How Good is a Single Basin?

Kai Lion
Dept of Computer Science
ETH Zürich
kalion@student.ethz.ch

Gregor Bachmann
Dept of Computer Science
ETH Zürich
gregor.bachmann@inf.ethz.ch

Lorenzo Noci
Dept of Computer Science
ETH Zürich
lorenzo.noci@inf.ethz.ch

Thomas Hofmann
Dept of Computer Science
ETH Zürich
thomas.hofmann@inf.ethz.ch

Abstract

The multi-modal nature of neural loss landscapes is often considered to be the main driver behind the empirical success of deep ensembles. In this work, we probe this belief by constructing various "connected" ensembles which are restricted to lie in the same basin. Through our experiments, we demonstrate that increased connectivity indeed negatively impacts performance. However, when incorporating the knowledge from other basins implicitly through distillation, we show that the gap in performance can be mitigated by re-discovering (multi-basin) deep ensembles in a single basin. Thus, we conjecture that while the extra-basin knowledge is at least partially present in any given basin, it cannot be easily harnessed without learning it from other basins.

1 Introduction

Deep neural networks coupled with first order stochastic optimizers give rise to many intriguing charateristics. For instance, two training runs based on different random initializations and batch orderings end up in vastly different "basins" in the landscape, i.e. regions of low loss that are separated by a high barrier (Choromanska et al., 2015). Such minimizers not only differ in location but also in terms of the function they represent; ensembling multiple minimisers strictly improves over the individual performances (Lakshminarayanan et al., 2017). The success of such *deep ensembles* is widely attributed to the described multi-modal nature of the landscape, where functions from different basins exhibit high predictive diversity which in turn leads to better performance.

We challenge this notion by constructing variants of deep ensembles that are constrained to a single basin. We design several variants of varying degrees of connectivity and observe the following: (1) Ensemble performance indeed decreases as the degree of connectivity of the models increases. (2) Leveraging knowledge from other basins through distillation (Hinton et al., 2015) can break this trend, leading to the discovery of very performant single-basin ensembles that almost match the performance of standard deep ensembles. A single basin hence does contain very diverse models; however, they might be difficult to find without additional knowledge. In summary, we make the following contribution:

- We design a rich set of connected ensembles and characterize a trade-off between diversity and connectivity.
- We show that implicitly incorporating knowledge from other basins allows us to design strong connected ensembles that significantly close the gap in performance to deep ensembles. We thus demonstrate that a single basin could suffice for the construction of diverse ensembles.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Setting

We consider image classification problems with a training dataset $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ consisting of n i.i.d tuples of labelled examples $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, K\}$. We study the class of neural network functions $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^K$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$ where $\boldsymbol{\theta}$ denotes the concatenation of all the parameters. We learn $\boldsymbol{\theta}$ through empirical risk minimization $\min_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)$ with $\ell : \mathbb{R}^K \times \{1, \ldots, K\} \to \mathbb{R}^+$ denoting the loss function. To approximately minimize this objective, we use some form of stochastic gradient descent and refer to a minimizer $\boldsymbol{\theta}$ as a *mode*. Such a mode is located in a *loss basin*, referring to the approximately convex region of low loss around it.

Deep Ensembles. We consider $M \in \mathbb{N}$ runs of SGD under different initializations and batch orderings, resulting in a set of minimizers $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$. Due to the non-convexity of neural landscapes, simple convex combinations $\bar{\boldsymbol{\theta}} = \sum_{i=1}^{M} \lambda_i \boldsymbol{\theta}_i$ with $\sum_{i=1}^{M} \lambda_i = 1$ do not constitute minimizers of the test loss. In the literature this is often referred to as lack of (joint) linear mode-connectivity (Garipov et al., 2018). While averaging parameters proves detrimental, averaging the predictions (i.e. ensembling) leads to a substantially more powerful model, i.e. $\bar{f}(\boldsymbol{x}) := \frac{1}{M} \sum_{i=1}^{M} f_{\boldsymbol{\theta}_i}(\boldsymbol{x})$ outperforms any individual model $\boldsymbol{\theta}_i$.

3 Exploring a Single Basin

Connected Ensembles We now explore techniques to replicate the success of deep ensembles, intentionally restricting ourselves to a single basin. In other words, we aim to construct a *connected* ensemble $\{\tilde{\boldsymbol{\theta}}_1, \ldots, \tilde{\boldsymbol{\theta}}_M\}$ that matches the performance of the original ensemble while at the same time guaranteeing linear mode-connectivity. We provide a visualization of the idea in Fig. 1. We focus on *Residual Networks* (ResNets) (He et al., 2016) in the main text, but also evaluate our more competitive methods on *Vision Transformers* (ViTs) (Dosovitskiy et al., 2021) in the Appendix.

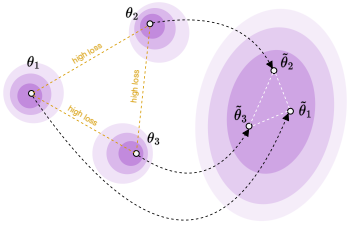


Figure 1: Illustration of a deep ensemble $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$ and a connected ensemble $\{\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, \tilde{\boldsymbol{\theta}}_3\}$.

Connectivity. In order to assess connectivity of a given set of models $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$, we measure

$$\bar{q}(\boldsymbol{\lambda}) = \mathrm{Acc}\left(\sum_{i=1}^{M} \lambda_i \boldsymbol{\theta}_i\right) - \sum_{i=1}^{M} \lambda_i \,\mathrm{Acc}(\boldsymbol{\theta}_i), \qquad \boldsymbol{\lambda} \sim \mathrm{Dir}(\mathbf{1})$$

where $\mathrm{Acc} : \Theta \to [0, 1]$ maps a configuration $\boldsymbol{\theta}$ to its generalization accuracy $\mathrm{Acc}(\boldsymbol{\theta})$. W randomly sample convex combination weights $\boldsymbol{\lambda} \in [0, 1]^M$ from a Dirichlet distribution. We then average multiple draws of $\boldsymbol{\lambda}$ to obtain $\bar{q}$ and say that $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ are jointly linearly-connected if $\bar{q}$ does not decrease significantly below zero.

Stochastic Weight Ensembling (SWE) As a very simple first baseline, we consider a variant of *stochastic weight averaging* (SWA) (Izmailov et al., 2018), where instead of averaging the obtained iterates, we average the predictions, effectively forming an ensemble. More precisely, we train a *ResNet20* for T epochs with a decaying learning rate, producing the first sample $\tilde{\boldsymbol{\theta}}_1$ and then continue training with a constant learning rate, saving a checkpoint $\tilde{\boldsymbol{\theta}}_i$ every T epochs until we collected M samples. This ensures the same computational budget as the deep ensemble. We report the resulting test performance and joint connectivity values in Table 1. We also display the corresponding values for deep ensembles as a reference. We observe that SWE is surprisingly effective, matching the performance of the deep ensemble on CIFAR10 while maintaining a high degree of connectivity. On the more challenging task of CIFAR100 however, we find a significant gap of roughly 3% in test accuracy.

Constrained Ensembles (Con. Ens.) We leverage the insights of Frankle et al. (2020) about the stability of SGD; Along the training trajectory $\{\boldsymbol{\theta}^{(t)} : t \leq T\}$ of SGD, there exists a point $\boldsymbol{\theta}^{(t)}$ after which any subsequently started SGD run with a different batch ordering ends up in the

		Deep Ens.	Con. Ens.	SW Ens.	Perm. Ens.	Dist. Ens.	Dist. Deep Ens.
C-10	Acc	$94.43_{\pm 0.12}$	$94.17_{\pm 0.05}$	$94.0_{\pm 0.18}$	$94.43_{\pm 0.12}$	$94.46_{\pm 0.20}$	$94.45_{\pm 0.02}$
	$\bar{q}$	$-71.74_{\pm 2.38}$	$-0.10_{\pm 0.10}$	$1.48_{\pm 0.04}$	$-25.84_{\pm 4.20}$	$-0.14_{\pm 0.07}$	$-55.7_{\pm 1.71}$
C-100	Acc	$78.15_{\pm 0.10}$	$75.92_{\pm 0.20}$	$74.95_{\pm 0.49}$	$78.15_{\pm 0.10}$	$77.56_{\pm 0.18}$	$78.42_{\pm 0.20}$
	$\bar{q}$	$-68.16_{\pm 1.72}$	$0.86_{\pm 0.18}$	$3.30_{\pm 0.10}$	$-44.89_{\pm 0.91}$	$0.39_{\pm 0.11}$	$-48.32_{\pm 0.15}$
T-IN	Acc	$62.85_{\pm 0.12}$	$59.8_{\pm 0.1}$	$58.36_{\pm 0.6}$	$62.85_{\pm 0.12}$	$62.61_{\pm 0.43}$	$63.29_{\pm 0.33}$
	$\bar{q}$	$-53.78_{\pm 0.85}$	$0.75_{\pm 0.10}$	$2.80_{\pm 0.14}$	$-46.30_{\pm 2.08}$	$-1.35_{\pm 0.48}$	$-35.79_{\pm 0.77}$

Table 1: Comparison of ResNet20 ensemble accuracy (Acc) and connectivity $\bar{q}$ (in percent) for all ensemble variants on CIFAR10 (C-10), CIFAR100 (C-100), and Tiny ImageNet (T-IN).

same loss basin. Surprisingly, this time point t can be as early as a few epochs in training. This offers a recipe for the following family of connected ensembles; (1) Train a model up to time t, (2) use $\boldsymbol{\theta}^{(t)}$ as a starting point for M runs of SGD, and (3) continue training for $T - t$ epochs with different batch orderings, leading to a *constrained* ensemble $\{\tilde{\boldsymbol{\theta}}_1(t), \dots, \tilde{\boldsymbol{\theta}}_M(t)\}$. Again, we use the same computational budget as a deep ensemble. We notice that the time parameter t intuitively balances diversity and connectivity; a smaller t yields more diverse but less connected solutions, while conversely, a larger t leads to greater connectivity at the expense of lower diversity. We display the resulting performance and connectivity results in Table 1. We obtain a very similar picture as for SWE, i.e. constrained ensembles also match the performance on CIFAR10, offer strong connectivity, but fall short on CIFAR100, albeit with a significant improvement.

4 Re-discovering Deep Ensembles in a Single Basin

Our preliminary results lead us to conclude that discovering a connected deep ensemble with matching performance is a challenging endeavour. We thus take a step back and revisit our research question from a slightly different angle:

If access to a deep ensemble was granted, could one re-discover it in a single basin?

This is conceptually a simplified goal as knowledge of other basins can now be leveraged to guide the search within a single basin. A positive answer however would still be very impactful as it proves the existence of a connected deep ensemble, motivating further research into efficient exploration of a single basin. In the following approaches, we will thus assume that we have access to a deep ensemble $\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M\}$.

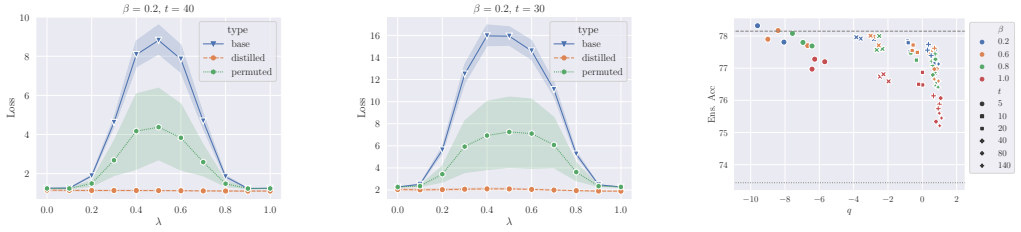
Permuted Ensembles (Perm. Ens.) We first investigate the PERMUTATIONCOORDINATEDESCENT (PCD) algorithm from Ainsworth et al. (2023). We choose the first member $\boldsymbol{\theta}_1$ as a reference model and we aim to apply permutations $\boldsymbol{\pi}_i$ to each remaining member $\boldsymbol{\theta}_i$ such that $\boldsymbol{\theta}_1$ and $\boldsymbol{\pi}_i(\boldsymbol{\theta}_i)$ live in the same basin. Such a permutation is discovered by aligning the weights of each member with the reference model, we refer to Ainsworth et al. (2023) for more details. Since permutations constitute a symmetry of neural networks, the performance of the new members $\boldsymbol{\pi}_i(\boldsymbol{\theta}_i)$ remains unchanged, and we thus have a mathematical guarantee to achieve the same accuracy as the original ensemble. But this guarantee comes at a cost; the degree of pairwise connectivity between two permuted members $\boldsymbol{\pi}_i(\boldsymbol{\theta}_i)$ and $\boldsymbol{\pi}_j(\boldsymbol{\theta}_j)$ can vary significantly, as illustrated by the wide confidence intervals in Fig. 2a and 2b. Similarly, joint connectivity is also violated as shown in Table 1. This is not surprising as the objective only optimizes for pairwise alignment. We further show in Appendix A that optimising for "joint" alignment of models does not achieve joint connectivity either.

Distilled Ensembles. (Dist. Ens.) In this approach, we combine our insights from *constrained* ensembles with the mechanism of model distillation, as introduced by Hinton et al. (2015). Again denote by $\boldsymbol{\theta}_1^{(t)}$ the stability point of SGD for the reference model $\boldsymbol{\theta}_1$. We aim to re-discover the j-th member $\boldsymbol{\theta}_j$ in the same basin as $\boldsymbol{\theta}_0$ by minimizing a distillation objective towards $\boldsymbol{\theta}_j$, i.e. we

minimize

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} (1-\beta) \cdot \tau^2 \cdot \mathrm{KL}\left(\sigma\left(\frac{f_{\boldsymbol{\theta}_j}(\boldsymbol{x}_i)}{\tau}\right), \sigma\left(\frac{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}{\tau}\right)\right) - \beta \log\left(\left[\sigma\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right)\right]_{y_i}\right) \quad (1)$$

where β and τ are additional hyperparameters and σ is the softmax function. We then start the



(a) CIFAR100 (b) Tiny ImageNet (c) Connectivity vs test accuracy

Figure 2: Pairwise connectivity for ResNet20 ensembles in (a) and (b). Averaged over five randomly selected pairs and repeated for three random seeds, totalling 15 pairs. Connectivity vs test accuracy in (c). The dashed horizontal line shows the accuracy of a deep ensemble, while the dotted horizontal line shows the mean member accuracy.

optimization from $\boldsymbol{\theta}_1^{(t)}$ to encourage connectivity of solutions and denote the minimizers of Eq. 1 by $\tilde{\boldsymbol{\theta}}_{1 \to j}$. β trades-off the optimization towards matching the ground truth and functional similarity to the j-th member. Note that for $\beta = 1$, the approach essentially reduces to *constrained* ensembles. $\tau > 1$ is a temperature scaling parameter used to facilitate the knowledge transfer to the student model. Table 1 illustrates that our distillation strategy with $\beta = 0.2$ produces very competitive ensembles for residual models, significantly closing the gap to standard deep ensembles across all datasets. Moreover, such a distilled ensemble exhibits a surprisingly strong degree of connectivity, not only fulfilling pairwise connectivity (see Fig. 2), but also the more challenging joint linear connectivity property $\bar{q}$, as shown in Table 1.

Connectivity vs accuracy. In Fig. 2c we show test accuracy as a function of connectivity $\bar{q}$. Without distillation (represented by the red markers), we see that increased connectivity negatively impacts performance. However, as soon as we employ distillation (blue markers), we manage to significantly mitigate the drop in performance without compromising connectivity. We provide further experiments on the impact of connectivity on performance and diversity in Appendix A.

Regularizing effect of distillation. Zhang et al. (2019) demonstrated that distillation can enhance student performance beyond that of its teacher. Thus, the improvement of distilled ensembles over constrained ensembles observed in Table 1 might be due by this regularizing effect of distillation. To isolate this effect, we consider the baseline of deep ensembles trained with the same distillation objective from Eq. 1. If the gain in ensemble performance observed for distilled ensembles would be primarily due to the regularizing effect of distillation rather than the incorporation of out-of-basin knowledge, then it is reasonable to expect similar improvements in ensemble performance when adding the distillation term to deep ensembles. As illustrated in Table 4 in Appendix A, distillation does not significantly improve ensemble performance for ordinary deep ensembles, highlighting that the gains are unlikely to be caused by the regularizing effect of distillation.

5 Discussion

In this work, we explored various approaches to construct ensembles constrained to lie in a single basin. We observe that constructing such a connected ensemble without any knowledge from other basins proves to be difficult and a significant gap to deep ensembles remains. Moreover, we observe a pronounced trade-off between (joint) linear mode-connectivity and the resulting ensemble performance. However, when incorporating other basins implicitly through a distillation procedure we manage to break this trade-off and strongly reduce this gap, producing connected ensembles that are (almost) on-par with deep ensembles. While relying on other basins renders our approach

very inefficient, it nevertheless demonstrates the existence of very performant ensembles in a single basin, requiring us to rethink the characteristics of loss landscapes. The existence of strong connected ensembles illustrates that, in principle, the functional diversity within a single basin is sufficient to achieve predictive performance that is comparable to an ensemble sampled from different modes. In other words, our results illustrate that escaping the basin is not a prerequisite for attaining competitive prediction accuracy. We hope that our insights can guide future work towards designing algorithms that more thoroughly and efficiently explore a single basin.

References

Abe, T., Buchanan, E. K., Pleiss, G., and Cunningham, J. P. (2023). Pathologies of Predictive Diversity in Deep Ensembles. arXiv:2302.00704 [cs, stat].

Ainsworth, S., Hayase, J., and Srinivasa, S. (2023). Git Re-Basin: Merging Models modulo Permutation Symmetries. In *The Eleventh International Conference on Learning Representations*.

Ba, L. J., Kiros, J. R., and Hinton, G. E. (2016). Layer Normalization. *CoRR*, abs/1607.06450. arXiv: 1607.06450.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2015). The Loss Surfaces of Multilayer Networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 192–204. PMLR. ISSN: 1938-7228.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. (2018). Essentially No Barriers in Neural Network Energy Landscape. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1309–1318. PMLR. ISSN: 2640-3498.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. (2020). Linear Mode Connectivity and the Lottery Ticket Hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3259–3269. PMLR. ISSN: 2640-3498.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. (2018). Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Goodfellow, I., Vinyals, O., and Saxe, A. (2015). Qualitatively Characterizing Neural Network Optimization Problems. In *International Conference on Learning Representations*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. ISSN: 1063-6919.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [cs, stat].

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. (2017). Snapshot Ensembles: Train 1, Get M for Free. In *International Conference on Learning Representations*.

Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456. PMLR. ISSN: 1938-7228.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2018). Averaging Weights Leads to Wider Optima and Better Generalization. In Globerson, A. and Silva, R., editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 876–885.

Juneja, J., Bansal, R., Cho, K., Sedoc, J., and Saphra, N. (2023). Linear Connectivity Reveals Generalization Strategies. arXiv:2205.12411 [cs].

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. *University of Toronto*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Le, Y. and Yang, X. (2015). Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3.

Lippe, P. (2022). UvA Deep Learning Tutorials.

Lucas, J. R., Bae, J., Zhang, M. R., Fort, S., Zemel, R., and Grosse, R. B. (2021). On Monotonic Linear Interpolation of Neural Network Parameters. In *Proceedings of the 38th International Conference on Machine Learning*, pages 7168–7179. PMLR. ISSN: 2640-3498.

Neyshabur, B., Sedghi, H., and Zhang, C. (2020). What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems*, volume 33, pages 512–523. Curran Associates, Inc.

Singh, S. P. and Jaggi, M. (2020). Model Fusion via Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 33, pages 22045–22055. Curran Associates, Inc.

Wortsman, M., Horton, M. C., Guestrin, C., Farhadi, A., and Rastegari, M. (2021). Learning Neural Network Subspaces. In *Proceedings of the 38th International Conference on Machine Learning*, pages 11217–11227. PMLR. ISSN: 2640-3498.

Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., and Ma, K. (2019). Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722.

A Additional Experiments

Does it also work for ViTs? In Table 2, we evaluate our distillation method on ViTs. As is the case for ResNets, the inclusion of the distillation term boosts ensemble performance without compromising connectivity.

		$\beta = 1.0$			$\beta = 0.2$		
		$\bar{q}_{\text{joint}}$	Mean Acc	Ens. Acc	$\bar{q}_{\text{joint}}$	Mean Acc	Ens. Acc
CIFAR10	ResNet20	$-0.14_{\pm 0.07}$	$93.15_{\pm 0.03}$	$94.17_{\pm 0.05}$	$-0.64_{\pm 0.11}$	$93.67_{\pm 0.12}$	$94.46_{\pm 0.20}$
	ViT	$-1.37_{\pm 0.41}$	$82.60_{\pm 0.02}$	$84.28_{\pm 0.23}$	$-1.49_{\pm 0.25}$	$83.14_{\pm 0.13}$	$84.55_{\pm 0.40}$
CIFAR100	ResNet20	$0.86_{\pm 0.18}$	$73.53_{\pm 0.23}$	$75.92_{\pm 0.20}$	$0.39_{\pm 0.11}$	$75.33_{\pm 0.12}$	$77.56_{\pm 0.18}$
	ViT	$-0.14_{\pm 0.08}$	$54.90_{\pm 0.26}$	$57.81_{\pm 0.29}$	$-0.29_{\pm 0.33}$	$56.12_{\pm 0.10}$	$58.70_{\pm 0.15}$
Tiny ImageNet	ResNet20	$0.75_{\pm 0.10}$	$55.80_{\pm 0.19}$	$59.83_{\pm 0.13}$	$-1.35_{\pm 0.48}$	$58.69_{\pm 0.17}$	$62.61_{\pm 0.43}$
	ViT	$1.76_{\pm 0.12}$	$35.36_{\pm 0.30}$	$39.50_{\pm 0.21}$	$1.57_{\pm 0.18}$	$38.46_{\pm 0.07}$	$42.31_{\pm 0.09}$

Table 2: Comparison of joint connectivity and ensemble performance for constrained ($\beta = 1.0$) and distilled ensembles ($\beta = 0.2$). Averaged over 3 seeds.

Jointly permuted ensembles. We now evaluate whether the lack of joint connectivity observed for permuted ensembles (see Table 1) can be diminished by extending the optimization objective used in PCD. More specifically, we change the objective function used in Ainsworth et al. (2023) to account for the joint alignment with respect to all other models and not just the reference model. Thus, when optimizing $\boldsymbol{\pi}_i(\boldsymbol{\theta}_i)$ we account for the alignment with respect to all other models $\boldsymbol{\pi}_j(\boldsymbol{\theta}_j)$ with $j \neq i$ in the ensemble. Using this modified objective and wrapping the pairwise procedure with another layer iterating over ensemble members, we obtain an algorithm that optimizes for joint alignment and to which we refer to as Multi-PCD. While joint connectivity does improve, the resulting ensemble is still far from being connected as measured by $\bar{q}_{\text{joint}}$ in Table 3. We thus conclude that permutations can not be leveraged to re-discover an ordinary multi-basin ensemble in a single loss basin.

	Deep Ens.	PCD	Multi-PCD
CIFAR10	$-71.74_{\pm 2.38}$	$-25.84_{\pm 4.20}$	$-14.64_{\pm 3.66}$
CIFAR100	$-68.16_{\pm 1.72}$	$-44.89_{\pm 0.91}$	$-41.02_{\pm 1.55}$
Tiny ImageNet	$-53.78_{\pm 0.85}$	$-46.30_{\pm 2.08}$	$-44.54_{\pm 2.65}$

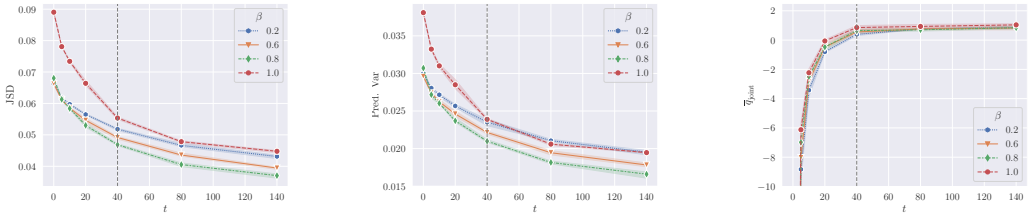
Table 3: Joint connectivity $\bar{q}_{\text{joint}}$ of deep ensembles and permuted ensembles optimizing for pairwise (PCD) and joint alignment (Multi-PCD). Averaged over 3 seeds.

Diversity-Connectivity trade-off. In Fig. 3, we plot two measures of predictive diversity used in Abe et al. (2023) and connectivity as a function of t for a grid of β values. In Fig. 3a, we show the one-vs-all Jensen-Shannon divergence of predictions and in Fig. 3b we show the variance of the ensemble members' true-class predictions. For more detailed information, we refer to Abe et al. (2023). Notably, we observe a *diversity-connectivity trade-off*, as diversity decreases with higher connectivity.

Regularizing effect of distillation. As described in the main text, we also consider a baseline of deep ensembles trained with an additional distillation loss. We report the results in Table 4 and note that we do not observe any significant improvements through the inclusion of a distillation objective, corroborating the findings from the main text.

		Deep Ens.			Deep Ens. + $\beta = 0.2$		
		$\bar{q}_{\text{joint}}$	Mean Acc	Ens. Acc	$\bar{q}_{\text{joint}}$	Mean Acc	Ens. Acc
CIFAR10	ResNet20	$-71.74_{\pm 2.38}$	$93.01_{\pm 0.08}$	$94.43_{\pm 0.12}$	$-71.30_{\pm 3.01}$	$93.54_{\pm 0.04}$	$94.45_{\pm 0.02}$
	ViT	$-55.81_{\pm 1.99}$	$82.43_{\pm 0.33}$	$85.10_{\pm 0.27}$	$-55.70_{\pm 1.71}$	$82.97_{\pm 0.22}$	$84.87_{\pm 0.31}$
CIFAR100	ResNet20	$-68.16_{\pm 1.72}$	$73.44_{\pm 0.12}$	$78.15_{\pm 0.10}$	$-69.03_{\pm 2.19}$	$75.20_{\pm 0.15}$	$78.42_{\pm 0.20}$
	ViT	$-47.28_{\pm 0.19}$	$54.91_{\pm 0.10}$	$59.88_{\pm 0.12}$	$-48.32_{\pm 0.15}$	$56.20_{\pm 0.08}$	$59.92_{\pm 0.26}$
Tiny ImageNet	ResNet20	$-53.78_{\pm 0.85}$	$55.36_{\pm 0.33}$	$62.85_{\pm 0.20}$	$-56.54_{\pm 0.70}$	$58.65_{\pm 0.23}$	$63.29_{\pm 0.33}$
	ViT	$-33.04_{\pm 0.70}$	$35.57_{\pm 0.38}$	$44.05_{\pm 0.19}$	$-35.79_{\pm 0.77}$	$38.37_{\pm 0.31}$	$44.29_{\pm 0.21}$

Table 4: Isolating the additional regularizing effect of distillation. Averaged over 3 seeds.



(a) Jensen-Shannon Divergence (b) Predictive Variance (c) $\bar{q}_{\text{joint}}$

Figure 3: Predictive variance, Jensen-Shannon divergence, and joint connectivity as a function of time parameter t for ResNet20 ensembles on CIFAR100. The dashed vertical lines mark the t used in Table 1.

B Related Works

Ensembling techniques. There is a plethora of previous work that studies novel ensembling techniques, often with a focus on reduced cost or better weight averaging properties. Fast Geometric Ensembling (FGE) from Garipov et al. (2018) and Snapshot Ensembles (SSE) from Huang et al. (2017) both adapt a similar strategy as the SWE approach but use a cyclical learning rate to intentionally break connectivity and produce more efficient ensembles. Instead of ensembling models, Izmailov et al. (2018) average weights along the SGD trajectory using a cyclical or constant learning rate. Wortsman et al. (2021) on the other hand directly learn lines and curves whose endpoints they leverage for ensembling. They also report improved performance when using the midpoint as a summary of the ensemble. Another related line of work studies fusion of several independent models. Singh and Jaggi (2020) leverage optimal transport to align the weights of multiple models and produce a fused endpoint. Ainsworth et al. (2023) take a similar approach and fuse different networks by finding fitting permutations to maximize similarity.

Combining SSE, FGE, and SWA. We decided to use a procedure that combines elements from SSE, FGE, and SWA as a baseline. We argue that this approach is most effective at training an ensemble while ensuring linear mode connectivity and computational comparability, at training and inference time, with deep ensembles. As outlined in the main text, we refer to this method as Stochastic Weight Ensembling (SWE). More specifically, SWE is ensembling models in function space, acquiring them using a sequential procedure. We first decay the learning rate to a level that enables exploration of the basin without leaving it, and keep the learning rate constant thereafter. We sample a model every T epochs where T is on the order of epochs required to train a single model. The difference to SSE is that we specifically do *not* encourage exploration of different basins and thus refrain from cyclically increasing the learning rate. The procedure is also different from SWA, as we do not average in weight space, but in function space. Lastly, it is also different from FGE, as the cycle length is comparable to that of SSE, ruling out the *fast* in FGE.

Mode Connectivity. An intellectual ancestor to linear mode connectivity can be seen in the work of (Goodfellow et al., 2015). They consider the 1D subspace spanned by the initial and fully trained parameter vectors and find that the loss is monotonically decreasing the closer we get to the final parameter vector. (Lucas et al., 2021) confirmed these results and coined the phenomenon *monotonic linear interpolation*. In the context of our work, we interpret this monotonic linear interpolation phenomenon as a descent into a loss basin whose functional diversity we aim to explore. Frankle et al. (2020) demonstrated that there is a point in training $\boldsymbol{\theta}^{(t)}$ after which SGD runs sharing $\boldsymbol{\theta}^{(t)}$ as initialization remain linearly mode connected. Neyshabur et al. (2020) observed linear mode connectivity in a transfer learning setup, where models pre-trained on a source task remain linearly mode connected after training on the downstream task. Juneja et al. (2023) provide counterexamples to mode connectivity outside of image classification tasks. Draxler et al. (2018); Garipov et al. (2018) found non-linear paths of low loss between independently trained models, questioning the idea that the loss landscape is composed of isolated minima.

Diversity. As mentioned in the introduction, it is commonly believed that encouraging predictive diversity is a prerequisite for improving ensemble performance. This belief is derived from classical results in statistics on bagging and boosting weak learners (Freund et al., 1999; Breiman, 1996). While it is true that disagreement among members is a necessary condition for an ensemble to outperform any single member, recent work has shown that encouraging predictive diversity can be detrimental to the performance of deep ensembles with high-capacity members (Abe et al., 2023). In other words, the intuition from those classical results might not be applicable. The counter-intuitive observation of Abe et al. (2023) is explained by the fact that diversity encouraging penalties affect all predictions irrespective of their correctness. As a result, these penalties can adversely affect the performance of individual members, which in turn can undermine the performance of the ensemble.

C Implementation Details

Computational Cost If not stated otherwise, we consider ensembles of size $M = 5$. The table below illustrates the computational cost on a per model basis.

		Deep Ens.	SWE	Distilled Ens.				Constrained Ens.			
		T	T	β	T	t	Dist. Epochs	β	T	t	Dist. Epochs
CIFAR10	ResNet20	110	110	0.2	110	10	100	1.0	110	10	100
	ViT	165	—	0.2	165	15	150	1.0	165	15	150
CIFAR100	ResNet20	190	190	0.2	190	40	150	1.0	190	40	150
	ViT	165	—	0.2	165	15	150	1.0	165	15	150
Tiny ImageNet	ResNet20	130	130	0.2	130	30	100	1.0	130	30	100
	ViT	140	—	0.2	140	15	125	1.0	140	15	125

Table 5: Comparison of computational cost for different experiments in the main text. For deep ensembles T refers to the number of epochs per sample. Similarly, for SWE, T is the cycle length in-between taking a sample. For constrained and distilled ensembles, t is the epoch after which we split the runs and starting distilling for Dist. Epochs.

Optimizers With the exception of experiments conducted with ViTs, we use SGD as an optimizer with a peak learning rate of 0.1. We use a cosine decay schedule with linear warmup for the first 10% of training. Momentum is set to 0.9. For ViTs, we use Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size is at 128 and we set the temperature in the distillation experiments to $\tau = 3$. For SWE, we apply the same linear warmup cosine decay schedule as for the other ensemble methods, but stop decaying the learning rate at 0.01 and hold it constant thereafter to enable exploration of the basin.

Datasets We experiment with the classic image classification baselines CIFAR (Krizhevsky, 2009) and Tiny ImageNet (Le and Yang, 2015). For all experiments, we make use of data augmentation. More specifically, we use horizontal flips, random crops, and color jittering.

Architectures We use the ResNet20 implementation from Ainsworth et al. (2023) with three blocks of 64, 128, and 256 channels, respectively. We note that this implementation uses LayerNorm (Ba et al., 2016) instead of BatchNorm (Ioffe and Szegedy, 2015), as it eliminates the burden of recalibrating the BatchNorm statistics when interpolating between networks. Our Vision Transformer implementation is based on Lippe (2022) and composed of six attention layers with eight heads, latent vector size of 256 and hidden dimensionality of 512. We apply it to flattened 4×4 image patches.

Permuted Ensembles We use the PERMUTATIONCOORDINATEDESCENT implementation from Ainsworth et al. (2023) to bring deep ensemble models into alignment. The implementation of the PERMUTATIONCOORDINATEDESCENT algorithm can be found at `https://github.com/samuela/git-re-basin`.

Joint Connectivity As mentioned in the main text, we draw samples $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N \sim \text{Dir}(\mathbf{1})$ to approximately assess the joint connectivity of ensemble members. For each seed, we evaluate $N = 50$ samples and compute $\bar{q}_{\text{joint}} = \frac{1}{N} \sum_{i=1}^{N} q_{\text{joint}}(\boldsymbol{\lambda}_i)$

Hardware We ran experiments on a cluster with NVIDIA GeForce RTX 2080 Ti and NVIDIA GeForce RTX 3090 GPUs.