How Good is a Single Basin?

Kai Lion
ETH Zurich

Lorenzo Noci
ETH Zurich

Thomas Hofmann
ETH Zurich

Gregor Bachmann
ETH Zurich

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 within 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

The intricate characteristics of neural loss landscapes pose one of the most intriguing puzzles in deep learning. The surfaces are characterized by their highly non-convex nature, giving rise to an exponential number of minima belonging to various *modes* or *basins*. Simple, first-order optimizers such as stochastic gradient descent are then used to navigate these landscapes, leading to a complex interplay between stochasticity and non-convexity. As a consequence, a plethora of interesting behaviours are observed empirically: (1) models trained under different random initializations and batch orderings end up in different parts of the weight space while maintaining similar generalization performance (Choromanska et al., 2015), (2) these independently trained models can be connected by simple but non-linear curves (Draxler et al., 2018; Garipov et al., 2018) in parameter space, and (3) ensembling these leads to an increase in performance (Lakshminarayanan et al., 2017). The gain in performance can be attributed to the predictive diversity of the obtained models (akin to the classic bias-variance tradeoff), i.e. different runs of the same model result in "different interpretations of the data" due to the non-convexity and the stochasticity of the learning process. The idea of distinguishing between models from different basins is essential and has been used as a synonym for higher model diversity. The common belief is that *diversity is achieved by visiting different basins in the loss landscape.* This belief suggests that ensembling models from the same basin is of limited use, as models are not sufficiently diverse. Indeed, many works use this interpretation of diversity as a design principle for their inference algorithms (Huang et al., 2017; D'Angelo and Fortuin, 2021; Loshchilov and Hutter, 2017; Zhang et al., 2020).

In this work, we take the opposite direction and investigate instead how performant ensembles can be if they are restricted to lie in a single basin. To that end, we design a set of ensemble baselines that increasingly manage to close the gap to deep ensembles in terms of test accuracy with the additional constraint that each ensemble member lies in the same basin. We coin such ensembles *connected ensembles* alluding to the fact that models from the same basin are linearly mode-connected. Our more intuitive approaches for constructing connected ensembles suggest that increased connectivity is often associated with lower performance. However, by leveraging the insights of Frankle et al. (2020) regarding the stability of SGD and adopting the distillation procedure introduced by Hinton et al. (2015), this relationship can be disrupted. Connected ensembles constructed using this distillation method show significantly improved predictive performance. We further demonstrate the broad applicability of the connected ensemble class on two model families and three widely used benchmarks. More specifically, we consider ensembles of *ResNets* (He et al., 2016) and the more recent *Vision Transformers* (Dosovitskiy et al., 2021).

These results have strong implications for the design of learning algorithms that leverage multiple model runs, demonstrating that in principle, one does not need to break linear mode-connectivity to find diverse models but can rather ensemble models that are within the same basin. From a theoretical point of view, our results on distilled ensembles imply that the information of multiple basins can be re-discovered within a single basin to a large extent. However, more work is necessary to assess whether this knowledge can be leveraged without the explicit aid of other basins. More specifically, we make the following contributions:

- 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.

The structure of this paper is as follows. In Section 2, we introduce the background and motivate the research question. In Section 3, we describe methods to explore a single loss basin. We expand on those results by further enhancing the single-basin ensemble performance with different techniques in Section 4. We provide a review of related work in Section 5 and discuss implications of our empirical findings in Section 6.

2 BACKGROUND

2.1 Mode-Exploring Ensembling Techniques

In deep ensembles (Lakshminarayanan et al., 2017), each model is trained on the same dataset by using different random initializations and batch orderings. As a result, the ensemble members reliably converge to different loss basins in weight-space, which makes this procedure highly effective at exploring the non-convex, multi-modal loss landscape of neural networks. Naturally, the success of deep ensembles in terms of predictive accuracy, uncertainty quantification and out-of-distribution robustness (Ovadia et al., 2019; Wenzel et al., 2020) is commonly attributed to this ability to sample from different modes that contain functionally diverse models (Fort et al., 2020). Consequently, researchers have developed sequential ensembling methods designed to mimic the mode-sampling ability of deep ensembles. In particular, methods that leverage a cyclical (sometimes constant) learning rate schedule are supposed to explore different parts of the weight-space

by cyclically increasing the learning rate and decreasing it again before sampling a model (Loshchilov and Hutter, 2017). Further examples are cyclical SGM-CMC (Zhang et al., 2020), as well as Snapshot Ensembles (Huang et al., 2017), Fast Geometric Ensembling (Garipov et al., 2018), and Stochastic Weight Averaging (Izmailov et al., 2018). D'Angelo and Fortuin (2021) study ensembles with repulsive terms in weight and function space to prevent collapse to a single mode.

Frankle et al. (2020) observed that there is a point early in training after which SGD runs with different batch orderings and augmentations become *linearly mode-connected*, allowing to interpolate linearly between different modes without incurring a high loss. Two models which are linearly mode-connected can be thought of as being located in the same loss basin, which, from a classical point of view, can be detrimental for the predictive diversity of an ensemble. Hence, the procedures described sequentially train models and sample from different modes by traversing regions of high loss to escape the basin and break *linear mode-connectivity*.

2.2 Permutation Hypothesis

The permutation hypothesis (Entezari et al., 2022), which recently has attracted much attention, posits that, for wide enough networks, the solutions obtained through independent SGD runs become linearly mode-connected if their weights are permuted appropriately. Ainsworth et al. (2023) provided empirical evidence for the permutation hypothesis by proposing weight and activation matching algorithms that bring neurons into alignment without altering the underlying function. Given two models found by SGD, their method finds a permutation of weights such that the two models become linearly mode connectable, i.e., lie in the same loss basin.

The permutation hypothesis implies that the loss landscape can be collapsed into a single loss basin if we account for redundant basins that are due to permutation symmetries. This implication has significant consequences for mode-exploring ensembling techniques. The underlying assumption of such ensembling techniques is that predictive diversity can only be achieved by virtue of visiting different modes in weight-space. The permutation hypothesis, however, implies that there is nothing to be gained from visiting different modes beyond what can already be found in a single basin. In other words, visiting different basins might not be necessary to obtain models that constitute a strong ensemble.

Given these new insights into the redundancies of the loss landscape and their potential implications for deep ensembles, we depart from the dominant paradigm

of exploring multiple modes and instead intentionally limit ourselves to a single mode to form a connected ensemble. By doing so, we aim to provide an answer to the following question:

How Good Can a Single-Basin Ensemble Be?

3 EXPLORING A SINGLE BASIN

3.1 Setting

Setup. 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. Note that many modern neural network parameterizations contain redundancies in the weight-space in the sense that two parameter vectors can encode the same function, i.e., $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$ could yield $f_{\boldsymbol{\theta}_1}(\boldsymbol{x}) = f_{\boldsymbol{\theta}_2}(\boldsymbol{x}) \,\forall \boldsymbol{x} \in \mathbb{R}^d$. In our work, we limit ourselves to the set of symmetries induced by permutations[1]. 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. We use $\mathrm{KL}(\cdot, \cdot)$ to denote the KL-Divergence between two probability distributions over the discrete space of classification labels and denote the softmax function with $\sigma : \mathbb{R}^K \to (0, 1)^K$.

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 $\boldsymbol{\theta}_i$. We demonstrate this effect on the classic image classification benchmarks CIFAR (Krizhevsky, 2009) and Tiny ImageNet (Le and Yang, 2015) using a ResNet20 (He et al., 2016) ensemble of size $M = 5$ in Table 1. While the increase in test accuracy is moderate for CIFAR10, we observe

---

[1]We note that there are other symmetries, such as scaling or sign-flip symmetries, that might cause this phenomenon. However, these other symmetries are beyond the scope of this work.
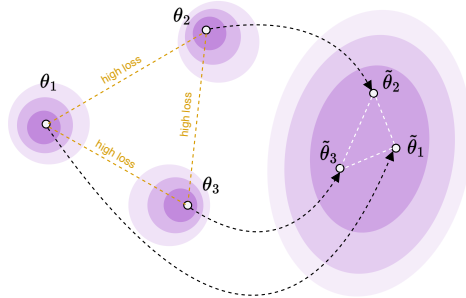

Figure 1: Illustration of toy deep ensemble $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$ and the matching, connected ensemble $\{\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, \tilde{\boldsymbol{\theta}}_3\}$.

Dataset	Mean Acc	Ens. Acc
CIFAR10	$93.01_{\pm 0.08}$	$94.43_{\pm 0.12}$
CIFAR100	$73.44_{\pm 0.12}$	$78.15_{\pm 0.10}$
Tiny ImageNet	$55.36_{\pm 0.33}$	$62.85_{\pm 0.20}$

Table 1: Average member test accuracy $\frac{1}{5} \sum_{i=1}^5 \mathrm{Acc}(\boldsymbol{\theta}_i)$ and deep ensemble performance of ResNet20. Averaged over 3 seeds.

very strong improvements for the more complex tasks CIFAR100 and Tiny ImageNet. We remark that a deep ensemble of size M incurs a computational cost of $C_{\mathrm{DE}} = M \times T$ where T is the number of training epochs used for a single model. We ensure that our baselines match in terms of computational cost in order to guarantee that no improvement is achieved simply due to longer training. We defer implementation details such as values for T to Appendix C.

3.2 Finding Connected Deep Ensembles

We now explore various methods to replicate the success of deep ensembles while intentionally limiting ourselves to only leveraging a single basin. In other words, we aim to construct a *connected* ensemble $\{\tilde{\boldsymbol{\theta}}_1, \ldots, \tilde{\boldsymbol{\theta}}_M\}$ that approximates 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. While for simple baselines we focus on *Residual Networks* (ResNets) (He et al., 2016), we also evaluate our more competitive methods on *Vision Transformers* (ViTs) (Dosovitskiy et al., 2021). Note that we do not consider pre-trained ViTs, as our experiments heavily depend on whether models share a pre-trained initialization. We largely focus on test accuracy, cross-entropy loss, connectivity, and diversity as the main metrics of comparison.

Connectivity. In order to assess the connectivity of a given set of models $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$, we perform several tests. Firstly, we study the pairwise connectivity of any two sets of models, i.e. we evaluate

$$q_{\text{pair}}(\lambda) = \text{Acc}(\lambda \boldsymbol{\theta}_i + (1-\lambda)\boldsymbol{\theta}_j) \\ - (\lambda \text{Acc}(\boldsymbol{\theta}_i) + (1-\lambda) \text{Acc}(\boldsymbol{\theta}_j)) \quad (1)$$

for $\lambda \in [0, 1]$, where $\text{Acc} : \Theta \to [0, 1]$ maps a configuration $\boldsymbol{\theta}$ to its generalization accuracy $\text{Acc}(\boldsymbol{\theta})$. If q_{pair} does not decrease significantly, we say that $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ are pairwise linearly-connected. We further visualize the test accuracy as a function of λ (see e.g. Fig. 2) and inspired by Garipov et al. (2018), also produce two-dimensional plots visualizing the parameter-plane spanned by three models (see e.g. Fig. 4). In order to assess joint connectivity, we measure

$$q_{\text{joint}}(\boldsymbol{\lambda}) = \text{Acc}\left(\sum_{i=1}^{M} \lambda_i \boldsymbol{\theta}_i\right) - \sum_{i=1}^{M} \lambda_i \text{Acc}(\boldsymbol{\theta}_i), \ \boldsymbol{\lambda} \sim \text{Dir}(\mathbf{1}) \quad (2)$$

While we could rely on a densely sampled grid for λ in the pairwise case, for the joint connectivity we resort to randomly sampling convex combination weights $\boldsymbol{\lambda} \in [0, 1]^M$ from a Dirichlet distribution. We then average multiple draws of $\boldsymbol{\lambda}$ to obtain $\bar{q}_{\text{joint}}$ and say that $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ are jointly linearly-connected if $\bar{q}_{\text{joint}}$ does not decrease significantly.

Diversity. To evaluate ensemble diversity, we consider two metrics used by Abe et al. (2023). More specifically, we consider the variance over ensemble members' true-class predictions $\text{Var}(\{\boldsymbol{\theta}_i\}_{i=1}^{M})$ and the one-vs-all Jensen-Shannon-Divergence $\text{JSD}(\{\boldsymbol{\theta}_i\}_{i=1}^{M})$. For more details on those metrics, we refer to Abe et al. (2023).

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. Similarly to Izmailov et al. (2018), we use a decaying learning rate schedule that converges to a fixed value, enabling exploration of the basin without leaving it. 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 highlight that this approach is very similar to Snapshot Ensembles (Huang et al., 2017), but instead of encouraging explorations of different basins by using

cyclical learning rates, we ensure connectivity by using a constant learning rate. We report the resulting test performance and joint connectivity values in Table 2. 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. We leverage the insights of Frankle et al. (2020) regarding 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 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 for $T - t$ epochs with different batch orderings, leading to a *constrained ensemble* $\{\tilde{\boldsymbol{\theta}}_1(t), \ldots, \tilde{\boldsymbol{\theta}}_M(t)\}$. Again we use the same computational budget as a deep ensemble. We notice that the time parameter t intuitively trades off diversity and connectivity; the smaller t is, the more diverse but less connected are the solutions and similarly for large t. We display the resulting performance and connectivity results in Table 2. 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 $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$. We emphasize that the purpose of this investigation is *not*

	Deep Ensemble		Constrained Ensemble		SWE	
	$\bar{q}_{\text{joint}}$	Ens. Acc	$\bar{q}_{\text{joint}}$	Ens. Acc	$\bar{q}_{\text{joint}}$	Ens. Acc
CIFAR10	$-71.74_{\pm 2.38}$	$94.43_{\pm 0.12}$	$-0.14_{\pm 0.07}$	$94.17_{\pm 0.05}$	$1.48_{\pm 0.04}$	$94.00_{\pm 0.18}$
CIFAR100	$-68.16_{\pm 1.72}$	$78.15_{\pm 0.10}$	$0.86_{\pm 0.18}$	$75.92_{\pm 0.20}$	$3.30_{\pm 0.10}$	$74.95_{\pm 0.49}$
Tiny ImageNet	$-53.78_{\pm 0.85}$	$62.85_{\pm 0.20}$	$0.75_{\pm 0.10}$	$59.83_{\pm 0.13}$	$2.80_{\pm 0.14}$	$58.36_{\pm 0.60}$

Table 2: Comparison of ResNet20 ensemble connectivity and accuracy (in percent) for deep ensembles and connected ensembles. Averaged over 3 seeds.

	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.

to propose an alternative method for ensembling. Instead, we aim to investigate how constraining ensembles to a single loss basin impacts predictive performance and calibration to gain a better understanding of the loss landscapes of neural networks.

4.1 Permuting Deep Ensembles

Pairwise Alignment. As a first candidate, we 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 performance as the original ensemble. Unfortunately, this guarantee comes at a cost; while $\boldsymbol{\theta}_1$ and $\boldsymbol{\pi}(\boldsymbol{\theta}_i)$ are indeed linearly-connected for $i = 1, \dots, M$ as demonstrated in Ainsworth et al. (2023), pairwise connectivity between two permuted members $\boldsymbol{\pi}_i(\boldsymbol{\theta}_i)$ and $\boldsymbol{\pi}_j(\boldsymbol{\theta}_j)$ does not hold. Similarly, joint connectivity is also violated as shown in Table 3. This is not surprising as the objective only optimizes for pairwise alignment.

Joint Alignment. Next, we evaluate whether the lack of joint connectivity 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 align-

ment 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.

4.2 Distilling Deep Ensembles

Distilled Ensemble. 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 rediscover the j-th member $\boldsymbol{\theta}_j$ in the same basin as $\boldsymbol{\theta}_1$ by minimizing a distillation objective towards $\boldsymbol{\theta}_j$, i.e. we minimize

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} (1 - \beta) \cdot \tau^2 \cdot \text{KL}\left(\sigma\left(\frac{f_{\boldsymbol{\theta}_j}(\boldsymbol{x}_i)}{\tau}\right), \sigma\left(\frac{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}{\tau}\right)\right)$$
$$- \beta \log\left(\left[\sigma\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right)\right]_{y_i}\right) \quad (3)$$

where β and τ are additional hyperparameters. We then start the optimization from $\boldsymbol{\theta}_1^{(t)}$ to encourage connectivity of solutions and denote the minimizers of Eq. 3 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 the temperature parameter commonly used in knowledge distillation frameworks to increase the entropy of the distribution over class labels, facilitating the knowledge transfer to the student model. Table 4 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
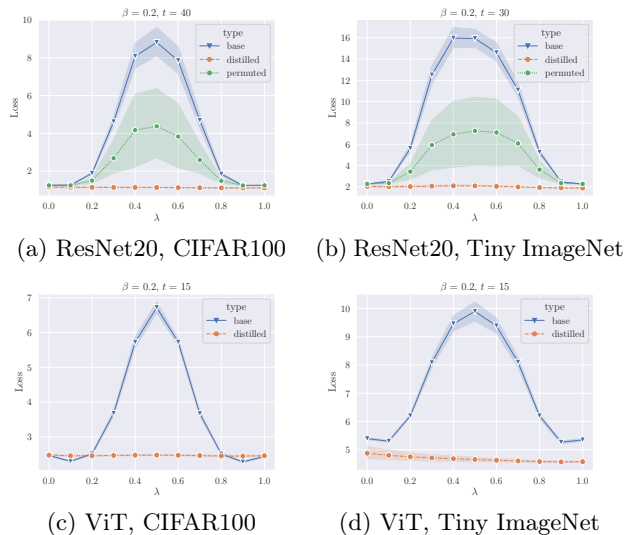
(a) ResNet20, CIFAR100   (b) ResNet20, Tiny ImageNet

(c) ViT, CIFAR100   (d) ViT, Tiny ImageNet

Figure 2: Linear Mode Connectivity of ResNet20 and ViT ensembles. We approximate q_{pair} through lines showing averages of five randomly selected pairs. The experiment is repeated with three random seeds, totalling 15 pairs. The shading shows the standard deviation.
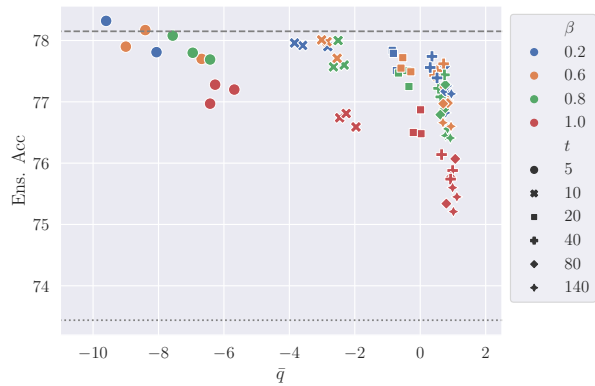

Figure 3: Connectivity $\bar{q}$ plotted against test accuracy for ResNet20 on CIFAR100. The dashed horizontal line shows the accuracy of a deep ensemble, while the dotted horizontal line shows the mean member accuracy.
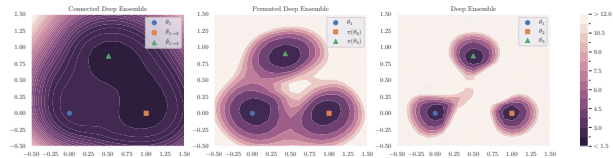

Figure 4: The plots display the 2D planes spanned by the three weight vectors given by the parameters of a ResNet20 trained on Tiny ImageNet mentioned in the legend with the first model at the origin. The plane is constructed as in Garipov et al. (2018).

datasets. Moreover, such a distilled ensemble exhibits a surprisingly strong degree of connectivity, not only fulfilling pairwise connectivity (see Fig. 2), but also the very challenging joint linear connectivity $\bar{q}_{\text{joint}}$, as shown in Fig. 4 and Table 4.

Note that we do not explicitly encourage similarity between two distilled points $\tilde{\boldsymbol{\theta}}_{1 \to i}$ and $\tilde{\boldsymbol{\theta}}_{1 \to j}$. However, the stability of SGD after time t suffices to guarantee joint connectivity. In contrast to the weight-matching procedure introduced above, the distillation framework can also readily be applied to any architecture, including architectures with non-elementwise non-linearities such as Vision Transformers.

Role of β and t. We now investigate the impact of the distillation procedure by varying the hyperparameter β in Eq. 3. When $\beta = 1$, our training procedure reduces to standard training restricted to a single basin, recovering the constrained ensembles from Section 3.2. In contrast, when $0 < \beta < 1$, we optimize the convex combination of the cross-entropy loss and a similarity encouragement term relative to an out-of-basin model. The results in Table 4 demonstrate that similarity encouragement to an out-of-basin model significantly boosts the accuracy of the connected ensemble for both architectures considered. We further highlight the important role of the splitting time t in Fig. 5, trading-off connectivity and performance. Also, note that, as expected, the deterioration in per-

formance as connectivity increases is more pronounced for ensembles that do not include a distillation term ($\beta = 1.0$).

Connectivity and Accuracy. In Fig. 3, 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 (e.g., blue markers), we manage to significantly mitigate the drop in performance without compromising connectivity, narrowing the gap to the baseline of deep ensembles.

Isolating the regularizing effect of distillation. Zhang et al. (2019); Furlanello et al. (2018) have demonstrated that distillation can enhance student performance, to the point that the student can even surpass the performance of its teacher. Thus, the improvement of distilled ensembles beyond the performance of constrained ensembles observed in Table 4 might be caused by this regularizing effect of distillation. To clearly isolate the magnitude of this effect, we consider a baseline of a deep ensemble trained with the same distillation

Kai Lion, Lorenzo Noci, Thomas Hofmann, Gregor Bachmann

		$\beta = 1.0$			$\beta = 0.2$		
		$\bar{q}_{\text{joint}}$	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 4: Comparison of joint connectivity and ensemble performance for constrained ($\beta = 1.0$) and distilled ensembles ($\beta = 0.2$). Averaged over 3 seeds.
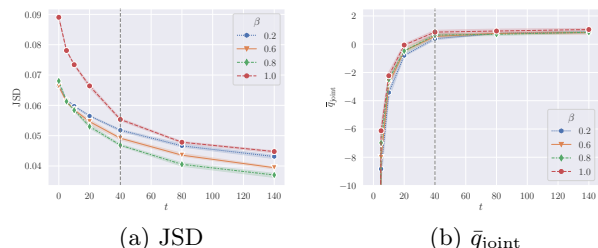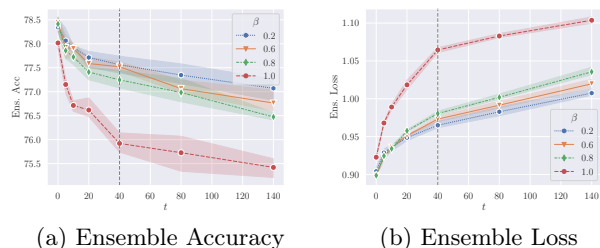

(a) Ensemble Accuracy   (b) Ensemble Loss

Figure 5: Accuracy, loss, and mean accuracy as a function of time parameter t for ResNet20 on CIFAR100. The dashed vertical lines mark the t used in Table 4.

(a) JSD   (b) $\bar{q}_{\text{joint}}$

Figure 6: 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 4.

objective in Eq. 3. If the gain in ensemble performance observed for distilled ensembles would be primarily due to this 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 5, distillation does not significantly improve ensemble performance for ordinary deep ensembles, highlighting that the gain observed in Table 4 is unlikely to be caused by the regularizing effect of distillation.

Diversity and Connectivity. In search of a suitable explanation for the performance gap between connected and deep ensembles, we turn to predictive diversity, which is often deemed as the primary driver of ensemble performance (Dietterich, 2000; Breiman, 1996; Freund et al., 1999). In Fig. 6, we plot diversity and connectivity as a function of t for a grid of β values. In the limit of $t \to 0$, we recover ordinary deep ensembles, due to the loss of joint connectivity. In contrast, for $t > 40$, we enter the regime of connected ensembles whose members fulfill joint connectivity. Note that with increasing connectivity, the diversity, as measured by the one-vs-all Jensen-Shannon Divergence decreases, giving rise to a *diversity-connectivity trade-off* that is likely to be a main driver of the performance gap be-

tween single-basin and ordinary multi-basin ensembles.

5 RELATED WORKS

Connected Ensembles. There is a plethora of previous work that studies novel ensembling techniques, often with a focus on reduced cost or better weight averaging properties. Jolicoeur-Martineau et al. (2023) propose an algorithm that simultaneously trains all ensemble members, dynamically updating the members with a running average. Garipov et al. (2018) and 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. 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. We pro-

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

vide a more detailed overview of related techniques in Appendix A due to space constraints.

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.

6 DISCUSSION

In this work, we have 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 and diversity. 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 for convolutional

networks. While relying on other basins renders our approach very inefficient, it nevertheless proves the existence of very performant ensembles in a single basin, requiring us to rethink the characteristics of loss landscapes. We remark however that the picture is less clear for architectures with less inductive bias such as ViTs.

The existence of strong connected ensembles demonstrates 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.

Benton, G., Maddox, W., Lotfi, S., and Wilson, A. G. G. (2021). Loss surface simplexes for mode connecting volumes and fast ensembling. In *International Conference on Machine Learning*, pages 769–779. PMLR.

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.

D'Angelo, F. and Fortuin, V. (2021). Repulsive Deep Ensembles are Bayesian. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, Lecture Notes in Computer Science, pages 1–15, Berlin, Heidelberg. Springer.

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.

Entezari, R., Sedghi, H., Saukh, O., and Neyshabur, B. (2022). The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks. In *International Conference on Learning Representations*.

Fort, S., Hu, H., and Lakshminarayanan, B. (2020). Deep Ensembles: A Loss Landscape Perspective. arXiv:1912.02757 [cs, stat].

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.

Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., and Anandkumar, A. (2018). Born Again Neural Networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1607–1616. PMLR. ISSN: 2640-3498.

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.

Hendrycks, D. and Dietterich, T. (2019). Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. arXiv:1903.12261 [cs, stat].

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.

Jolicoeur-Martineau, A., Gervais, E., Fatras, K., Zhang, Y., and Lacoste-Julien, S. (2023). PopulAtion Parameter Averaging (PAPA). arXiv:2304.03094 [cs].

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.

Loshchilov, I. and Hutter, F. (2017). SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*.

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.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model' s uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32. 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.

Wenzel, F., Snoek, J., Tran, D., and Jenatton, R. (2020). Hyperparameter Ensembles for Robustness and Uncertainty Quantification. In *Advances in Neural Information Processing Systems*, volume 33, pages 6514–6527. 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.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. (2020). Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. In *International Conference on Learning Representations*.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]

   (b) Complete proofs of all theoretical results. [Not Applicable]

   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes]

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

   (d) Information about consent from data providers/curators. [Not Applicable]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A MORE RELATED WORKS

A.1 Mode Connected Ensembles

Population Parameter Averaging. Jolicoeur-Martineau et al. (2023) propose an algorithm that simultaneously trains all ensemble members. To exploit and preserve linear mode connectivity among members, they repeatedly train the members for a number of epochs on different SGD noise and data augmentations, and replace each member with the average weight vector every fifth epoch.

Subspace Learning. Wortsman et al. (2021) present a method that learns subspaces of networks by uniformly sampling a network from within the subspace to be learned and backpropagating the error signal to the networks spanning the space. They prevent a collapse of the subspace by adding a penalty that encourages orthogonality between the networks constituting the subspace. In a similar manner, (Benton et al., 2021) present a procedure to find low-loss complexes consisting of simplexes whose vertices are found through standard training. A collapse is prevented through a regularization penalty that encourages subspaces with large volume.

Ensembling from a Loss Landscape Perspective. Fort et al. (2020) conclude that subspace sampling methods (weight averaging, MC dropout, local Gaussian approximations) yield solutions that lack diversity in function space and therefore, when comparing against deep ensembles, offer an inferior diversity-accuracy tradeoff. We would like to stress that this finding, which is seemingly contradictory to our work, is not applicable in our setting. The methods used in Fort et al. (2020) start exploring the subspace starting from a fully optimized solution whereas our approach starts exploration right after reaching the stability point. Moreover, the methods employed by Fort et al. (2020) are different to using our distillation approach, which, we argue, is more informative, but admittedly not computationally efficient.

Snapshot Ensembles (SSE). Huang et al. (2017) use a cyclic cosine annealing learning rate schedule to sample multiple models (*snapshots*) within a single training run and ensemble their predictions. Their results demonstrate that snapshots acquired in later cycles become linearly mode connected as they lie in the same basin (cf. Figure 4 in Huang et al. (2017)). Their key insight is that snapshots that lie in the same basin only add limited predictive diversity, suggesting that a cyclical learning rate schedule with a sufficiently high peak learning rate is crucial to escape the basin and attain sufficient predictive diversity for the ensemble to perform well. Our investigation makes the case that staying in the same basin and maintaining linear mode connectivity is not necessarily an impediment to achieving predictive diversity. Put differently, the benefit of ensembling does not depend on escaping the local minima the previous models are sampled from.

Fast Geometric Ensembling (FGE). Fast Geometric Ensembling (FGE) (Garipov et al., 2018) is similar in spirit to SSE. According to Garipov et al. (2018), the feature that distinguishes FGE from SSE is the shorter cycle length between sampling a model. FGE intentionally takes fewer steps in-between sampling models in order to not leave a region of low loss.

Stochastic Weight Averaging (SWA). Izmailov et al. (2018) average weights along the SGD trajectory using a cyclical or constant learning rate. They find that averaging SGD iterates, which often lie on the edge of a loss basin, leads to a more central point and thus to an increase in generalization performance.

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.
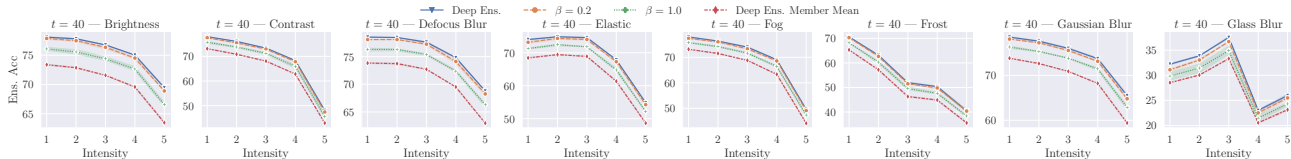
Figure 7: Out-of-distribution robustness of different ResNet20 ensembles on the corrupted CIFAR100 datasets from Hendrycks and Dietterich (2019).

		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 6: Computational cost of ensembles. 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.

A.2 Diversity in Deep Ensembles

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.

B MORE EXPERIMENTS

Predictive Variance. In addition to the Jensen-Shannon divergence in Fig. 6a, we also analyzed the predictive variance. We display the results in Fig. 8 and observe that the results corroborate the conclusions drawn in the main text.

Out-of-distribution Robustness. A significant advantage of deep ensembles over single models is their robustness towards distribution shifts (Lakshminarayanan et al., 2017). Thus, we evaluate whether connected ensembles can match the out-of-distribution robustness of multi-basin deep ensembles in Fig. 7. We use a subset of corruptions from CIFAR100-C (Hendrycks and Dietterich, 2019) and note that distilled ensembles exhibit a similar degree of robustness to distribution shifts as their multi-basin counterparts.
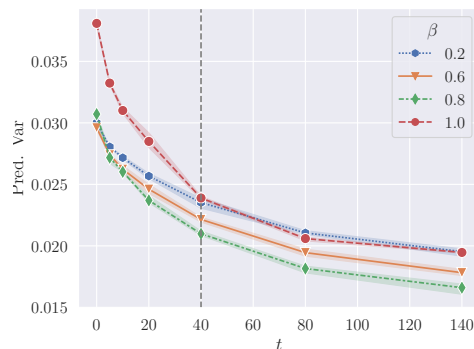

Figure 8: Predictive variance as a function of time parameter t for ResNet20 ensembles on CIFAR100. The dashed vertical lines mark the t used in Table 4.

C   IMPLEMENTATION DETAILS

Computational Cost.   If not stated otherwise, we consider ensembles of size $M = 5$. Table 6 illustrates the computational cost on a per model basis.

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 PERMUTATIONCOORDINAT-EDESCENT algorithm can be found at https://github.com/samuela/git-re-basin. We include pseudo-code for the Multi-PCD procedure that is based on PERMUTATIONCOORDINATEDESCENT and described in Section 4.1. We adopt the notation from Section 3 in Ainsworth et al. (2023) and suppose a model $f_{\boldsymbol{\theta}_i}$ is parameterized by $\boldsymbol{\theta}_i = \{W_\ell^{(i)}\}_{\ell=1}^L$. The permutation matrix applied to the ℓ-th layer of model i is denoted as $P_\ell^{(i)} \in \mathbb{S}_{n_\ell}$.

Algorithm 1 MULTIPERMUTATIONCOORDINATEDESCENT

Require: Ensemble members $\boldsymbol{\theta}_i = \{W_\ell^{(i)}\}_{\ell=1}^L$ with $i = 1, \ldots, M$

   $P_\ell^{(i)} \leftarrow I \quad \forall \ell = 1, \ldots, L$ and $i = 1, \ldots, M$

   while not converged **do**

      for $i \in \text{RandomPermutation}(1, \ldots, M)$ **do**

         while not converged **do**

            for $\ell \in \text{RandomPermutation}(1, \ldots, L)$ **do**

               $P_\ell^{(i)} \leftarrow \text{LAP}\left(\sum_{j \in \{1, \ldots, M\} \setminus i} W_\ell^{(j)} P_{\ell-1}^{(i)} (W_\ell^{(i)})^T + (W_{\ell+1}^{(j)})^T P_{\ell+1}^{(i)} W_{\ell+1}^{(i)}\right)$

            end for

         end while

         $\boldsymbol{\theta}_i \leftarrow \{P_1^{(i)} W_1^{(i)}, P_2^{(i)} W_2^{(i)} (P_1^{(i)})^T, \ldots, W_L^{(i)} (P_{L-1}^{(i)})^T\}$

      end for

   end while

   return $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ ▷ Permuted ensemble members

---

Note that this procedure is different from Algorithm 3 described in Ainsworth et al. (2023), where single members are permuted with respect to the leave-one-out average model and a single merged set of parameters is returned.

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.