# Investigating the relationship between diversity and generalization in deep neural networks

Anonymous Full Paper
Submission 60

## Abstract

In ensembles, improved generalization is frequently attributed to *diversity* among members of the ensemble. By viewing a single neural network as an implicit ensemble, we apply well-known ensemble diversity measures to study the relationship between diversity and generalization in artificial neural networks. Our results show that i) deeper layers of the network have higher levels of diversity and ii) layer-wise accuracy positively correlates with diversity. Additionally, we study the effects of well-known regularizers such as Dropout, DropConnect and batch size, on diversity and generalization. We generally find that increasing the strength of the regularizer increases the diversity in the neural network and this increase in diversity is positively correlated with model accuracy. We show that these results hold for several benchmark datasets (such as Fashion-MNIST and CIFAR-10) and architectures (MLPs and CNNs). Our findings suggest new avenues of research into the generalization ability of deep neural networks.

## 1 Introduction

A complete understanding of why deep neural networks (DNNs) generalize well to unseen data remains an open problem. For example, it is well-known that overparameterized neural networks achieve good generalization despite interpolating their training data [1–3]. Furthermore, rapid progress has been made in developing methods, called *regularization*, that encourages generalization. Examples of these methods include *Dropout* [4], *weight decay* [5], *input or weight noise* [5].

A widely-used regularization method is *ensembling*, where the output of several models are aggregated to produce a final output. Crucial to the generalization ability of the ensemble is the *diversity* of the models [6–9]. Depending on the task, there are several definitions of diversity that may be used [10]. For example, diversity can be measured by the correlation between the models' predictions in the ensemble, in which case, low correlation would indicate high diversity. In general, higher diversity among the models is thought to correspond to better generalization of the ensemble, albeit with a trade-off where too much diversity can negatively impact generalization [6, 7, 11, 12].

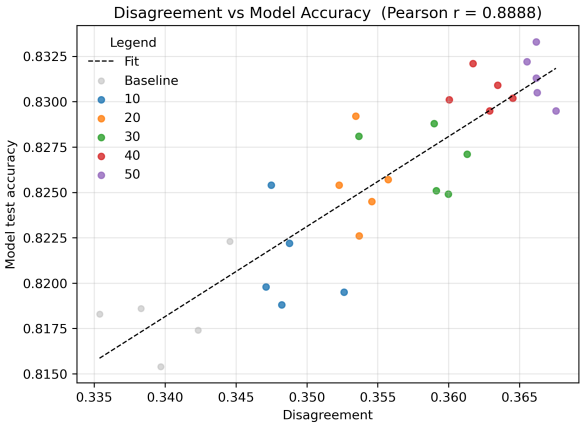


**Figure 1.** Model performance against averaged diversity (across layers) for models trained on the Fashion-MNIST dataset using DropConnect as a regularizer with values in the range $0-50$. Diversity is shown by the disagreement measure. A higher disagreement value constitutes a higher diversity. There is a clear increase in model performance as the averaged diversity increases.

Recently, an insightful approach to investigating generalization in deep learning models has been to view a single deep learning model as an *implicit ensemble*. For the problem of vanishing gradients in deep residual networks, Veit et al. [13] view a deep residual network as a collection of paths and show that the paths have ensemble-like behavior. Another approach by Olson et al. [14] decomposes a single neural network into an ensemble of low-bias subnetworks and, by using correlation as a proxy for diversity and showing that the subnetworks exhibit low correlation, argues that an internal regularization process helps mitigate overfitting in neural networks. Noting regularities in the activation patterns of the hidden nodes of a DNN, Davel et al. [15] view a single hidden node as a classifier. More recently, through investigating the problem of catastrophic forgetting in continual learning, Benjamin et al. [16] show that a neural network in the lazy regime can be decomposed into an implicit ensemble consisting of the weights of the neural network.

However, in viewing a neural network as an implicit ensemble, the role of diversity in a neural network remains unexplored. In this paper, we follow the approach of Davel et al. [15] and view a single neural network as an implicit ensemble with hidden nodes as classifiers. This allows us to investigate the

role of diversity with respect to generalization of the network. The main contributions of the paper are as follows:

1. We examine the diversity of the hidden nodes using established diversity measures.
2. We empirically investigate the relationship between node-level diversity and generalization across several benchmark datasets and for different architectures.
3. We analyze the effect of well-known regularization methods that encourages generalization and examine the effect of these methods on diversity. We show that diversity correlates with the generalization of the neural network (see Figure 1).

Our results provide new insight into the ability of neural networks to generalize and offers new avenues of research into the generalization of neural networks.

# 2 Background

## 2.1 Ensemble Methods

Ensemble methods combine predictions of multiple classifiers to achieve better generalization than individual models. Classic approaches such as bagging [17] and boosting [18], have shown that ensembles reduce variance, improve robustness, and often outperform single models across diverse tasks.

A key factor to the success of ensembles is *diversity* among the base classifiers. If all classifiers make identical errors, the ensemble offers no advantage. However, when classifiers make different errors, the ensemble can correct individual mistakes, yielding improved accuracy [6, 7]. Theoretical and empirical studies have shown that ensembles benefit most when base learners are both accurate and diverse [10]. Diversity can be defined in several ways, for example as the degree of correlation between classifiers' predictions, with lower correlation implying higher diversity.

## 2.2 Diversity Measures

Several measures have been proposed to quantify diversity in ensembles [10]. In this paper, we focus on four well-known metrics: disagreement, double-fault, Q-statistic, and entropy.

These measures (besides entropy) are based on the outcomes of pairs of classifiers, which can be summarized using a $2 \times 2$ contingency table (Table 1). Assuming a $C$ class classification problem, let $D_i$ and $D_k$ be two classifiers. Given a data set $\{(x^{(i)}, y^{(i)})\}_{i=1}^{s}$, let $N^{11}, N^{10}, N^{01}$, and $N^{00}$ denote the frequency of the following cases, respectively: both $D_i$ and $D_k$ correct, $D_i$ correct and $D_k$ incorrect, $D_k$ correct and $D_i$ incorrect, and both $D_i$ and $D_k$ incorrect.

We consider three pairwise metrics, namely:

- **Disagreement**

$$Dis_{i,k} = \frac{N^{10} + N^{01}}{N}, \tag{1}$$

where $N$ is the total number of samples. This measures the proportion of instances where the classifiers disagree. Higher disagreement relates to higher diversity.

- **Double-fault**

$$df_{i,k} = \frac{N^{00}}{N}, \tag{2}$$

This measures the proportion of instances where both classifiers misclassify the same sample. A lower double-fault relates to a higher diversity (they fail on different samples).

- **Q-statistic**

$$Q = \frac{(N^{11}N^{00} - N^{10}N^{01})}{(N^{11}N^{00} + N^{10}N^{01})} \tag{3}$$

Values range from $-1$ to $+1$, with 0 indicating independence (which we view as the highest level of diversity).

We also consider one non-pairwise metric:

- **Entropy** Entropy quantifies the uncertainty in ensemble predictions and serves as a proxy for classifier disagreement [10, 19]. For each sample $x^{(k)}$, we estimate the probability as

$$P(y \mid x^{(k)}) = \frac{1}{L} \sum_{i=1}^{L} \mathbf{1}\{D_i(x^{(k)}) = y\}, \tag{4}$$

where $L$ is the number of subpredictors, and $y \in \{1, ..., C\}$. The per sample entropy is:

$$\mathrm{H}(x^{(k)}) = -\sum_{y} P(y \mid x^{(k)}) \log P(y \mid x^{(k)}). \tag{5}$$

We can then calculate the average over the dataset:

$$\mathrm{Ent}_{\mathrm{CC}} = \frac{1}{N} \sum_{k=1}^{N} H(x^{(k)}) \tag{6}$$

Higher values of entropy indicate higher diversity between subpredictor predictions.

These measures provide complementary views of diversity—disagreement emphasizes complementarity, double-fault emphasizes error overlap, Q-statistic emphasizes correlation, and entropy emphasizes overall variety.

**Table 1.** 2x2 contingency table which represents the frequencies of two classifiers $D_i$ and $D_k$. Here, $N^{11}$, $N^{10}$, $N^{01}$, and $N^{00}$ are the observed frequencies in each cell. This means that $N^{11}$ is the number of times that both $D_i$ and $D_k$ are correct, $N^{10}$ is the number of times only $D_i$ is correct, $N^{01}$ the number of times only $D_k$ is correct, and $N^{00}$ the number of times that both $D_i$ and $D_k$ are incorrect.

| | $D_k$ correct (1) | $D_k$ incorrect (0) |
|---|---|---|
| $D_i$ correct (1) | $N^{11}$ | $N^{10}$ |
| $D_i$ incorrect (0) | $N^{01}$ | $N^{00}$ |

# 3 Method

## 3.1 Implicit Ensemble Framework

In order to investigate the effects of diversity in deep neural networks, we view a *single* neural network as an implicit ensemble. In particular, we adopt the framework introduced by Davel et al. [15], which treats each hidden node as a weak classifier. We refer to such a node classifier as a *subpredictor*.

Intuitively, hidden nodes tend to specialize during training. That is to say, they become sensitive to certain patterns or classes. Following Davel et al. [15], we estimate, for each node $n$ and class $c$, the class-conditional probability $P(z_n \mid c)$ of each node's pre-activation $z_n(x)$ given a sample $x$ by applying a kernel density estimator (KDE) trained using all $s$ training samples' activation values observed at the node.

Applying Bayes' rule yields a node-level posterior over classes for an input with pre-activation $z_n$:

$$P(c \mid z_n) = \frac{P(z_n \mid c)\, P(c)}{\sum_{c'} P(z_n \mid c')\, P(c')}, \tag{7}$$

where $P(c)$ is the class prior. $P(c \mid z_n)$ is then node $n$'s probability for class $c$ when given a certain input. Then, for a given pre-activation $z_n(x)$ and a node $n$ in layer $\ell$, the node's class prediction is computed as $\hat{y}_{\ell,n}^{(x)} = \arg\max_c P(c \mid z_n(x))$. We refer to this as a *hard vote*.

**Majority vote ensemble:** To ensemble the output's of the subpredictors in a layer $\ell$, we employ a simple majority vote rule to determine our layer-wise prediction, and by extension, layer-wise accuracy. These hard votes are then used for computing the diversity of the predictions across nodes within the given layer.

**Nodes as subpredictors** We first demonstrate that individual nodes can indeed be treated as meaningful subpredictors within the implicit ensemble. To do this, we compute the accuracy of each node when predicting class labels. Figure 2 shows a heatmap of node accuracies across layers, while Figure 3 presents the layer-wise accuracy for the train, validation, and test sets. These results confirm that individual nodes can perform as subpredictors.

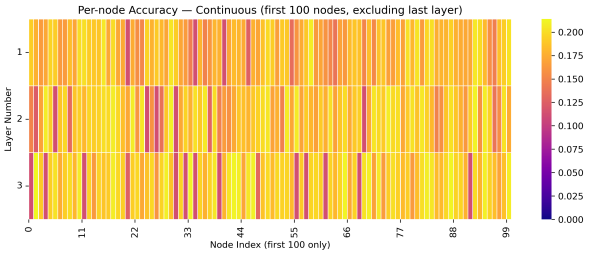


**Figure 2.** Accuracy heatmap of node predictions in a three layer MLP. Only the first 100 nodes are shown here. Most nodes in each layer have an individual accuracy of around 15-20% (higher than random guessing). Results were obtained from the MNIST test set for a model trained on 2,000 samples of the MNIST dataset.

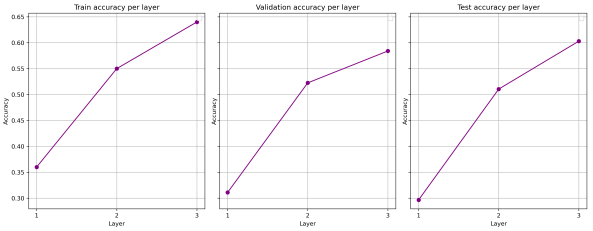


**Figure 3.** Layer-wise accuracy graphs for the train, validation, and test on MNIST. Accuracy improves with network depth. Each layer accuracy has significantly higher accuracy than compared to the individual node accuracies. This clearly showcases the ensembling strength of individual nodes within a layer.

Although the predictive *ability* of individual nodes within a neural network layer may be limited, their performance is consistently better than random chance. When the output of these subpredictors are aggregated, their collective predictions at the layer level yields better accuracies over the training, validation, and testing sets, similar in fashion to the principles of ensemble learning, where multiple 'weak learners' are combined to create a single 'strong learner.'

## 3.2 Applying Diversity Metrics

We focus on the diversity metrics mentioned in Section 2.2.

We compute these metrics directly from the node-level predictions. For pairwise measures like Q-statistic, disagreement, or double-fault, we calculate the average over all unique pairs of nodes within a specific layer. For our non-pairwise metric i.e., entropy, we compute the measures across all nodes in the layer directly.

To facilitate these calculations, we first needed to construct our *contingency tables* (see Table 1) for each combination of subpredictors within a layer. This was done by first creating a binary accuracy matrix, $B$, from our subpredictor predictions. Each

entry, $B_n^{(x)}$, indicates whether node $n$ correctly classified sample $x$ by comparing the sample's true class label $y^{(x)}$ to the node's predicted class label $\hat{y}_{\ell,n}^{(x)}$:

$$B_n^{(x)} = \begin{cases} 1, & \text{if } \hat{y}_{\ell,n}^{(x)} = y^{(x)} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

We then use the binary accuracy matrix to construct the contingency table for each pair of nodes $(n_1, n_2)$. This table records the number of samples for which both nodes made the same or different predictions, based on whether their predictions were correct (1) or incorrect (0). Then using the equations given in 2.2, we can compute each diversity metric across layers.

### 3.3 Training protocol

Due to how easily MLPs learn datasets like MNIST and Fashion-MNIST (with evaluation accuracies often being above 98%), and to more clearly see the effects of inducing diversity, we train the MLP networks on a small stratified subset of the training data (e.g., 2,000 samples) until interpolation (100% training accuracy). This protocol ensures that models generalize to different degrees despite being perfectly fit to the training subset. CNN models are trained using the full CIFAR-10 training set. Models are trained using cross-entropy loss and optimized with the Adam optimizer, and no batch normalization has been applied. All experiments are repeated across five different random model seeds. We report the mean $\pm$ standard deviation for most of our results.

Details regarding datasets used and model architectures can be found in Appendix A.

### 3.4 Inducing diversity through regularization

Regularization methods are commonly used to improve generalization in deep learning [5]. To study the effect of induced diversity, we apply common regularization techniques that we believe should have an effect on the diversity within the network:

- **Dropout:** [4] randomly deactivates nodes during training, forcing different subsets of nodes to specialize. Applied to hidden layers with varying rates (0.1-0.5 for MLPs, 0.1-0.3 for CNNs).
- **DropConnect:** [20] similar to Dropout, except the weights are randomly removed rather than the activations. The same rates are applied as for Dropout.
- **Batch size variation:** [21] smaller batch sizes increase gradient noise which could possibly introduce some diversity to the nodes during training. Models are trained with batch sizes of: 16, 32, 64, 128, 256 (baseline), 512.

**Expected effects of induction:** For *Dropout* and *DropConnect*, increasing the drop probability $p$ should raise the node-level diversity by discouraging co-adaptation and encouraging specialization across stochastic subnetworks [4]. For *batch size*, smaller batches are expected to produce higher diversity because they introduce more gradient noise and more parameter updates per epoch, possible promoting more node-level diversity.

In this work, we use these techniques to explicitly manipulate diversity and investigate its effect on generalization in implicit ensembles.

## 4 Empirical Results

### 4.1 Diversity trends (no induced regularization)

Having established that nodes can act as subpredictors, we evaluate whether classical diversity measures provide meaningful insights in this implicit setting.

On our three-layer MLP, mean diversity (averaged over five seeds) increases with depth for most metrics. Disagreement, Double-fault and entropy all show a monotonic increase from layer 1 to layer 3, with the Disagreement measure illustrated in Figure 4. This overall trend suggests that subpredictors in later layers produce more varied outputs. However, the Q-statistic is an outlier, showing an initial increase followed by a slight decrease in the last layer (Figure B.1). Additional results for the other metrics are available in Table B.1.

The increase in diversity with depth mirrors the trend in layer accuracy, as later layers are more accurate. We found a strong correlation between diversity and layer accuracy across all metrics, in particular, the Disagreement measure shows a near-linear relationship with layer accuracy (Pearson $r = 0.98$, Figure 4). Other metrics show qualitatively similar correlations, though the Q-statistic has a slightly weaker correlation.

Across datasets, the depth trend is reliable only for Disagreement and Double-fault. Q-statistic and entropy vary in their diversity trends by dataset and layer, and do not yield a consistent monotonic pattern, so we treat them as secondary in the cross-dataset summaries (see Table B.1).

### 4.2 Inducing Diversity

**Regularization effects on model performance** We investigate how certain regularization techniques—Dropout, DropConnect, and batch size—affect generalization and diversity.

On Fashion-MNIST, increasing the DropConnect probability $p$ improves model test accuracy (Figure 5). We observe the same pattern with Dropout

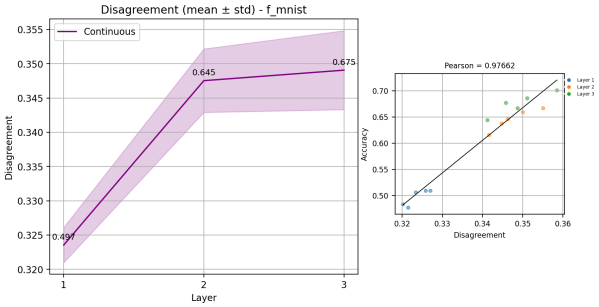


**Figure 4.** *(left)* Disagreement of node-level predictions across layers of a three-layer MLP trained on Fashion-MNIST, averaged over five seeds, and *(right)* the correlation between the disagreement and layer accuracy. The annotated values correspond to the averaged layer accuracies, which rise in tandem with disagreement. The right panel confirms this relationship, showing a strong positive correlation (Pearson $r = 0.98$) between disagreement and layer accuracy across seeds and layers.

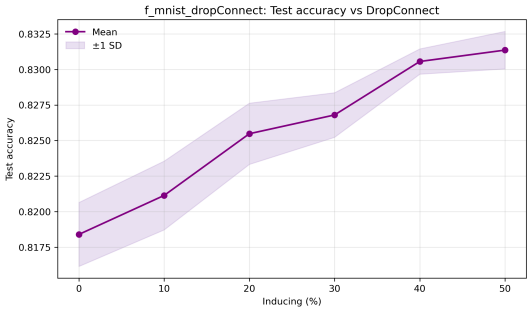


**Figure 5.** Model test performance against percentage DropConnect applied during training for the Fashion-MNIST dataset. There is a clear increasing trend in accuracy as the drop probability being applied is increased.

(when increasing $p$) and with smaller batch sizes (Table C.1).

**Regularization effects on diversity** The same settings that help increase model accuracy, also increases Disagreement and reduces Double-fault. Averaged across layers, disagreement increases (and double-fault decreases) as $p$ rises for Dropout/DropConnect and as batch size decreases. The Q-statistic shows a similar decrease as double-fault on Fashion-MNIST, while entropy has an initial increase followed by a continuous decrease (see Table C.3).

Across datasets, the only robust trends are for disagreement and double-fault; Q-statistic and entropy are unstable and dataset-dependent.

**The link between generalization and diversity** We explicitly test whether induced diversity correlates with generalization on Fashion-MNIST. Our results show that increasing regularizer strength—through higher drop probabilities for Dropout/DropConnect or smaller batch sizes—yields both higher test accuracy and higher average diversity (Figure 1),

**Table 2.** Correlation (Pearson $r$) between model test accuracy and averaged diversity under induced diversity sweeps (DropConnect, Dropout, batch size) for the Fashion-MNIST dataset.

| Diversity measure | DropConnect | Dropout | Batch size |
|---|---|---|---|
| Disagreement | 0.8888 | 0.6289 | 0.5566 |
| Double-fault | -0.8970 | -0.8285 | -0.6734 |
| Q-statistic | -0.7503 | -0.1365 | -0.4831 |
| Entropy | -0.4859 | -0.7126 | 0.4253 |

although the results are metric- and regularizer-dependent.

We see this clearly in Table 2. Disagreement shows a strong positive correlation with accuracy ($r \approx 0.89, 0.63, 0.56$)), while Double-fault is strongly negative ($r \approx -0.90, -0.83, -0.67$). Q-statistic is especially weak under Dropout, and entropy is mostly negative but flips positive for the batch size sweep.

Based on these results, we conclude that there is a positive association between generalization and diversity—higher node-level diversity corresponds to higher test accuracy—most consistently captured by Disagreement and Double-fault.

**Further results** The trends and results mentioned above repeat across MNIST, K-MNIST, and CIFAR-10 datasets and hold for both MLPs and CNNs (Table C.4).

# 5 Conclusion

By viewing a network as an implicit ensemble of node-level classifiers, we measured diversity inside MLPs and a CNN. We find that diversity increases from shallow to deeper layers. We also find that well-known regularizers such as Dropout, Dropconnect, and batch size increases the diversity in a neural network. Additionally, we find that this increase in diversity follows model generalization performance: average diversity correlates with test accuracy.

For the diversity measures applied to a single neural network, we find that disagreement and double-fault are the most reliable measures of diversity, while Q-statistic and entropy are less consistent across datasets and regularizers.

**Future work:** Overall, these results show that diversity could be a used as a method of predicting generalization in a neural network. Additional possible research avenues includes exploring additional diversity induction methods—e.g., adding an explicit diversity term to the loss function— and extending the analysis to other architectures (ResNets, Transformers) and more challenging datasets (CIFAR-100, ImageNet).

# References


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

[2] M. Belkin, D. Hsu, S. Ma, and S. Mandal. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854. DOI: 10.1073/pnas.1903070116.

[3] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. "Deep double descent: Where bigger models and more data hurt". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12 (2021), p. 124003. DOI: 10.1088/1742-5468/ac3a74.

[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958. DOI: 10.5555/2627435.2670313.

[5] S. J. Prince. *Understanding Deep Learning*. The MIT Press, 2023. URL: http://udlbook.com.

[6] L. Hansen and P. Salamon. "Neural Network Ensembles". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1990), pp. 993–1001. DOI: 10.1109/34.58871.

[7] A. Krogh and J. Vedelsby. "Neural network ensembles, cross validation, and active learning". In: *Advances in Neural Information Processing Systems* 7 (1994).

[8] T. G. Dietterich. "Ensemble methods in machine learning". In: *International Workshop on Multiple Classifier Systems*. Springer. 2000, pp. 1–15. DOI: 10.1007/3-540-45014-9_1.

[9] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan. "Ensemble Deep Learning: A Review". In: *Engineering Applications of Artificial Intelligence* 115 (2022), pp. 105–151. DOI: 10.1016/j.engappai.2022.105151.

[10] L. I. Kuncheva and C. J. Whitaker. "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy". In: *Machine Learning* 51 (2003), pp. 181–207. DOI: 10.1023/A:1022859003006.

[11] L. A. Ortega, R. Cabañas, and A. Masegosa. "Diversity and Generalization in Neural Network Ensembles". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 11720–11743. DOI: 10.48550/arXiv.2110.13786.

[12] D. Wood, T. Mu, A. M. Webb, H. W. Reeve, M. Luján, and G. Brown. "A unified theory of diversity in ensemble learning". In: *Journal of Machine Learning Research* 24.359 (2023), pp. 1–49. DOI: 10.48550/arXiv.2301.03962.

[13] A. Veit, M. J. Wilber, and S. Belongie. "Residual networks behave like ensembles of relatively shallow networks". In: *Advances in Neural Information Processing Systems* 29 (2016). DOI: 10.48550/arXiv.1605.06431.

[14] M. Olson, A. Wyner, and R. Berk. "Modern neural networks generalize on small data sets". In: *Advances in Neural Information Processing Systems* 31 (2018). DOI: 10.5555/3327144.3327279.

[15] M. H. Davel, M. W. Theunissen, A. M. Pretorius, and E. Barnard. "DNNs As Layers Of Cooperating Classifiers". In: *Proc. AAAI Conference on Artificial Intelligence (AAAI)*. 2020. DOI: 10.1609/aaai.v34i04.5782.

[16] A. S. Benjamin, C. Pehle, and K. Daruwalla. "Continual learning with the neural tangent ensemble". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 58816–58840. DOI: 10.5555/3737916.3739791.

[17] L. Breiman. "Bagging predictors". In: *Machine Learning* 24 (1996), pp. 123–140. DOI: 10.1007/BF00058655.

[18] Y. Freund and R. E. Schapire. "A Desicion-Theoretic Generalization of on-Line Learning and an Application to Boosting". In: *Computational Learning Theory*. Ed. by P. Vitányi. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 23–37. DOI: 10.1006/jcss.1997.1504.

[19] P. Cunningham and J. Carney. "Diversity versus quality in classification ensembles based on feature selection". In: *European Conference on Machine Learning*. Springer. 2000, pp. 109–116. DOI: 10.1007/3-540-45164-1_12.

[20] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. "Regularization of neural networks using dropconnect". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 1058–1066. DOI: 10.5555/3042817.3043055.

[21] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. *On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima*. 2017. arXiv: 1609.04836 [cs.LG].

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[23] H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv:1708.07747* (2017). DOI: 10.48550/arXiv.1708.07747.

[24] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. "Deep learning for classical Japanese literature". In: *arXiv preprint arXiv:1812.01718* (2018). DOI: 10.48550/arXiv.1812.01718.

[25] A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

# A Model setup

**Datasets** We evaluate our approach on four standard image classification benchmarks:

- MNIST [22], Fashion-MNIST [23], and K-MNIST [24] grayscale digit and clothing classification datasets with 10 classes. K-MNIST is a variant of MNIST with Japanese characters.
- CIFAR10 [25], a colored natural image dataset with 10 classes.

**Architectures** We used different architectures to confirm our results.

- For MNIST, Fashion-MNIST, and K-MNIST, we use fully-connected multi-layered perceptrons (MLPs) with a depth of three hidden layers, each with a width of 512 nodes.
- For CIFAR-10, we use a convolutional neural network (CNN) with multiple convolutional and pooling layers followed by fully connected layers (see Table A.2 for details). Diversity measures are applied only to the fully connected layers, where subpredictors are naturally defined at the node level.

**Hyperparameter setup** The specific details regarding what hyperparameters were used to train each model can be found in Table A.1. All models were trained with a baseline batch size of 256, while the varying batch size experiments then used the other values. No Dropout or DropConnect was applied to normal baseline models. A learning rate scheduler was also applied for all different models.

The specific model architecture for the CNNs can be found in Table A.2.

**Table A.1.** Hyperparameter setup for different model architectures

| Hyperparameter | Value |
|---|---|
| Optimizer | Adam |
| Learning rate | 0.0003 |
| Batch size | {16, 32, 64, 128, 256 (baseline), 512} |
| Epochs (max) | 1000 |
| Dropout/DropConnect rates (MLPs) | {0, 0.1, 0.2, 0.3, 0.4, 0.5} |
| Dropout/DropConnect rates (CNNs) | {0, 0.1, 0.15, 0.2, 0.25, 0.3} |
| Learning rate scheduler | {Step size = 1, gamma = 0.99} |

**Table A.2.** CNN specification (conv bias=True, activation=ReLU, BN=False). "$\times n$" indicates repeated blocks.

| Stage | Layers | Kernel / Stride / Pad | Output size (for 32×32 input) |
|---|---|---|---|
| Input | | | 32×32×3 |
| S1 | Conv 64 ×2 | (3, 1, 1) | 32×32×64 |
| | MaxPool | (2, 2, 0) | 16×16×64 |
| S2 | Conv 128 ×2 | (3, 1, 1) | 16×16×128 |
| | MaxPool | (2, 2, 0) | 8×8×128 |
| S3 | Conv 256 ×3 | (3, 1, 1) | 8×8×256 |
| | MaxPool | (2, 2, 0) | 4×4×256 |
| S4 | Conv 512 ×3 | (3, 1, 1) | 4×4×512 |
| | MaxPool | (2, 2, 0) | 2×2×512 |
| S5 | Conv 512 ×3 | (3, 1, 1) | 2×2×512 |
| | MaxPool | (2, 2, 0) | 1×1×512 |
| Head | Flatten → MLP(3×512) → Dense($C$) | | $C$ |

# B Additional results

**Table B.1.** Average layer-wise accuracy and diversity across datasets (averaged over 5 seeds). Diversity includes Disagreement (Dis), Double-Fault (DF), Q-statistic (Q), and Entropy (Ent).

| Dataset | Layer | Acc. | Dis. | DF | Q | Ent |
|---|---|---|---|---|---|---|
| MNIST | L1 | 0.288 | 0.285 | 0.605 | 0.490 | 2.620 |
| | L2 | 0.523 | 0.315 | 0.558 | 0.467 | 2.690 |
| | L3 | 0.605 | 0.325 | 0.540 | 0.450 | 2.720 |
| K-MNIST | L1 | 0.482 | 0.280 | 0.667 | 0.130 | 3.040 |
| | L2 | 0.533 | 0.304 | 0.643 | 0.159 | 3.050 |
| | L3 | 0.543 | 0.305 | 0.636 | 0.190 | 3.030 |
| Fashion-MNIST | L1 | 0.497 | 0.323 | 0.544 | 0.450 | 2.620 |
| | L2 | 0.645 | 0.347 | 0.512 | 0.387 | 2.710 |
| | L3 | 0.675 | 0.349 | 0.505 | 0.405 | 2.715 |
| CIFAR-10 | L1 | 0.563 | 0.331 | 0.520 | 0.476 | 2.510 |
| | L2 | 0.575 | 0.333 | 0.515 | 0.464 | 2.580 |
| | L3 | 0.596 | 0.336 | 0.507 | 0.455 | 2.595 |

Across datasets (MNIST, K-MNIST, Fashion-MNIST, and CIFAR-10) and architectures (MLPs, CNNs), layer accuracy increases monotonically with depth (Table B.1).

Depth-wise diversity shows parallel but metric-specific trends. Disagreement increases with depth and Double-fault decreases, indicating higher diversity in later layers for all datasets. In contrast, the Q-statistic is dataset-dependent: on Fashion-MNIST it drops from the first to the second layer (increase in diversity) but rebounds in the final layer (breaking the expected monotonic decrease, see Figure B.1), while on K-MNIST it increases with depth (implying *lower* diversity in deeper layers). Entropy generally rises with depth on MNIST,

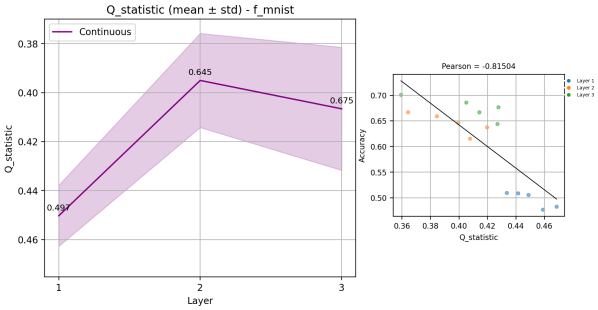


**Figure B.1.** *(left)* Q-statistic of node-level predictions across layers of a three-layer MLP trained on Fashion-MNIST, averaged over five seeds, and *(right)* the correlation between the Q-statistic and layer accuracy. The annotated values correspond to the averaged layer accuracies. The right panel shows a strong relationship between the Q-statistic and layer accuracy (Pearson $r = -0.81$) across seeds and layers.

**Table C.1.** Test accuracy versus induction level. Accuracy values are averaged over five seeds; $\Delta$ indicates the change in percentage points relative to the baseline model (0% dropout/dropconnect or largest batch size).

| Dataset | Method | Induction level | Accuracy (%) |
|---|---|---|---|
| Fashion-MNIST | Dropout | 0% | 0.8178 |
| | | 10% | 0.8197 |
| | | 20% | 0.8216 |
| | | 30% | 0.8222 |
| | | 40% | 0.8244 |
| | | 50% | 0.8260 |
| Fashion-MNIST | DropConnect | 0% | 0.8178 |
| | | 10% | 0.8215 |
| | | 20% | 0.8252 |
| | | 30% | 0.8269 |
| | | 40% | 0.8312 |
| | | 50% | 0.8319 |
| Fashion-MNIST | Batch size | 512 | 0.8170 |
| | | 256 | 0.8178 |
| | | 128 | 0.8260 |
| | | 64 | 0.8285 |
| | | 32 | 0.8345 |
| | | 16 | 0.8360 |

Fashion-MNIST, and CIFAR-10, but remains nearly unchanged across layer on K-MNIST. Overall, Disagreement and Double-fault provide the most consistent depth trends, whereas Q-statistic and entropy vary by dataset (Table B.1).

# C Induced diversity

**Accuracy increase from Dropout** Test accuracy increases monotonically with stronger dropout, but the gains are modest: from 0.8182 at 0% to 0.8260 at 50% (+0.82). This suggests small but consistent benefits over the explored Dropout range (Table C.1).

**Accuracy increase from DropConnect** Weight-level masking yields a larger improvement than normal Dropout on Fashion-MNIST: accuracy rises from 0.8178 to 0.8319 at 50% (+1.41), with a steady increase across all levels (Table C.1).

**Table C.2.** Test accuracy versus induction level. Accuracy values are averaged over five seeds; $\Delta$ indicates the change in percentage points relative to the baseline model (0% dropout/dropconnect or largest batch size). Batch size models were trained on a subset of the training data (5,000 samples) for CIFAR-10.

| Dataset | Method | Induction level | Accuracy (%) |
|---|---|---|---|
| CIFAR-10 | Dropout | 0% | 0.759 |
| | | 10% | 0.791 |
| | | 15% | 0.795 |
| | | 20% | 0.793 |
| | | 25% | 0.792 |
| | | 30% | 0.794 |
| CIFAR-10 | DropConnect | 0% | 0.759 |
| | | 10% | 0.794 |
| | | 15% | 0.799 |
| | | 20% | 0.794 |
| | | 25% | 0.794 |
| | | 30% | 0.792 |
| CIFAR-10 | Batch size | 512 | 0.500 |
| | | 256 | 0.489 |
| | | 128 | 0.520 |
| | | 64 | 0.552 |
| | | 32 | 0.564 |
| | | 16 | 0.568 |

**Accuracy increase from varying batch sizes** Reducing the batch size delivers sizable gains: the accuracy increases from 0.8170 (512 batch size, baseline is 256) to 0.8360 (16 batch size), a +1.90 improvement, with improvements at each reduction step (Table C.1).

Similar accuracy increases were obtained from models trained on the MNIST and K-MNIST datasets.

**CIFAR-10** For CIFAR-10, batch sizes sweeps were run on a 5,000 stratified subset to avoid confounds we observed on the full training set, where extremely small batches produced so many updates that their diversity appeared 'averaged out,' inverting the expected ordering. We expect the ordering to recover at larger batch sizes (e.g., 512 vs 1,024), but we did not test this. See Table C.2.

**Accuracy increase from Dropout** Test accuracy improves with moderate dropout, peaking at 15% (0.759 − 0.795; +3.6%) and remaining stable for 20 − 30% (Table C.2). From the literature, it is widely accepted that a moderate amount of Dropout applied to CNNs can boost performance, whereas excessive Dropout can negatively impact the performance [4].

**Accuracy increase from DropConnect** Again, weight-level masking appears to yield the largest

**Table C.3.** Effect of DropConnect rate on layer-wise diversity (averaged over runs) for the Fashion-MNIST dataset. Disagreement increases with a higher drop rate, while Double-fault and Q-statistic generally decrease; entropy shows a mild non-monotonic trend.

| Diversity measure | Drop rate $p$ (%) | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 10 | 20 | 30 | 40 | 50 |
| Disagreement | 0.340 | 0.348 | 0.354 | 0.358 | 0.363 | 0.366 |
| Double-fault | 0.520 | 0.514 | 0.505 | 0.495 | 0.487 | 0.476 |
| Q-statistic | 0.418 | 0.381 | 0.373 | 0.368 | 0.364 | 0.361 |
| Entropy | 2.677 | 2.710 | 2.700 | 2.691 | 2.668 | 2.653 |

model-level gain, peaking at 15% (0.759 − 0.799; +4.0%), with small declines at higher rates (Table C.2).

**Accuracy increase from varying batch sizes** On the 5,000 sample subset, reducing the batch size improves accuracy monotonically over most of the range (0.500 at 512 to 0.568 at 16; +6.8%), with gains becoming pronounced below 128 batch size (Table C.2).

**DropConnect effects on diversity** DropConnect generally increases diversity. As the drop rate rises from 0% to 50%, Disagreement increases steadily, while Double-fault and Q-statistic decrease. Together, these shifts indicate higher averaged diversity with a higher rate of DropConnect being applied (Table C.3).

Entropy is less stable. Entropy shows a mild, non-monotonic pattern with a small net decline, suggesting it is a weaker or more dataset-sensitive indicator in this setting. Overall, the consistent monotone trends in Disagreement, Double-fault, and Q-statistic support the view that DropConnect reliably induces diversity in Fashion-MNIST (although Q-statistic has inconsistent layer-wise diversity trends).

**Diversity tracks generalization** Across datasets and induction methods, induced diversity is strongly correlated with higher test accuracy (Table C.4). Disagreement is consistently positive (e.g., $r \approx 0.56 - 0.97$), and Double-fault is consistently negative (e.g., $r \approx -0.60tp - 0.99$), aligning with their interpretations (higher Disagreement / lower Double-fault leads to greater diversity and better accuracy).

**Method sensitivity** DropConnect yields some of the strongest correlations across all datasets, while Dropout is slightly weaker on average. Batch size shows very strong trends for MNIST and K-MNIST and more moderate trends for Fashion-MNIST and CIFAR-10.

**Metric stability/consistency** Q-statistic and entropy are less stable across datasets and regular-

**Table C.4.** Correlation between model test accuracy and diversity across datasets, diversity metrics, and regularization methods (Dropout, DropConnect, and varying batch size). For each dataset–method–metric triplet, diversity is first averaged across all hidden layers per model; Pearson correlation is then computed between test accuracy and this layer-averaged diversity across the full set of models (baseline plus all regularization levels).

| Diversity measure | Dataset | Method | Correlation |
|---|---|---|---|
| Disagreement | MNIST | Dropout | 0.9163 |
| | K-MNIST | Dropout | 0.9322 |
| | Fashion-MNIST | Dropout | 0.6289 |
| | CIFAR-10 | Dropout | 0.6420 |
| | MNIST | Batch size | 0.9737 |
| | K-MNIST | Batch size | 0.8367 |
| | Fashion-MNIST | Batch size | 0.5566 |
| | CIFAR-10 | Batch size | 0.5458 |
| | MNIST | DropConnect | 0.9588 |
| | K-MNIST | DropConnect | 0.9443 |
| | Fashion-MNIST | DropConnect | 0.8888 |
| | CIFAR-10 | DropConnect | 0.8936 |
| Double-fault | MNIST | Dropout | -0.9529 |
| | K-MNIST | Dropout | -0.9134 |
| | Fashion-MNIST | Dropout | -0.8285 |
| | CIFAR-10 | Dropout | -0.6023 |
| | MNIST | Batch size | -0.9852 |
| | K-MNIST | Batch size | -0.9405 |
| | Fashion-MNIST | Batch size | -0.6734 |
| | CIFAR-10 | Batch size | -0.7493 |
| | MNIST | DropConnect | -0.9784 |
| | K-MNIST | DropConnect | -0.9526 |
| | Fashion-MNIST | DropConnect | -0.8970 |
| | CIFAR-10 | DropConnect | -0.7961 |
| Q-statistic | MNIST | Dropout | -0.6095 |
| | K-MNIST | Dropout | 0.8047 |
| | Fashion-MNIST | Dropout | -0.1365 |
| | CIFAR-10 | Dropout | -0.4153 |
| | MNIST | Batch size | -0.9200 |
| | K-MNIST | Batch size | 0.8836 |
| | Fashion-MNIST | Batch size | -0.4831 |
| | CIFAR-10 | Batch size | -0.2076 |
| | MNIST | DropConnect | -0.8727 |
| | K-MNIST | DropConnect | 0.9031 |
| | Fashion-MNIST | DropConnect | -0.7503 |
| | CIFAR-10 | DropConnect | -0.4384 |
| Entropy | MNIST | Dropout | -0.9320 |
| | K-MNIST | Dropout | -0.6390 |
| | Fashion-MNIST | Dropout | -0.7126 |
| | CIFAR-10 | Dropout | 0.3025 |
| | MNIST | Batch size | -0.1802 |
| | K-MNIST | Batch size | -0.6067 |
| | Fashion-MNIST | Batch size | 0.4253 |
| | CIFAR-10 | Batch size | 0.3399 |
| | MNIST | DropConnect | -0.6234 |
| | K-MNIST | DropConnect | -0.6922 |
| | Fashion-MNIST | DropConnect | -0.4859 |
| | CIFAR-10 | DropConnect | 0.3672 |

izers. Q-statistic is negative on MNIST, Fashion-MNIST, and CIFAR-10, but flips positive on K-MNIST. Entropy is mostly negative on MNIST, Fashion-MNIST, and K-MNIST, but positive on CIFAR-10 for all regularizers, also being positive for Fashion-MNIST under the batch size variation (Table C.4).