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	$-53.78_{\pm 0.85}$	$-46.30_{\pm 2.08}$	$-44.54_{\pm 2.65}$

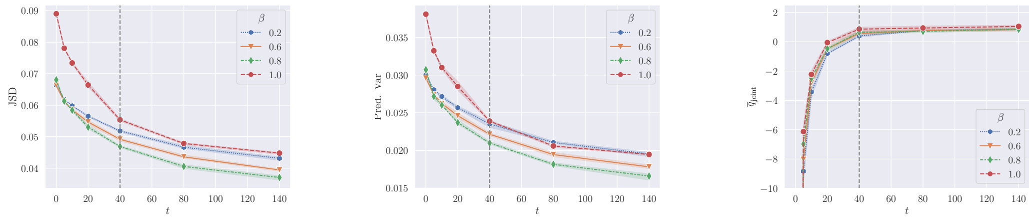
Table 3: Joint connectivity $\bar{q}_{\text{joint}}$ of deep ensembles and permuted ensembles optimizing for pairwise (PCD) and joint alignment (Multi-PCD). Averaged over 3 seeds.

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 $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N \sim \mathrm{Dir}(\mathbf{1})$ 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.