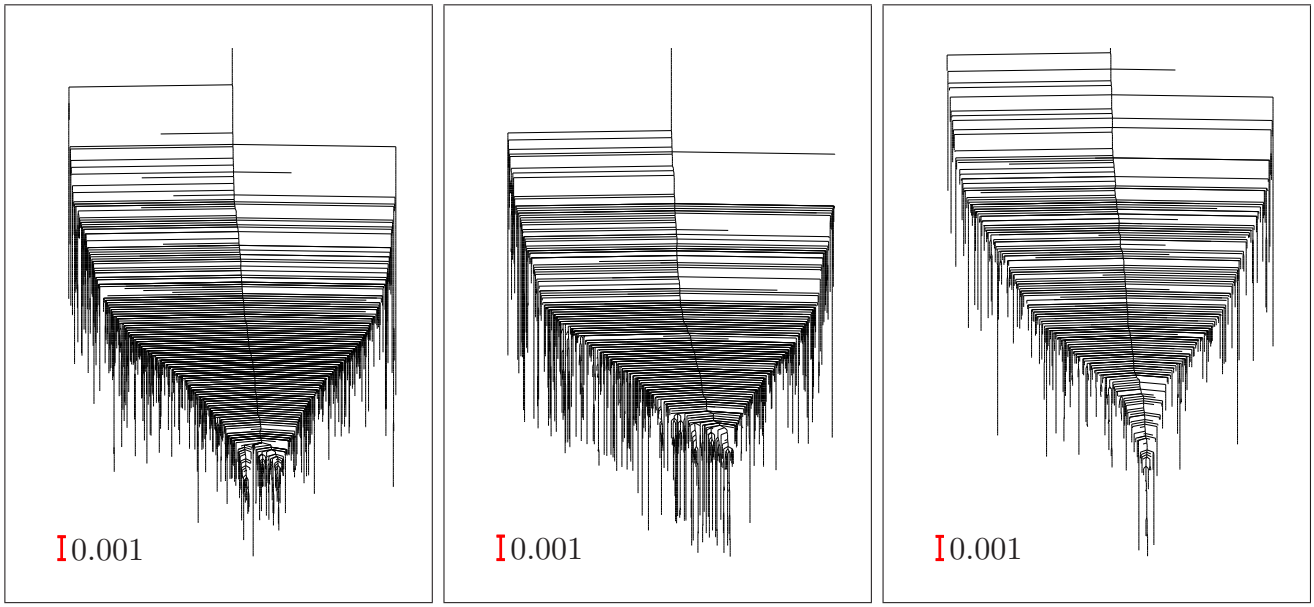


FIG. 7: Disconnectivity graphs for 2 (left) and 3 (middle) nearest-neighbours with reduced connectivity for the D1 dataset, compared to the fully-connected architecture (right). These results are for five nodes in the hidden layer and $\lambda = 0.00001$.

Since the regularization term is a convex **L2** penalty, it is possible that part of the single-funnelled appearance of the reduced-connectivity networks is due purely to regularization; i.e. higher **L2** regularization convexifies the landscape.[15] Again, for the fully-connected case, we observed a single-funnelled appearance, substantiating our previous suggestion that this type of landscape is architecture dependent. However, for the two and three nearest-neighbour models, we observe that some additional sub-funnel structure starts to emerge (Figure 7). This result highlights the strong effect of locality on single-layered architectures.

VI. LANDSCAPES FOR TWO AND THREE HIDDEN LAYERS

Here we present some results for the D1 dataset obtained with neural networks containing two (2HL) and three (3HL) hidden layers, to provide comparisons with the single hidden layer results. we consider 2HL with five nodes in each hidden layer, and 3HL with four nodes in each hidden layer, giving 69 and 72 training variables, respectively, for the D1 dataset obtained for the LJAT_3 classification problem. Disconnectivity graphs that focus on the lower-lying region of the landscape are shown in Figure 8, and the corresponding stationary point databases are described in Table IV. These results illustrate two trends, namely, the

growth in the number of stationary points with increasing hidden layers, and with decreasing training data, for a comparable number of variable edge weights. The corresponding databases are far from complete for $N_{\rm data} = 100$, but should provide a reasonable coverage of the low-lying region, which is the focus of interest here.

Comparing the lower panels of Figure 8 for $N_{\rm data} = 1000$, with the top panels for $N_{\rm data} = 100$, we see than the uphill barriers corresponding to pathways that lead to the global minimum are significantly smaller. Further analysis also shows that the minima span a wider range of loss function values for the 3HL architecture. These effects are maintained when more training data is included; a systematic analysis will be presented elsewhere.[18]

	100	1000
Hidden layers (%)	D1 (Min,Ts)	D1 (Min,Ts)
2	65591, 90622	3630, 3197
3	193036, 540962	13298, 20777

TABLE IV: Number of minima (Min) and transition states (Ts) for machine learning landscapes with two and three hidden layers, $\lambda = 0.0001$, for 100 and 1000 training data drawn from the D1 dataset.

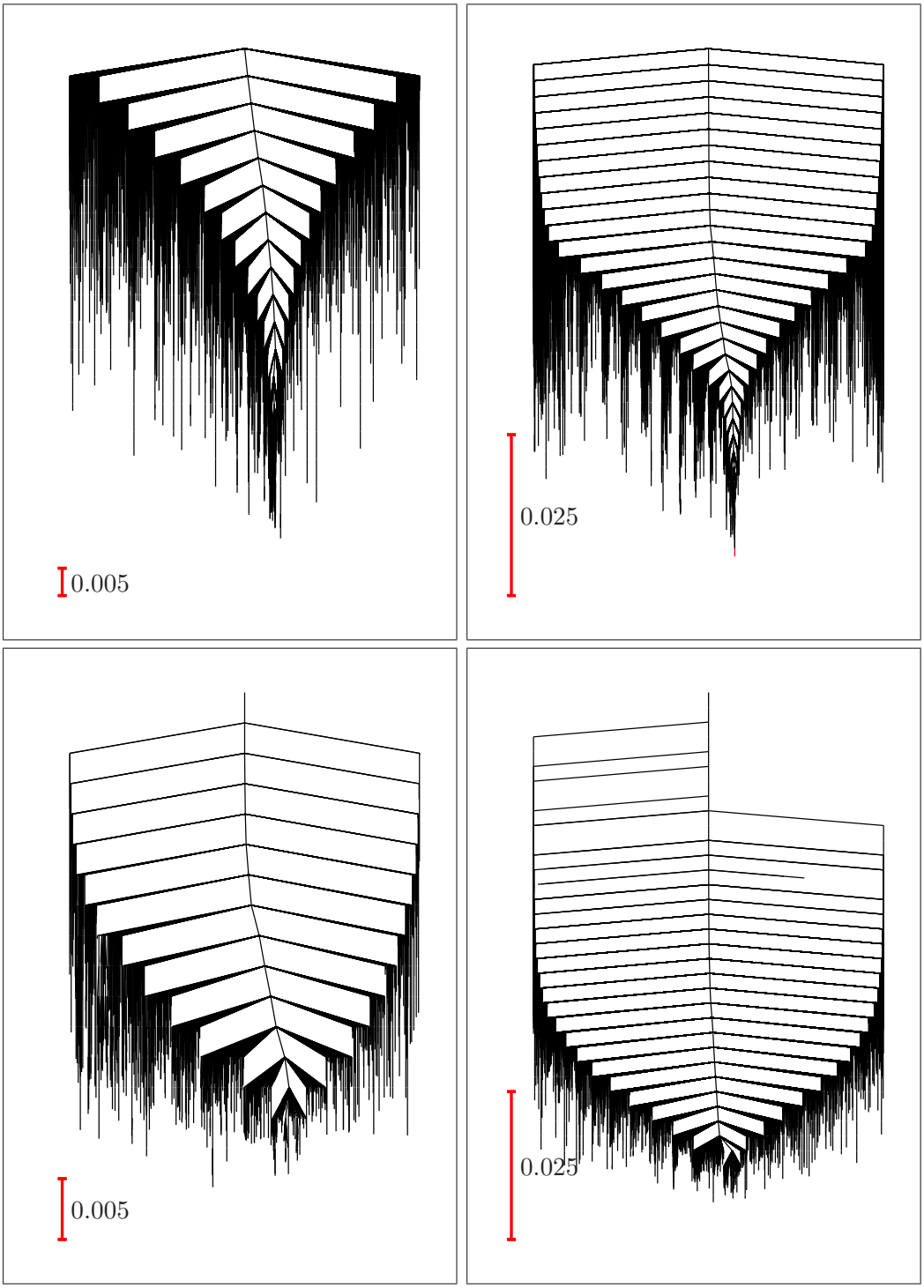


FIG. 8: Disconnectivity graphs obtained with 100 (top) and 1000 (bottom) training data for the D1 dataset $\lambda = 0.0001$ and neural networks with two (left) and three (right) hidden layers.

VII. CONCLUSIONS

Using custom generated high-quality geometry optimisation training data we showed that increasing training diversity (in this case, configuration space volume for an atomic cluster) leads to landscapes with many more stationary points and higher loss values. These results suggest a correspondence between the number of local minima and the statistical uncertainty of the loss function landscape.

In our mislabelling analysis, we found that neural networks can correctly filter uniform noise for very high levels of dataset poisoning and these results remain (empirically) true for averages over the database local minima. We also find that for mislabelling, a tight band of minima around the global minimum does not occur. Instead, the variance of the testing AUC increases significantly with the training error. Furthermore, we observe that many high loss training minima perform well on unseen testing input, as they do not overfit to noise, highlighting a bias-variance type trade-off. In future work we aim to consider other types of noise. Much of the realistic (and difficult) noise in machine learning datasets is not uniform, but instead highly feature dependent or adversarial.[66,77] As a first step, we plan to see whether a landscape analysis might reveal why it is more difficult to train under stochastic permutation noise than uniform random noise. We would also like to compare our noise analysis to neural networks with more than one hidden layer, which may be more resilient to labelling noise.[62]

We have also explored the landscapes of neural networks with reduced connectivity. For two and three nearest-neighbours, the networks retained sufficient expressive capacity. In particular, the network for two nearest-neighbours systematically outperformed the fully-connected case on unseen testing data. The networks with reduced connectivity are significantly more complex, due to the effects of stronger locality and the symmetry-broken architecture. For very limited connectivity (one nearest-neighbour), we found only a few minima with poor predictive capability, reflecting the reduced capacity of the network. Furthermore, as we reduced the regularization (convexity) of the landscape, the reduced-connectivity architecture produced much loss function landscapes with emerging subfunnel structure. These results may be helpful in understanding the difference between the performance of deep networks and shallow networks, and in determining architectures to obtain optimal capacity for neural networks (sparseness versus expressible trade-off). Future work

in this area will likely include a generalised systematic scheme for reduced-connectivity of deep neural networks. In particular, the trends observed for loss function landscapes as a function of the number of hidden layers, the number of training data, and the presence of mislabelling, should be investigated to test whether we can legitimately extrapolate to large networks where a detailed analysis of the landscape is not feasible.

VIII. ACKNOWLEDGEMENTS

This research was funded by the EPSRC.

Supplementary Information: calculation of the AUC measure, results for the D3 dataset and training set distributions.

Data Availability: the data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]D. J. Wales. *Energy Landscapes*. Cambridge University Press, Cambridge, 2003.

[2]J. N. Murrell and K. J. Laidler. Symmetries of activated complexes. *Trans. Faraday. Soc.*, 64:371–377, 1968.

[3]A. Anandkumar and R. Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. In *Conference on learning theory*, pages 81–102, 2016.

[4]Yann Dauphin, Razvan Pascanu, Çaglar Gülçehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *CoRR*, abs/1406.2572, 2014.

[5]F. H. Stillinger and T. A. Weber. packing structures and transitions in liquids and solids. *Science*, 225:983, 1984.

[6]J. P. K. Doye and D. J. Wales. Saddle points and dynamics of lennard-jones clusters, solids, and supercooled liquids. *J. Chem. Phys.*, 116:3777–3788, 2002.

[7]Anna Choromanska, M. B. Henaff, Michaël Mathieu, Gerard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *International Conference on Articial Intelligence and Statistics (AISTATS)*, 2015.

[8]L. Wu, Z. Zhu, and W. E. Towards understanding generalization of deep learning: Perspective of loss landscapes, 2017.

[9]H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *Adv. Neural Inf. Process. Syst.*, pages 6389–6399, 2018.

[10]Q. Nguyen and M. Hein. The loss surface of deep and wide neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2603–2612. JMLR. org, 2017.

[11]G. Swirszcz, W. Czarnecki, and R. Pascanu. Local minima in training of deep networks. 2016.

[12]A. J. Ballard, R. Das, S. Martiniani, D. Mehta, L. Sagun, J. D. Stevenson, and D. J. Wales. Energy landscapes for machine learning. *Phys. Chem. Chem. Phys.*, 19:12585–12603, 2017.

[13]A. J. Ballard, J. D. Stevenson, R. Das, and D. J. Wales. Energy landscapes for a machine learning application to series data. *J. Chem. Phys.*, 144:124119, 2016.

[14]R. Das and D. J. Wales. Machine learning prediction for classification of outcomes in local minimisation. *Chem. Phys. Lett.*, 667:158 – 164, 2017.

[15]D. Mehta, X. Zhao, E. A. Bernal, and D. J. Wales. Loss surface of xor artificial neural networks. *Phys. Rev. E*, 97:052307, 2018.

[16]M. Pavlovskaia, K. Tu, and S.-C. Zhu. Mapping energy landscapes of non-convex learning problems. *arXiv:1410.0576 [stat.ML]*, 2014.

[17]R. Das and D. J. Wales. Energy landscapes for a machine-learning prediction of patient discharge. *Phys. Rev. E*, 93:063310, 2016.

[18]P. C. Verpoort, A. A. Lee, and D. J. Wales. Machine learning landscapes for artificial neural networks (in preparation). 2019.

[19]B. Irie and S. Miyake. Capabilities of three-layered perceptrons. In *IEEE International Conference on Neural Networks*. IEEE, 1988.

[20]K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[21]C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[22]D. J. Wales. Exploring energy landscapes. *Ann. Rev. Phys. Chem.*, 69:401–425, 2018.

[23]R. Das and D. J. Wales. Machine learning landscapes and predictions for patient outcomes. *Royal Society Open Science*, 4:170175, 2017.

[24]S. R. Chitturi. Machine learning landscapes for neural networks with a single hidden layer. MPhil thesis, 2019. University of Cambridge.

[25]`GMIN`: A program for basin-hopping global optimisation, basin-sampling, and parallel tempering. http://www-wales.ch.cam.ac.uk/software.html.

[26]`OPTIM`: A program for geometry optimisation and pathway calculations. http://www-wales.ch.cam.ac.uk/software.html.

[27]`PATHSAMPLE`: A program for generating connected stationary point databases and extracting global kinetics. http://www-wales.ch.cam.ac.uk/software.html.

[28]Z. Li and H. A. Scheraga. Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proc. Natl. Acad. Sci. USA*, 84:6611–6615, 1987.

[29]Z. Li and H. A. Scheraga. Structure and free energy of complex thermodynamic systems. *J. Mol. Struct.*, 179:333, 1988.

[30]D. J. Wales and J. P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *J. Phys. Chem. A*, 101:5111, 1997.

[31]N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[32]J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.

[33]D. C. Liu and J. Nocedal. On limited memory bfgs method for large scale optimization. *Math. Prog.*, 45:503, 1989.

[34]C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA J. Appl. Math.*, 6(1):76–90, 1970.

[35]R. Fletcher. A new approach to variable metric algorithms. *Comput. J.*, 13(3):317–322, 1970.

[36]D. Goldfarb. A family of variable-metric methods derived by variational means. *Math. Comp.*, 24(109):23–26, 1970.

[37]D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Math. Comp.*, 24(111):647–656, 1970.

[38]S. A. Trygubenko and D. J. Wales. A doubly nudged elastic band method for finding transition states. *J. Chem. Phys.*, 120:2082–2094, 2004.

[39]S. A. Trygubenko and D. J. Wales. Analysis of cooperativity and localization for atomic rearrangements. *J. Chem. Phys.*, 121:6689–6697, 2004.

[40]G. Henkelman, B. P. Uberuaga, and H. Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *J. Chem. Phys.*, 113:9901–9904, 2000.

[41]G. Henkelman and H. Jónsson. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *J. Chem. Phys.*, 113:9978–9985, 2000.

[42]L. J. Munro and D. J. Wales. Defect migration in crystalline silicon. *Phys. Rev. B*, 59:3969–3980, 1999.

[43]Yi Zeng, Penghao Xiao, and Graeme Henkelman. Unification of algorithms for minimum mode optimization. *J. Chem. Phys.*, 140(4):044115, 2014.

[44]A. Banerjee, N. Adams, J. Simons, and R. Shepard. search for stationary points on surfaces. *J. Phys. Chem.*, 89:52–57, 1985.

[45]J. P. K. Doye and D. J. Wales. Surveying a potential energy surface by eigenvector- following - applications to global optimisation and the structural transformations of clusters.

Zeit. Phys. D, 40:194–197, 1997.

[46]F. Rao and A. Caflisch. The protein folding network. *J. Mol. Biol.*, 342:299–306, 2004.

[47]F. Noé and S. Fischer. Transition networks for modeling the kinetics of conformational change in macromolecules. *Curr. Opin. Struct. Biol.*, 18:154–162, 2008.

[48]D. Prada-Gracia, J. Gómez-Gardenes, P. Echenique, and F. Fernando. Exploring the free energy landscape: From dynamics to networks and back. *PLoS Comput. Biol.*, 5:e1000415, 2009.

[49]D. J. Wales. Energy landscapes: Some new horizons. *Curr. Opin. Struct. Biol.*, 20:3–10, 2010.

[50]O. M. Becker and M. Karplus. The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *J. Chem. Phys.*, 106:1495–1517, 1997.

[51]D. J. Wales, M. A. Miller, and T. R Walsh. Archetypal energy landscapes. *Nature*, 394:758–760, 1998.

[52]disconnectionDPS: A program for creating disconnectivity graphs. http://www-wales.ch.cam.ac.uk/software.html.

[53]F. Despa, D. J. Wales, and R. S. Berry. Archetypal energy landscapes: Dynamical diagnosis. *J. Chem. Phys.*, 122:024103, 2005.

[54]François Zielinski, Peter I Maxwell, Timothy L Fletcher, Stuart J Davie, Nicodemo Di Pasquale, Salvatore Cardamone, Matthew JL Mills, and Paul LA Popelier. Geometry optimization with machine trained topological atoms. *Scientific reports*, 7(1):12817, 2017.

[55]Anna Styrcz, Janusz Mrozek, and Grzegorz Mazur. A neural-network controlled dynamic evolutionary scheme for global molecular geometry optimization. *International Journal of Applied Mathematics and Computer Science*, 21(3):559–566, 2011.

[56]M. R. Lemes, C. R. Zacharias, and A. Dal Pino. Application of neural networks: a molecular geometry optimization study. In *Proceedings. Vol.1. Sixth Brazilian Symposium on Neural Networks*, pages 288–, Nov 2000.

[57]J. E. Jones and A. E. Ingham. On the calculation of certain crystal potential constants, and on the cubic crystal of least potential energy. *Proc. R. Soc. A*, 107:636–653, 1925.

[58]B. M. Axilrod and E. Teller. Interaction of the van der waals type between three atoms. *J. Chem. Phys.*, 11:299, 1943.

[59]P. G. Mezey. *Potential Energy Hypersurfaces*. Elsevier, Amsterdam, 1987.

[60]Stefano Martiniani, K. Julian Schrenk, Jacob D. Stevenson, David J. Wales, and Daan Frenkel. Turning intractable counting into sampling: Computing the configurational entropy of three-dimensional jammed packings. *Phys. Rev. E*, 93:012906, 2016.

[61]K. Swersky, J. Snoek, and R. P. Adams. Freeze-thaw bayesian optimization. *arXiv:1406.3896 [stat.ML]*, 2014.

[62]D. Rolnick, A. Veit, S. Belongie, and N. Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

[63]G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proc. IEEE. Comput. Soc. Conf. Comput. Vis. Pattern. Recognit.*, pages 1944 – 1952, 2017.

[64]A. J. Bekker and J. Goldberger. Training deep neural-networks based on unreliable labels. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2682 – 2686. IEEE, 2016.

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

[66]C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[67]L. Deng. The mnist database of handwritten digit images for machine learning research. *Signal Process. Mag.*, 29(6):141–142, 2012.

[68]N. Srivastava. Improving neural networks with dropout. masters thesis, university of toronto. 2013.

[69]N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Machine Learning Res.*, 15:1929–1958, 2014.

[70]L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013.

[71]Y. LeCun, J. S Denker, and S. A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[72]P. A. Braier, R. S. Berry, and D. J. Wales. How the range of pair interactions governs features of multidimensional potentials. *J. Chem. Phys.*, 93:8745, 1990.

[73]J. P. K. Doye and D. J. Wales. The structure and stability of atomic liquids - from clusters to bulk. *Science*, 271:484–487, 1996.

[74]D. J. Wales. A microscopic basis for the global appearance of energy landscapes. *Science*, 293:2067–2069, 2001.

[75]D. J. Wales. Highlights: Energy landscapes of clusters bound by short-ranged potentials. *ChemPhysChem*, 11:2491–2494, 2010.

[76]S. Changpinyo, M. Sandler, and A. Zhmoginov. The power of sparsity in convolutional neural networks. *arXiv preprint arXiv:1702.06257*, 2017.

[77]I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.