NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Applied Mathematics and Informatics"

BACHELOR'S THESIS

Research Project on the Topic:

Analysis of Neural Networks Internal Representations During Transfer Learning

Submitted by the Student:

group #БПМИ203, 4th year of study — Gritsaev Timofei Grigorievich

Approved by the Supervisor:

Sadrtdinov Ildus Rustemovich
Lecturer, Doctoral Student
Faculty of Computer Science, HSE University

Moscow 2024

Contents

Annotation

Transfer learning and ensembling are two popular techniques for improving the performance and robustness of neural networks. Usually, ensembles are trained from a single pre-trained checkpoint due to the significant expense of pre-training, however, the naive approach to fine-tuning ensembles results in similar models and suboptimal quality. In this work, we revise Snapshot Ensembling (SSE) and its modification StarSSE. We improve the former using similarity losses (Representation Topology Distance and Mean Squared Error) to save pre-trained knowledge. Our modifications reduce the degradation of snapshots and increase ensemble accuracy. We research possibilities of ensemble training time reduction via layers freeze and find that the naive ensemble training cannot benefit from Parameter-efficient Fine-tuning, while StarSSE can sacrifice accuracy but save GPU consumption. Then, we show that successful ideas like weights orthogonalization in the supervised training may not induce improvements in our setup. We increase StarSSE diversity by incorporating directly inducing its loss, though it does not increase ensemble accuracy due to individual accuracy decline. Overall, our experiments and analysis confirm that StarSSE achieves the best individual accuracy and diversity trade-off and should be considered as a strong training ensemble from a single pre-trained checkpoint algorithm.

Аннотация

Предобучение и ансамблирование – два популярных метода повышения качества и надежности нейронных сетей. Обычно учат ансамбли с помощью одной предобученной нейронной сети, поскольку процедура предобучения вычислительно дорогая. Тем не менее, наивный метод обучения ансамбля приводит к низкому разнообразию моделей и не лучшему качеству. В этой работе мы возвращаемся к Snapshot Ensembling (SSE) и его модификации StarSSE. Мы улучшаем первый алгоритм, используя регуляризации (Representation Topology Distance и Mean Squared Error) для сохранения предобученных знаний. Наши улучшенные алгоритмы снижают потери качества нейронных сетей и повышают точность ансамбля. Мы исследуем возможности сокращения времени обучения с помощью заморозки слоев и находим, что наивное обучение ансамбля не может извлечь выгоду из Parameter-efficient Fine-Tuning, а StarSSE может пожертвовать качеством в обмен на сокращение потребления GPU. Затем мы показываем, что успешные идеи, такие как ортогонализация весов, в режиме без предобучения могут не показать улучшения в режиме обучения с переносом знаний, то есть и некоторые хорошие идеи в режиме без предобучения могут оказаться бесполезны в режиме переноса

знаний. Мы увеличиваем разнообразие StarSSE, добавляя прямым образом воздействующую на это функцию ошибки, однако это не приводит к повышению точности ансамбля, так как индивидуальное качество падает. В целом, наши эксперименты и анализ подтверждают, что StarSSE достигает лучшего компромисса между индивидуальным качеством и разнообразием и является хорошим методом для обучения ансамбля из одной предобученной модели.

Keywords

Deep Learning, Computer Vision, Ensemble Learning, Transfer Learning, Topological Data Analysis, Diversity

1 Introduction

Averaging predictions of neural networks, so-called deep ensembling (DE), is a common technique for improving model performance, by enhancing not just precise task-based performance metrics like accuracy, but also the calibration of the model (Lakshminarayanan et al., 2017) and its robustness to noise (Gustafsson et al., 2020), domain shift (Gustafsson et al., 2020), and adversarial attacks (Kariyappa and Qureshi, 2019). Another exceedingly popular approach is transfer learning (Zhuang et al., 2020). It is a method that achieves high-quality neural networks in tasks with limited access to data, using pre-trained on vast amounts of data checkpoint.

Combining these two approaches is challenging because the diversity of models is crucial for successful ensembling. Using networks that are fine-tuned independently from different pre-trained checkpoints, Global DE produces a wide range of models and an excellent final ensemble (Sadrtdinov et al., 2023), but at a high computational cost due to the extensive pre-training phase. Fine-tuning networks from the same pre-trained checkpoint is more resource-efficient, however, final networks are less diverse because they are often located in the same basin of the loss landscape (Neyshabur et al., 2021), resulting in a lower quality ensemble (Mustafa et al., 2020).

Addressing the performance gap between training an ensemble from multiple pre-trained checkpoints and training an ensemble from a single pre-trained checkpoint, combining transfer learning and deep ensembling is an important research area due to its high applicability. Nevertheless, it is still poorly represented in literature.

In this study, we continue research started by Sadrtdinov et al. (2023). The authors recall SSE (Huang et al., 2017), an ensemble training algorithm using a cosine learning rate schedule, and propose StarSSE, which rejects using one continual trajectory so that trains more accurate models and achieves state-of-the-art quality in the transfer learning from one pre-trained checkpoint. However, there is still a noticeable gap between StarSSE and GlobalDE, which trains ensembles using different pre-trained checkpoints. We revise SSE and StarSSE and try to improve them from two perspectives: increase individual accuracy of the ensemble's models and increase diversity.

Throughout SSE experiments was noticed snapshots degradation, which negatively affects the final ensemble accuracy. Degradation appears due to moving away from a high-quality pre-trained checkpoint and losing pre-trained knowledge. To save pre-trained knowledge and train better individual models we propose SSE–RTD, incorporating Representation Topology Divergence (RTD) (Barannikov et al., 2022; Trofimov et al., 2023) into the loss term. The algorithm significantly decreases individual accuracy decline by the saving topological structure of the first ensemble's

network. Moreover, we achieve the same accuracy using MSE instead of RTD, thus, any differentiable metric reflecting the logical concept of similarity could be used in the saving pre-trained knowledge task. Then, we discuss how Parameter-efficient Fine-tuning (PEFT) (Xu et al., 2023) of the network decreases training speed and model performance. We find out that Local DE or Global DE cannot benefit from PEFT, but StarSSE suggests a time-quality trade-off. Finally, increasing diversity without losing individual quality could be another solution to improve StarSSE. For this goal, we create two new algorithms: StarSSE–WO and StarSSE–CE. The former incorporates weights orthogonalization to the loss term but does not even change the training dynamic, and the latter noticeably increases ensemble diversity while achieving not less ensemble accuracy.

Overall, our experiments and analysis show that StarSSE is a strong ensemble method with the best accuracy diversity combination, and increasing individual accuracy leads to a lower diversity while increasing diversity leads to an individual accuracy drop.

Our final code is available at https://github.com/tgritsaev/ens-hse-diploma.

Organization. The rest of this thesis is organized as follows. We discuss relevant related work in Section 2. Section 3 introduces the methodology of the work. In Section 4, we motivate saving pre-trained knowledge and suggest algorithms. PEFT for ensemble methods during transfer learning is discussed in Section 5. In Section 6, we show the importance of increasing diversity and discuss two new methods: StarSSE–WO and StarSSE–CE. Section 7 concludes the paper.

2 Related work

Geman et al. (1992) study bias and variance, which is a popular paradigm in Machine Learning (ML), they are the first who studied bias-variance trade-off for neural networks. They show convincing experimental evidence for the bias-variance trade-off in nonparametric methods such as k-nearest neighbor (kNN) and kernel regression. Since then, bias-variance trade-off has been a popular paradigm, for instance, XGBoost is one of the most popular solutions in the tabular domain (Chen and Guestrin, 2016), which high performance is explained using bias-variance trade-off. However, Yang et al. (2020) rethink the bias-variance trade-off for the generalization of neural networks from a modern perspective. They divide the analysis into under-parameterized and over-parameterized regimes (Belkin, 2021) and state that bias and variance both decrease with increasing neural network width in the over-parameterized regime. They split neural networks' total variance into two types: variance due to optimization and due to training set sampling. Thus, ensembling neural networks by averaging predictions is one of the standard ways to improve a solution, which could be explained by decreasing variance due to optimization, hence total variance decreases while bias remains the same.

Lakshminarayanan et al. (2017) conduct various experiments with DE involving classification and regression tasks to highlight the capabilities of the method. DE produces uncertainty estimates more accurately than those given by Bayesian neural networks and also improves model calibration. The predicted uncertainty is examined for examples from in-domain (ID) and out-of-domain (OOD) distributions to assess how resilient the approach is to changes in the dataset distribution and DE shows higher uncertainty levels for OOD examples. Gustafsson et al. (2020) introduce a framework to evaluate the robustness of computer vision applications in real-world scenarios and utilized it to compare two scalable techniques for estimating uncertainty: ensembling and MC-dropout. The findings of Gustafsson et al. (2020) reveal that ensembling consistently provides more reliable uncertainty estimates and show their robustness to noise, highlighting its superiority in real-world applications. Ensembles are stable to domain shift and adversarial attacks, Ovadia et al. (2019) and Kariyappa and Qureshi (2019) respectively demonstrate this by comparing ensembles to a single model and show better characteristics of ensembles. Moreover, applying DE in various domains (Ganaie et al., 2022), such as Reinforcement Learning, Natural Language Processing, Computer Vision, and other common areas, rarely require any modifications. These appealing qualities make DE a simple approach to improving solution and, thus, regularly applied.

Transfer learning is another well-known method to obtain a high-quality solution for a task with

limited data using less computation. First, a model is pre-trained on a huge amount of data, which is usually open-source and unlabeled, thus, pre-training is commonly unsupervised training requiring a lot of computational resources. For instance, Large Language Models (LLM) are pre-trained via the prediction next tokens for text prefixes (Radford and Narasimhan, 2018), in computer vision, parts of images are masked, and neural networks predict masked parts (Chen et al., 2020). Zhuang et al. (2020) connect and systematize existing transfer learning research and show that transfer learning is an extremely useful technique in various tasks and data domains.

Huang et al. (2017) introduce SSE, which trains an ensemble using cosine annealing with a restart learning rate schedule. Before this article, ensembling was performed by selecting independently trained neural networks (Caruana et al., 2004). In contrast to previous works, SSE spends the budget of a single model for training a whole ensemble.

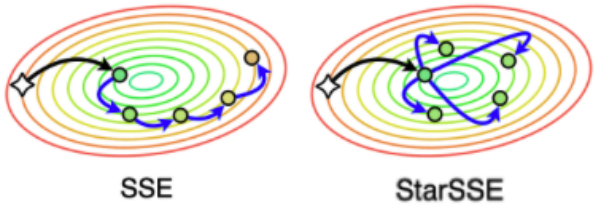

Figure 2.1: SSE and StarSSE intuitive visualization.

Sadrtdinov et al. (2023) train ensembles in the transfer learning setup and focus on methods with a cyclical learning rate like SSE because they either explore one basin of the loss landscape or go outside of the basin depending on the chosen hyperparameters. Thus, cyclical learning methods can explore in different regimes, local or semi-local. The local regime produces an ensemble of models located closer to the pre-trained checkpoint, which lose less pre-trained knowledge and, therefore, are more accurate but less diverse as they lie in the pre-trained basin. In contrast, the semi-local regime results in ensembles further from the pre-trained neural network, which lose more pre-trained knowledge, achieve lower quality and show high diversity. Standard SSE is developed for training from a randomly initialized model, not pre-trained, hence, the authors modify it and split training into two phases: fine-tuning the first neural network and fine-tuning the following models with cyclical learning rate. They use grid search for the values of training epochs, initial learning rate, and weight decay obtaining the highest accuracy on the validation set for choosing hyperparameters of the first phase. Once they find the optimal training epochs, initial learning rate, and weight decay for training the first network, they use grid search for the cyclical learning rate and number of epochs in the cycle giving the best quality of ensemble. To recapitulate, during the first phase they fine-tune a model with the highest accuracy and during

the second they train other models via maximizing ensemble accuracy. It can also be interpreted as projecting to the high-quality solutions space and searching through the space. Figure 2.2 compares SSE and Standard SSE accuracy in the transfer learning setup and shows the former superiority. As during transfer learning a neural network gradually loses the pre-trained information, SSE snapshots degrade throughout training, which results in lower ensemble accuracy. To overcome this issue they propose StarSSE, a modification of SSE, which is more suitable for exploring the pre-train basin in the transfer learning setup. StarSSE analogously fine-tunes the first model via maximizing individual accuracy and then trains the following models, initializing them with the first. Thus, StarSSE does not utilize continuous trajectory as SSE, instead, the second and following models start their training from the first model. Figure 2.1 visualizes SSE and StarSSE algorithms. StarSSE models are located closer to the first model and the pre-trained neural network, so lose less pre-trained knowledge and achieve higher quality, which results in better ensemble accuracy. Sadrtdinov et al. (2023) use two baselines: Local DE, a naively obtained ensemble from one pre-trained checkpoint, and Global DE, a naively obtained ensemble from various pre-trained checkpoints. Local DE trains ensemble by multiple fine-tuning one pre-trained checkpoint, while Global DE obtains ensemble by fine-tuning various pre-trained checkpoints. Finally, algorithms achieve accuracy from the highest to the lowest in the following order: Global DE, StarSSE, Local DE, and SSE. See Table 1 from the paper for the exact results across different datasets.

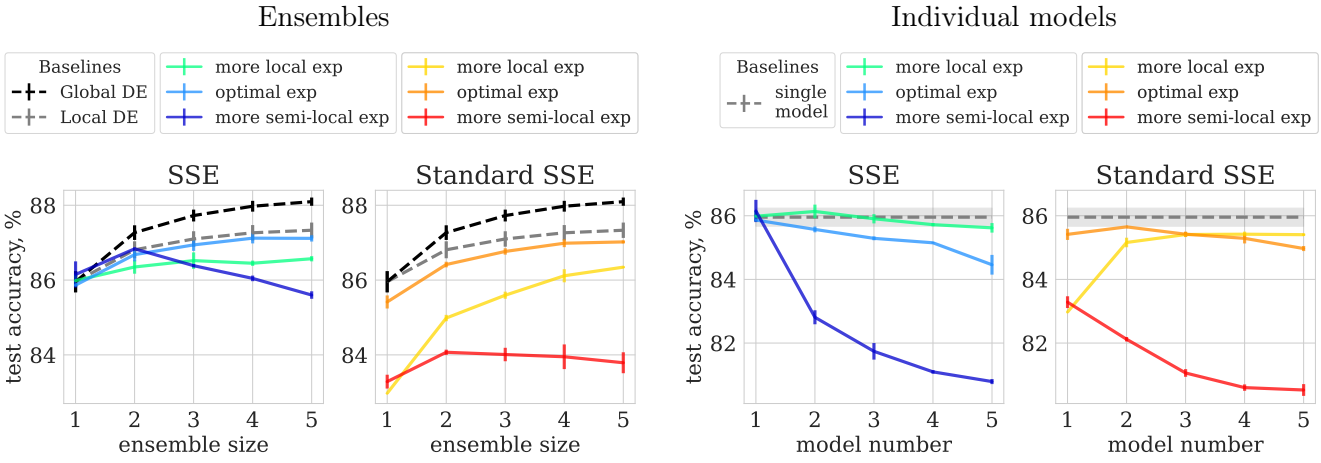

Figure 2.2: Results of ensembles (left plots) and individual models (right plots) for our variant of SSE with a fixed schedule for the first network fine-tuning and Standard SSE, which uses the same hyperparameters in each cycle. CIFAR-100, self-supervised pre-training. Hyperparameters for all three differently behaving experiments are the same for both methods. This figure is taken from Sadrtdinov et al. (2023).

We set up a goal to decrease SSE snapshots degradation. For this, we use Representation Topology Divergence (RTD) (Barannikov et al., 2022; Trofimov et al., 2023). RTD quantifies the dissimilarity

in multi-scale topology between two point clouds of equivalent size with a one-to-one correspondence between points. To the best of our knowledge, RTD is one of the few useful techniques for real ML datasets that is based on Topological Data Analysis. We employ a recently developed method by Trofimov et al. (2023) for differentiating RTD, since its minimization offers closeness in topological features with good theoretical guarantees. Our approach is related to continual learning (Verwimp et al., 2023), a sub-field of ML, which aims to allow ML models to continuously learn on new data, by accumulating knowledge without forgetting what was learned in the past.

Then, we try to fine-tune only a subpart of the network to reduce computational resources consumption. Such an appealing solution is called parameter-efficient fine-tuning (PEFT) and is regularly applied in NLP (Hu et al., 2023). As modern LLM consists of tens of billions of parameters (Floridi and Chiriatti, 2020; Wang and Komatsuzaki, 2021; Black et al., 2022), and fine-tuning a whole network could take a lot of time and computational resources. While PEFT can benefit from this perspective, it can also increase robustness and even quality in some cases (Wang et al., 2022; Xu et al., 2023). It gives motivation to consider experiments with some frozen layers in our setup.

Increasing ensemble diversity is our other research direction. There is no strict definition of diversity or its formula. Intuitively speaking, diversity is higher when models predict exactly the same and lower when predictions are different (Dietterich, 2000; Fort et al., 2020; Wood et al., 2024). Instead of all predictions, some works consider errors and their similarity (Wood et al., 2024). However, a lot of papers consider options to measure diversity (Kuncheva and Whitaker, 2003; Kornblith et al., 2019; Fort et al., 2020; Wood et al., 2024), some of them even link ensemble generalization with average individual model loss and diversity theoretically or empirically (Masegosa, 2020; Ortega et al., 2022; Allen-Zhu and Li, 2023; Wood et al., 2024). A plethora of papers suggest a loss, directly inducing diversity (Liu and Yao, 1999; Pang et al., 2019; Jain et al., 2020; Wortsman et al., 2021), however, all these works do not consider the transfer learning setup, and, to the best of our knowledge, there is no published paper that describes how to incorporate directly inducing diversity function in the transfer learning setup. Our unique contribution is by developing such a loss term, which diversifies models' errors and makes them dissimilar. Our next approach is the weight orthogonalization loss term, which is the same regularization term as used by Wortsman et al. (2021), but we find it useless in the transfer learning setup. Therefore, we highlight that supervised training of an ensemble differs from training ensemble from a pre-trained checkpoint. Another option to improve ensemble quality is to diversify networks by training them in different ways (varying optimization algorithm, learning rate, weight decay, data augmentation, etc.). This

technique is shown to be effective for both Local (Mustafa et al., 2020; Wortsman et al., 2022; Ramé et al., 2023) and Global DE (Gontijo-Lopes et al., 2022). Sadrtdinov et al. (2023) increase StarSSE diversity and ensemble accuracy using different augmentations while diversifying the learning rate and number of epochs is shown to reduce the quality. However, diversifying augmentations is domain specific and may not be easily applied in NLP. Moreover, it is reasonable to combine augmentations diversification and loss function approach to achieve state-of-the-art quality. As it is known, ensembling achieves the best results in uncertainty estimation (Lakshminarayanan et al., 2017; Ashukha et al., 2021), knowledge distillation (Allen-Zhu and Li, 2023), adversarial robustness (Pang et al., 2019), diversity is the key factor for these qualities (Nam et al., 2021; Allen-Zhu and Li, 2023; Pang et al., 2019). Thus, increasing diversity can result in not only more accurate predictions but also improve uncertainty estimation, knowledge distillation, and adversarial robustness.

3 Methodology

Data, architectures, and pre-training. We use a standard ResNet-50 architecture (He et al., 2015) and consider self-supervised pre-trained on the ImageNet dataset (Russakovsky et al., 2015)) with the BYOL method (Grill et al., 2020). We use 2 independently pre-trained checkpoints and average across 3 runs for both of them in all cases. Thus, we set up each experiment 6 times. We choose the image classification task CIFAR-100 (Krizhevsky and Hinton, 2009).

Fine-tuning of individual models. We replace the last fully connected layer with a randomly initialized layer having an appropriate number of classes and then fine-tune the whole network using mini-batch SGD with batch size 256, momentum 0.9, and cosine learning rate schedule (Loshchilov and Hutter, 2019). We use the optimal hyperparameters (number of epochs, weight decay, and initial learning rate), obtained from the previous work (Sadrtdinov et al., 2023), for fine-tuning the first network. For training all the following networks we usually use the same number of epochs and the same learning rate as optimal in Sadrtdinov et al. (2023), however, we vary the learning rate in Section 5. New hyperparameters for each algorithm are discussed in corresponding sections. Sadrtdinov et al. (2023) provide the exact value of their optimal hyperparameters in Appendix B.

Aggregating results. All values presented in the tables are averaged over six runs, so the mean and standard deviation are shown. For Local and Global DEs we take results from Sadrtdinov et al. (2023), which are averaged over 5 runs (they fine-tune 5 networks from each pre-trained checkpoint and average over different pre-trainings and fine-tuning random seeds). For SSE and StarSSE, we average over 6 runs, 3 runs from 2 different pre-trained checkpoints. Notice that Table **??** is taken from Sadrtdinov et al. (2023), thus, SSE and StarSSE results are aggregated over 2 runs. To evaluate diversity, we use an average normalized prediction difference between model pairs on test data, following Fort et al. (2020):

$$\text{diversity} = 100 \cdot \mathbb{E}_{m_1 \neq m_2} \frac{\mathbb{E}[\text{pred}_1 \neq \text{pred}_2]}{\max(\text{err}_1, \text{err}_2)},$$

where m_i stands for a model from an ensemble with predictions pred_i and test error level err_i. We normalize the prediction difference by a maximum of two errors to pay less attention to the diversity of the models with lower accuracy. In all tables, we provide information about average individual accuracy (avg. acc.), which is the mean over individual models in ensemble accuracy, diversity (div.), and ensemble accuracy (ens. acc.).

Overall, we follow the protocol from Sadrtdinov et al. (2023), but conduct experiments 6 times.

4 Approaches to eliminate snapshots degradation

Ensemble performance mainly depends on average accuracy and diversity. Figure 2.2 shows the test accuracy of ensembles of different sizes (left) and the test accuracy of individual models (right). The image shows steady and significant quality degradation throughout training, which negatively affects ensemble quality. A possible approach to improve ensemble quality is increasing average accuracy. For this, we can regularize the training procedure and prevent degradation.

SSE–RTD motivation. Pre-trained knowledge is continually lost during SSE training, we see it as the result of the accuracy decrease of the last models in an ensemble. Moreover, we measure the distance between weights of neural networks and measure various metrics between internal representations and observe a strong correlation between accuracy and distance. For this, we use outputs from Stage 2, Stage 3, Stage 4, Stage 5, AvgPool layers, which are shown in Figure 4.1. Figure 4.2 shows the dependence between test accuracy and four metrics. RTD, Mean-Squared Error (MSE), and Wasserstein distance are measured between the 1-st and the i-th model of the same SSE ensemble, where $i \in \{2, 3, 4, 5\}$. Figure 4.2 (right) shows along the x-axis the logarithm of MSE between AvgPool layer weights of the 1-st and the i-th models. Figure 4.3 shows test accuracy and MSE between the 1-st and the i-th outputs from different layers (top), test accuracy and MSE between weights of different layers of the 1-st and the i-th models. For all figures, the color shows the snapshot number in the ensemble. We consider experiments from Sadrtdinov et al. (2023), thus, the hyperparameters set is taken from Sadrtdinov et al. (2023), Appendix B. We draw two conclusions from these graphics: as more a model moves away from the 1-st neural network, as more accuracy decreases; generally, the last models are more distant from the 1-st than earlier ones. Both ideas are intuitive and logical, but it is important that various metrics, measured on internal representations or weights, geometric or topological, prove it.

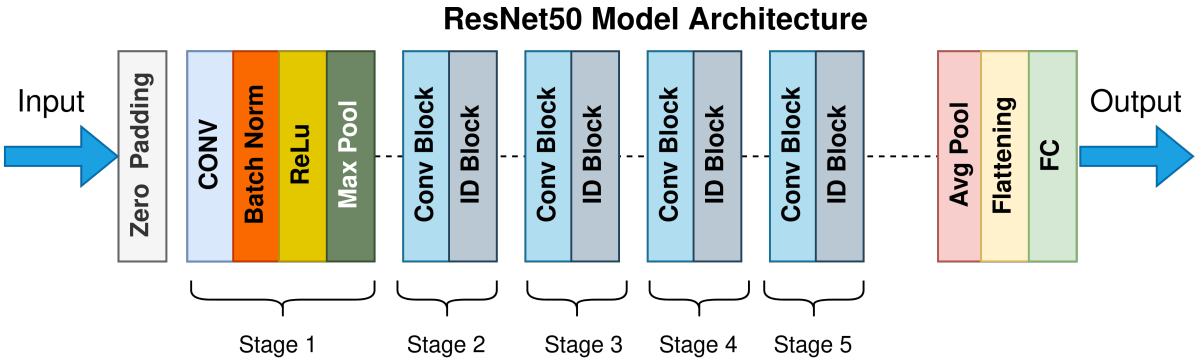

Figure 4.1: ResNet50 architecture.

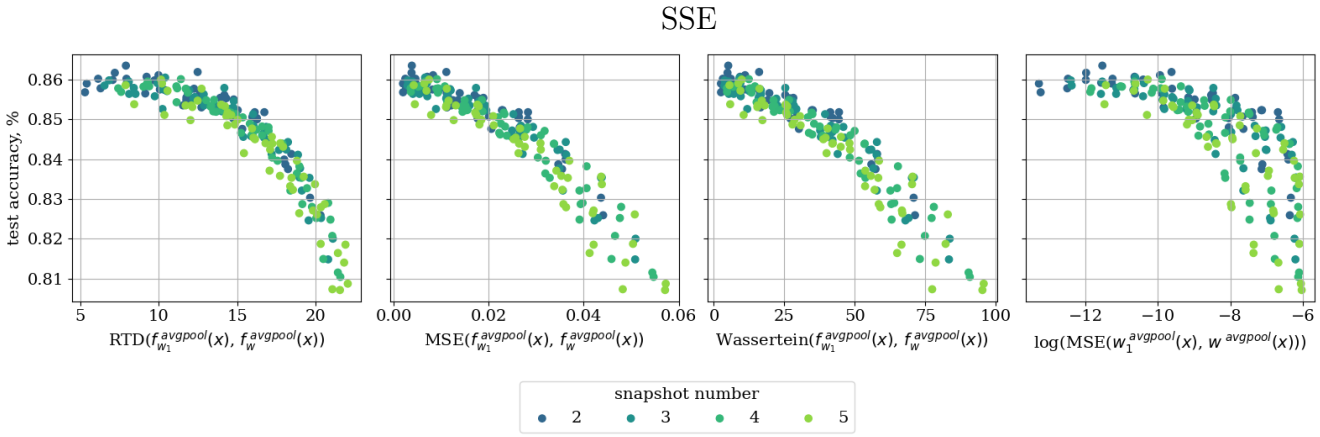

Figure 4.2: **Left, Center Left, Center Right:** test accuracy of individual models and a metric, measured between AvgPool layer output of the 1-st model and the i-th model from the same ensemble. We use RTD, MSE and Wassertein (from left to right). **Right:** test accuracy of individual models and logarithm of MSE between AvgPool layer weights. **Color** indicates the number of the model in the ensemble.

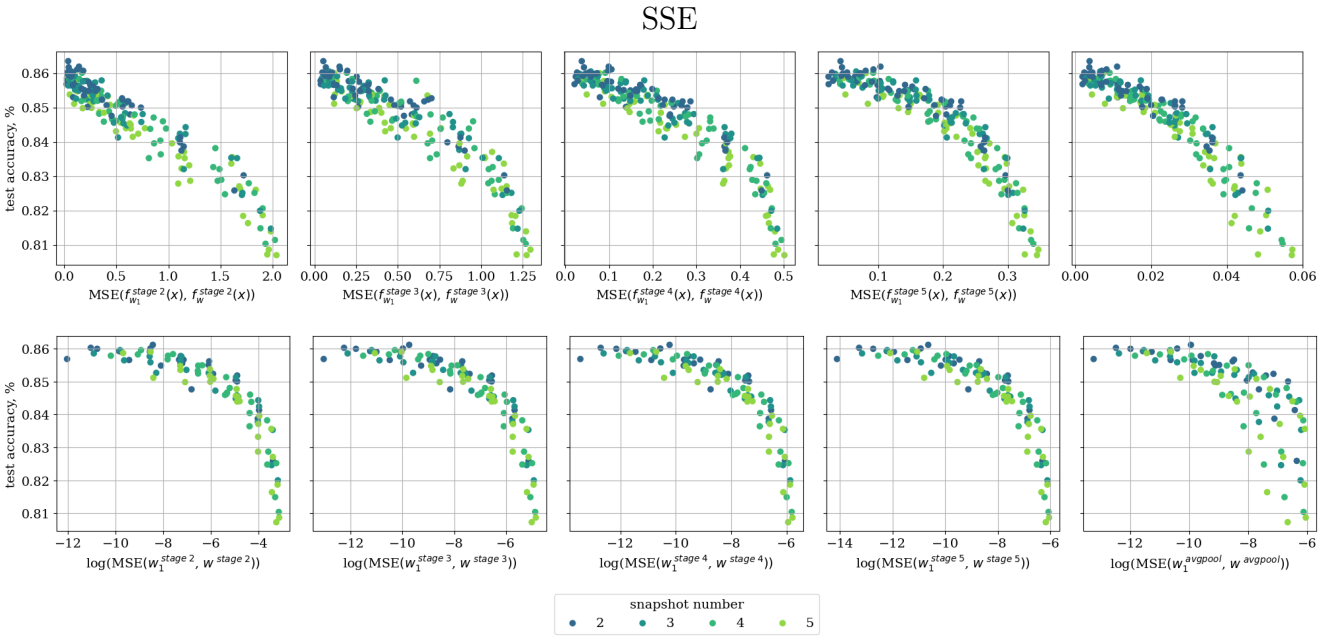

Figure 4.3: **Top:** test accuracy of individual models and MSE measured between different layers output of the 1-st model and the i-th model from the same ensemble. We use the outputs of Stage 2, Stage 3, Stage 4, Stage 5 and AvgPool layers. **Bottom:** test accuracy of individual models and logarithm of MSE between weights of different. We use weights of Stage 2, Stage 3, Stage 4, Stage 5 and AvgPool layers. **Color** indicates the number of the model in the ensemble.

SSE–RTD algorithm. The algorithm almost completely repeats SSE: analogously trains the first optimal neural network (phase I) and then continually fine-tunes snapshots with the same hyperparameters as original SSE, but incorporating regularization term, which makes internal

Table 4.1: Average individual accuracy, diversity, and ensemble accuracy for considered in the thesis algorithms. The fractions in the titles of StarSSE with PEFT show the fractions of used time on V100 32GB relative to the original StarSSE.

Algorithm	avg. acc.	div.	**ens. acc.**
Local DE	86.02±0.09	66.25±3.56	87.33±0.21
Global DE	86.02±0.09	80.21±1.6	**88.10**±0.11
SSE	85.52±0.34	71.57±5.94	87.11±0.06
StarSSE	85.88±0.25	68.12±1.15	**87.41**±0.12
SSE–RTD	85.71±0.07	68.04±5.78	87.36±0.09
SSE–MSE	86.46±0.221	56.62±5.95	87.37±0.02
StarSSE (1) x0.74 time	85.57±0.356	71.37±2.02	87.26±0.06
StarSSE (2) x0.6 time	85.94±0.225	60.94±1.25	87.1±0.14
StarSSE (3) x0.55 time	86.01±0.195	17.38±1.02	86.09±0.22
StarSSE–WO	85.83±0.24	67.31±1.77	87.4±0.16
StarSSE–CE (2)	85.4±0.49	75.21±3.79	**87.44**±0.2

representations of following models and the first neural network more similar (phase II), thus:

$$\begin{aligned}
\text{phase I}: \mathcal{L}_{\text{SSE-RTD}}(w, x) &= \mathcal{L}_{\text{SSE}}(w, x) = \text{CE}(f_w(x), y) \\
\text{phase II}: \mathcal{L}_{\text{SSE-RTD}}(w, x) &= \mathcal{L}_{\text{SSE}}(w, x) + \alpha \cdot \text{RTD}(f^e_w(x), f^e_{w_1}(x)) = \\
&= \text{CE}(f_w(x), y) + \alpha \cdot \text{RTD}(f^e_w(x), f^e_{w_1}(x))
\end{aligned}$$

where x is a training dataset, y are targets, $f_w(x)$ and $f^e_w(x)$ are predictions and internal embeddings of a neural network with weights w on a dataset x, in particularly, w_1 are the weights of the optimal neural network, obtained in the first phase; CE and RTD mean cross-entropy loss and representation topology divergence respectively.

SSE–RTD experiments. We consider experiments with the optimal SSE hyperparameters, obtained in the previous work (Sadrtdinov et al., 2023), and $\alpha \in \{0.0316, 0.1, 0.316, 1, 3.16, 10\}$. We apply RTD-loss to internal representations after the following layers: Stage 2, Stage 3, Stage 4, Stage 5, Avg Pool. Figure 4.1 shows these layers. We provide results and analysis only for the optimal hyperparameter α. Figure 4.4 compares SSE–RTD individual model quality (left) and ensemble of different sizes (right) with SSE. Table 4.1 shows that SSE–RTD performs significantly better than the original SSE and achieves accuracy comparable to Local DE.

SSE–RTD analysis. According to our experiments, SSE–RTD achieves significantly higher individual accuracy than SSE. However, the algorithm still suffers from snapshots degradation, as the second models outperform the last models (Figure 4.4 (left)). As SSE–RTD performs worse

than StarSSE but requires more computational resources, we find it useless in practice.

Meanwhile, RTD is still not a popular metric, thus, it is important to mention that our experiments show that this metric is sensitive to the topological structure of the data representation and agrees with the intuitive assessment of dissimilarity. Thus, RTD could be helpful in continual learning and tasks where important not to forget previous information.

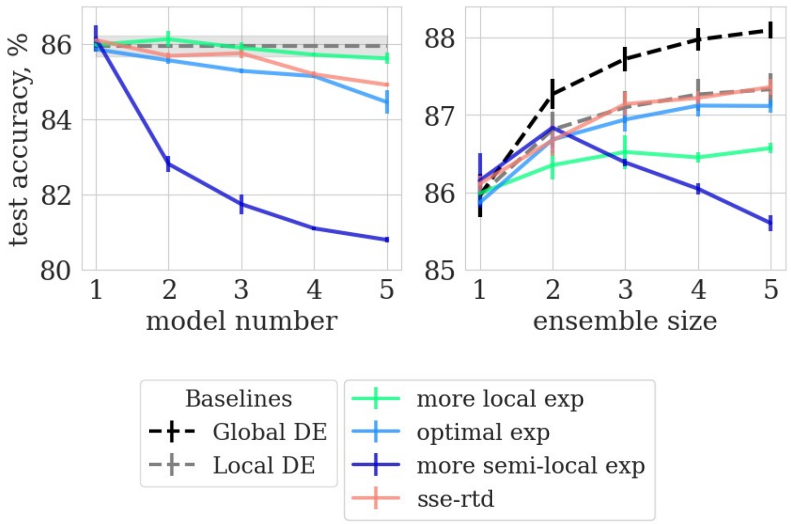

Figure 4.4: Results of individual models (left plots) and ensembles of different sizes (right plots).

SSE–MSE. We conduct experiments with SSE–MSE, which is the same as SSE–RTD but applies MSE instead of RTD. The motivation behind this experiment is to figure out if RTD gives unique regularization qualities or we can achieve the same performance using other differentiable losses. We choose $\alpha \in \{0.1, 1, 10, 20, 40\}$ and find $\alpha = 20$ as the optimal value. Table 4.1 shows that SSE–MSE performs similarly to SSE–RTD, thus, we conclude it does not matter which differentiable metric to use.

Previous approaches to improve SSE. Sadrtdinov et al. (2023) (Appendix H) provide similar experiments. They consider Elastic Weight Consolidation (EWC, Kirkpatrick et al. (2017)), a popular continual learning method, which employs weighted L2 regularization. However, they use EWC on the pre-trained checkpoint as this method cannot be easily applied using the first model. Utilizing the pre-trained checkpoint is a suboptimal strategy, and due to our experiments, it is more effective to use the first model for regularization. They also use L2 weights regularization on the first pre-trained checkpoints. Both methods are similar to ours, but they utilize weights instead of internal representations. Nevertheless, Sadrtdinov et al. (2023) achieve comparable to SSE–RTD and SSE–MSE results, thus, we claim algorithms perform similarly.

StarSSE–RTD. We conduct experiments with StarSSE–RTD, which is trained like StarSSE but

utilizes RTD-loss as SSE–RTD. StarSSE–RTD does not surpass StarSSE because the original algorithm gives maximal diversity with minimal individual accuracy decrease by design. Figure 4.5 shows test accuracy and distance from the 1st model to all others in the same StarSSE ensemble for every set of hyperparameters, considered in Sadrtdinov et al. (2023). Figure 4.5 (top and center) are similar to Figure 4.3, Figure 4.5 (bottom) is similar to Figure 4.2, but for this graphic, we consider StarSSE ensembles and color shows the type of hyperparameters: more local, optimal, or semi-local. Results show an "effective" zone existence, the zone without significant quality degradation while moving further this zone results in rapid degradation. Moreover, the best ensemble is located on the edge of this zone, which gives maximal diversity, and StarSSE finds this edge. onsequently, trying to regularize and save pre-trained knowledge in StarSSE is meaningless because the ensemble should corrupt pre-trained internal representations to obtain diverse models.

The findings of this section lead to the following statement.

Hypothesis: All training methods with continuous trajectory will achieve lower quality than parallel methods like StarSSE in the transfer learning from one pre-trained checkpoint.

The second neural network should move away from the first model. Otherwise, an algorithm will work like a local method, which is shown ineffective in the previous work (Sadrtdinov et al., 2023). Then, if the same training dynamic continues, the following models accuracy continue decrease. If the following models are located near the second, it ends up being the local method. This hypothesis is confirmed by training dynamics in all our experiments.

Moreover, Sadrtdinov et al. (2023) experiment with non-cyclical local ensemble methods in the transfer learning setup and show that they are generally less effective than cyclical ones. They choose KFAC-Laplace (Ritter et al., 2018), SWAG (Maddox et al., 2019) and SPRO (Benton et al., 2021).

Hence, the ensemble training, initializing the second and further models from the first optimal neural network seems to be the most effective scheme in the transfer learning setup.

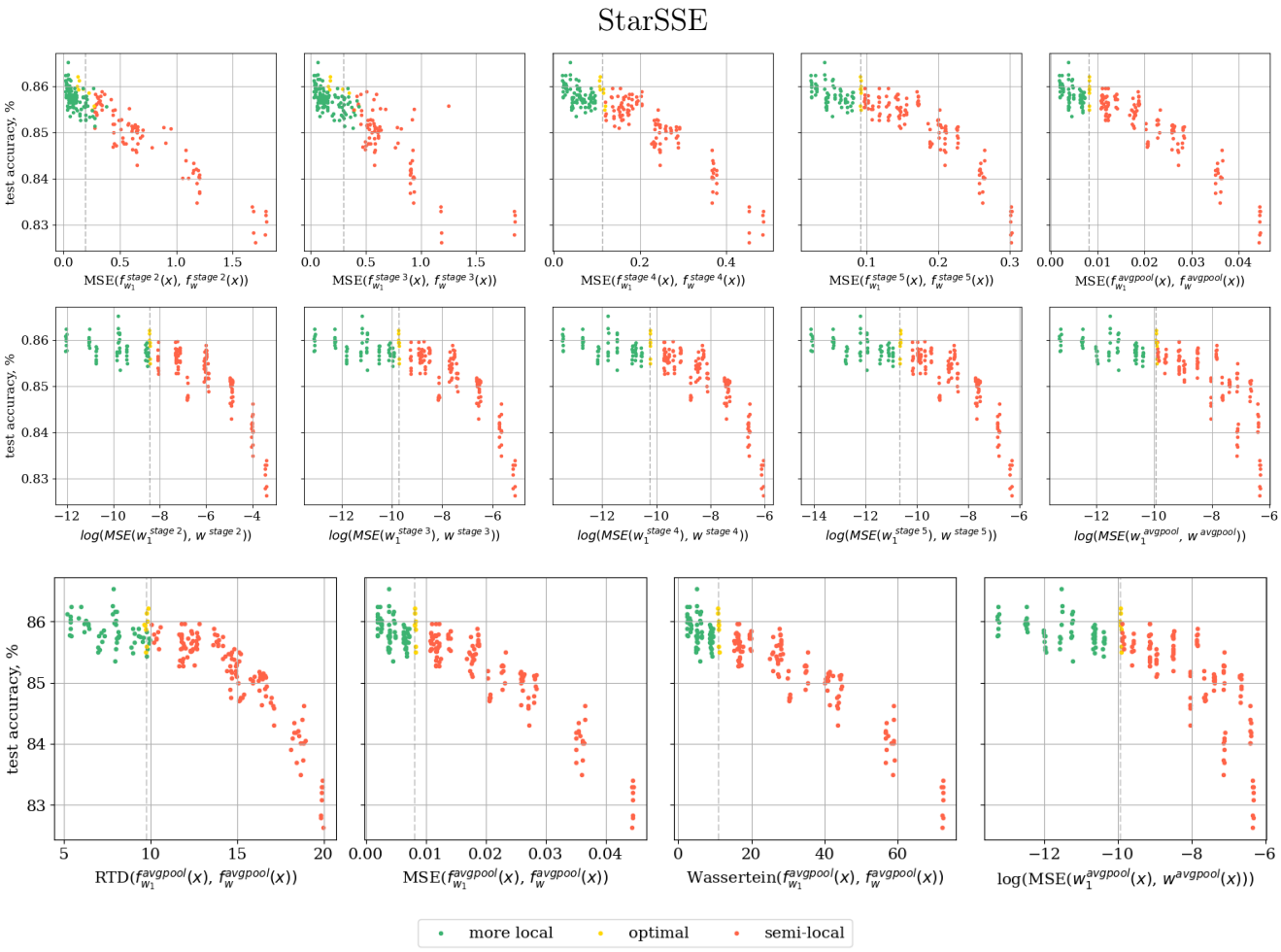

Figure 4.5: Test accuracy of the i-th model in the ensemble and distance between the 1-st model in the same ensemble and the i-th model. **Top:** MSE between internal representations from different layers. **Center:** logarithm of MSE between weights of different layers. **Bottom:** three metrics (RTD, MSE, Wassertein from left to right) between AvgPool layer outputs and logarithm of MSE between weights of AvgPool layer. **Color** indicates more local, optimal and semi-local ensembles, dashed line split optimal from more local and semi-local.

5 Parameter-efficient Fine-tuning for training ensembles

Figure 4.1 presents ResNet50 architecture. In this section, we modify StarSSE and consider three types of experiments: the layers before Stage 4 are frozen (1), the layers before Stage 5 are frozen (2), and the layers before and including Stage 5 are frozen (3). (1) trains of all parameters, (2) trains 64%, and (3) less than 1%, however, they reduce the training epoch time by 1.66, 2.5, and 3.125 times respectively. Note that only deactivating training of the layers at the beginning of the model is meaningful as gradients are calculated and updated starting from the final neural network's layer. Then, freezing more layers results in a higher number of trainable parameters and a lower saved time ratio, so we decide to freeze at least all layers before Stage 3.

At first, we experiment with fine-tuning a frozen pre-trained checkpoint and our results show dramatic accuracy drops to approximately 84%, 81%, and 52% depending on how many parameters are deactivated. Therefore, we claim that Local DE and Global DE cannot benefit from PEFT.

As long as freezing the network during the first cycle results in a huge quality decrease, we decide to make another attempt: fine-tune the whole pre-trained checkpoint to obtain the first model, then fine-tune the other four networks with frozen layers. It leads to the same excellent first model and does not require new hyperparameters search, although, freezing layers results in a more-local solution with all the problems.

Fundamentally, for all three experiments, we do not change any hyperparameter except for the learning rate during the second and the following training cycles. We do not consider less number of epochs as it results in a more local ensemble while increasing the number cancels out time-saving. Other hyperparameters are not as important. We consider the following {x1, x2, x3, x4, x5, x6} learning rates for all experiments, where, for example, x2 means doubled optimal learning rate. Other hyperparameters are optimal from Sadrtdinov et al. (2023) as usual. We use V100 32GB.

Our experiments show that optimal learning rates are between x3 and x5 for all experiments. Table 4.1 compares the optimal results of this algorithm with others, the numbers in the titles show the fraction of used GPU time, compared to StarSSE with no modification. Thus, (1) requires 0.74 of original time, (2) – 0.6 and (3) – 0.55. Note that PEFT also reduces the memory usage.

Overall, we show that Local DE or Global DE cannot benefit from PEFT as StarSSE, however, it is also negatively affected by deactivating parameters during training. To use this approach for training ensembles in the transfer learning setup or not depends on GPU resources constraints, but we suppose that someone may find this time-quality trade-off beneficial.

6 Approaches to increase diversity

Increasing diversity is another approach to improving ensemble performance. Despite Local and Global DE consisting of the same quality neural networks, the latter includes significantly more diverse models, which explains the quality gap between these approaches. SSE and StarSSE optimal neural networks experiments are much more diverse than the ones of the more local experiments, which makes up for lower ID accuracy when constructing ensembles. Moreover, the high diversity of models in optimal StarSSE exceeds one of the Local DE models, allowing optimal StarSSE to outperform the Local DE on clean data. The models of the more semi-local experiments achieve even higher diversity, though it is not enough to overcome their significant quality degradation.

To illustrate high StarSSE variance, we train StarSSE with the optimal hyperparameters twice and choose the ensemble size equal to 15. Then, for both runs, we consider all possible ensembles of size 5, where the first model in an ensemble is always the first trained model because it outperforms others. Thus, we consider $2 \cdot C_{14}^4 = 2002$ different ensembles of size 5. This experiment shows that StarSSE performance is unstable and varies from 87.2 to 87.65, the expected accuracy is nearly 87.42. We consider a similar experiment for Local DE and obtain $C_{15}^5 = 3003$ different ensembles. Figure 6.1 illustrates the results of these experiments, it shows average accuracy, diversity, and ensemble accuracy. We train SVM and illustrate separating hyperplanes between the best and the worst halves of the ensembles. These images confirm the high correlation between individual accuracy with ensemble accuracy and diversity with ensemble accuracy. However, individual accuracy and diversity do not determine ensemble accuracy, at least in our definition of quality and diversity, as we see rare examples of high-quality ensembles on the bottom-left side of the images and low-quality ensembles on the top-right side. Table 6.1 shows the average individual accuracy, average diversity, and average ensemble accuracy of the worst 5% and the best 5% StarSSE and Local DE ensembles. It is reasonable to suggest that for fixed hyperparameters resulted diversity is more important than average snapshot accuracy.

Table 6.2 shows the pairwise model diversity of two StarSSE runs. It is seen that diversity between two random models in an ensemble is not a constant, but if we could obtain an ensemble with maximal pairwise diversity, the ensemble diversity increases, therefore, ensemble accuracy increases.

Inspired by the above-mentioned we suggest two regularizations and implement two methods: StarSSE with weights orthogonalization (StarSSE–WO) and StarSSE with loss leading to more diverse predictions (StarSSE–CE).

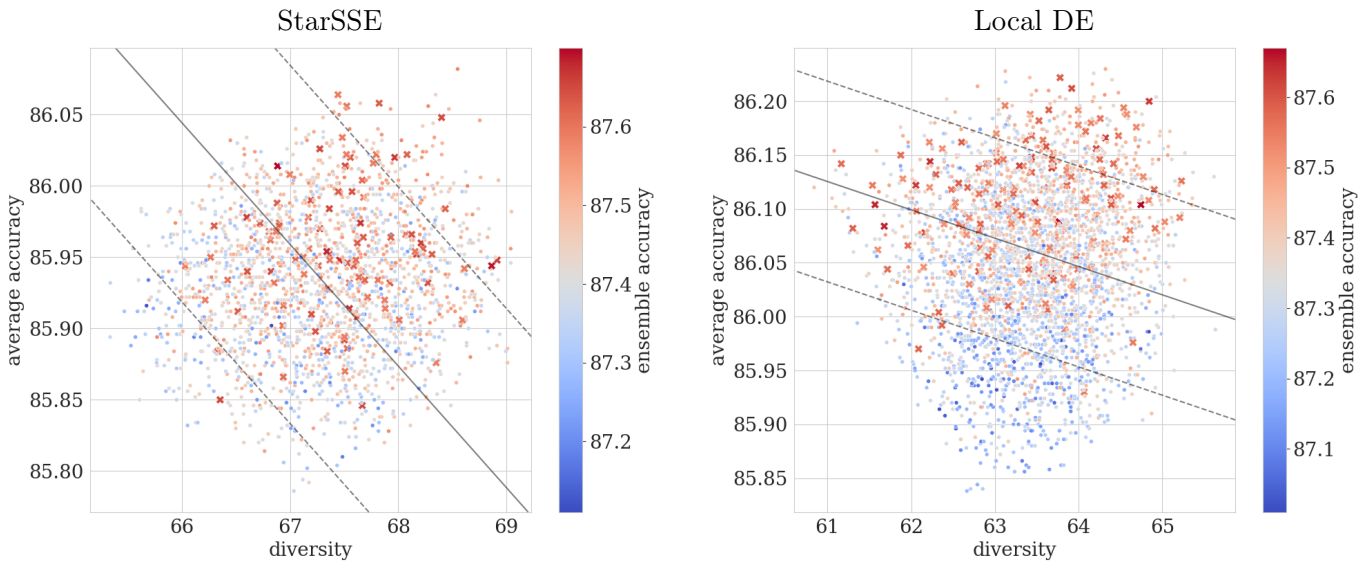

Figure 6.1: Average snapshots accuracy and diversity for optimal hyperparameters varying only by random seeds. Colors show ensemble accuracy, marker 'x' shows the top 5% best solutions. Solid lines shows separating hyperplane between the best halves and the worst halves of the ensembles, dashed lines show 95% confidence interval.

Table 6.1: StarSSE and Local DE mean average accuracy, mean diversity and mean ensemble accuracy of the worst 5% and the best 5% ensembles, trained with the optimal hyperparameters.

Algorithm		worst 5%	best 5%
StarSSE	avg.acc.	85.91±0.04	85.95±0.05
	div.	67.11±0.78	67.44±0.6
	ens. acc.	87.24±0.03	87.62±0.03
Local DE	avg.acc.	86.05±0.05	86.09±0.05
	div.	63.13±0.72	63.81±0.69
	ens. acc.	87.19±0.3	87.57±0.03

Table 6.2: Diversity between models of the two StarSSE ensembles. Roman numerals state the number of the model in the ensemble. Maximal diversity value is highlighted in bold.

	I	**II**	**III**	**IV**	**V**
I	0.0	73.26	70.83	69.58	70.76
II	73.26	0.0	70.93	72.06	**74.49**
III	70.83	70.93	0.0	71.07	71.14
IV	69.58	72.06	71.07	0.0	71.57
V	70.76	**74.49**	71.14	71.57	0.0

	I	**II**	**III**	**IV**	**V**
I	0.0	**75.07**	67.85	69.37	70.62
II	**75.07**	0.0	74.44	73.13	74.65
III	67.85	74.44	0.0	70.7	68.13
IV	69.37	73.13	70.7	0.0	68.88
V	70.62	74.65	68.13	68.88	0.0

StarSSE–WO motivation. Cosine similarity between weights defined as $\cos(w_1, w_2) = \frac{w_1^\top w_2}{||w_1|| \cdot ||w_2||}$ is highly correlated with the pairwise model diversity (Fort et al., 2020; Wortsman et al., 2021). Fort et al. (2020) show that prediction similarity and cosine similarity are high through one training trajectory and low between models initialized with different checkpoints.

Our results show a strong negative correlation equal to -0.625 between diversity and cosine similarity. Figure 6.2 illustrates diversity and cosine similarity between pairs, aggregated across all experiments with different hyperparameters. More precisely, for each trained ensemble we consider $\frac{5 \cdot 4}{2} = 10$ pairs of neural networks, calculate diversity and cosine similarity for each pair, and aggregate this information for all experiments with different hyperparameters, considered in the previous work (Sadrtdinov et al., 2023). The graphic gives motivation to use the minimizing cosine similarity term.

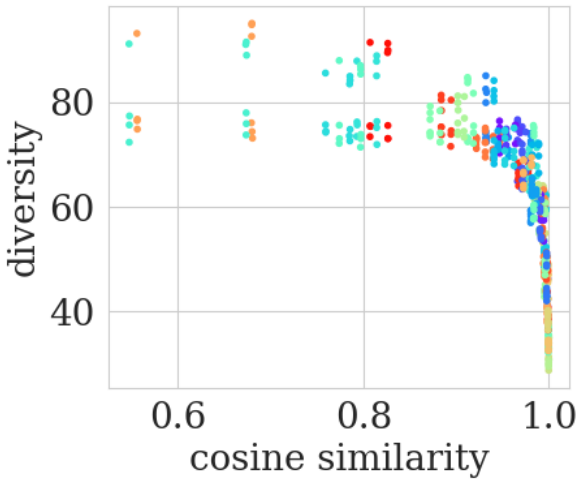

Figure 6.2: Diversity and cosine similarity for experiments with different hyperparameters. Each color means unique experiment.

StarSSE–WO algorithm. As SSE–RTD and SSE, StarSSE–WO almost completely repeats StarSSE: analogously trains the first optimal neural network (phase I) and then parallelly fine-tunes snapshots with the same hyperparameters as StarSSE, but incorporating the regularization term, which orthogonalizes the weights of an ensemble, thus:

$$\text{phase I: } \mathcal{L}_{\text{StarSSE–WO}}(w, x) = \mathcal{L}_{\text{StarSSE}}(w, x) = \text{CE}(f_w(x), y)$$

$$\text{phase II, training the i-th model: } \mathcal{L}_{\text{StarSSE–WO}}(w, x) = \mathcal{L}_{\text{StarSSE}}(w, x) + \alpha \cdot \left(\frac{w^\top w_j}{||w|| \cdot ||w_j||} \right)^2 =$$

$$= \text{CE}(f_w(x), y) + \alpha \cdot \left(\frac{w^\top w_j}{||w_i|| \cdot ||w_j||} \right)^2,$$

all variables are the same as in the SSE–RTD algorithm description, and j is a random integer from 2 to $i - 1$, w_j is the j-th model weights. Thus, the algorithm orthogonalizes weights between all models except the first.

StarSSE–WO experiments. We conduct experiments with the optimal StarSSE hyperparameters and $\alpha \in \{-31.6, -10, -3.16, 1, 3.16, 10, 31.6\}$. Negative α values seem meaningless, but, after unsuccessful experiments with $\alpha > 0$, we want to show that our loss does not change the training dynamic. Table 6.3 shows average accuracy, diversity, and ensemble accuracy for different α.

StarSSE–WO analysis. The diversity fluctuates for different α in Table 6.3. The regularization term does not work as we expected, which is the reason to suggest that the relation between diversity and cosine similarity is not as straightforward as we think. Figure 6.4 shows diversity and

Table 6.3: StarSSE-WO average snapshot accuracy, diversity and ensemble accuracy with different α.

	$\alpha = -31.6$	$\alpha = -10$	$\alpha = -3.16$	$\alpha = 1$	$\alpha = 3.16$	$\alpha = 10$	$\alpha = 31.6$
avg. acc.	85.83±0.12	85.84±0.11	85.85±0.09	85.88±0.09	85.8±0.1	85.79±0.11	85.64±0.06
div.	67.31±1.77	68.12±2.13	66.77±1.85	67.52±1.02	68.76±1.39	66.3±1.86	67.52±0.83
ens. acc.	**87.4**±0.16	87.33±0.17	87.3±0.17	**87.39**±0.1	87.35±0.29	87.26±0.22	87.11±0.04

cosine similarity split by different experiments, rows correspond to one experiment. The global correlation between diversity and cosine similarity is -0.625, we see this strong correlation from Figure 6.2, but the mean correlation between diversity and cosine similarity for pairs inside one run is 0.02, which states no relation for such high precision. We see correlation, but not causation. This result is surprising, because Wortsman et al. (2021) apply the same loss, however, in slightly another setup, not in the transfer learning, and they confirm regularization term importance for training diverse ensembles. Then, we witness similar regularization term values, accurate to normalization on α, which means that training dynamics in the transfer learning setup contribute much more than weights orthogonalization term. Finally, Table 4.1 shows the insignificant difference between StarSSE–WO and StarSSE.

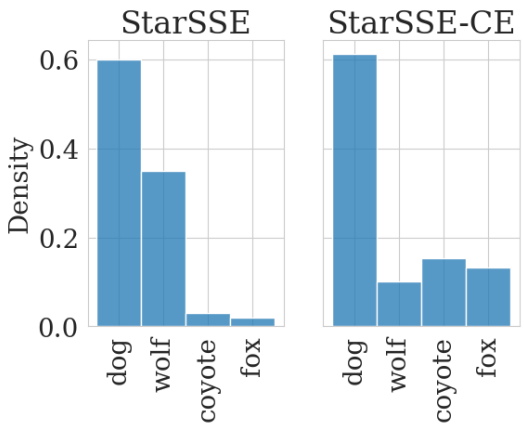

Figure 6.3: The idea behind StarSSE–CE.

Figure 6.4: Diversity and cosine similarity for random StarSSE ensembles. One ensemble in a row. Left shows all pairs, center shows pairs including the first ensemble's model, right shows other pairs, which does not include the first model. Orange color shows the pairs with the first model in the ensemble, green color shows the pairs without the first model in the ensemble.

StarSSE–CE motivation. As in StarSSE–WO, we try to make StarSSE more versatile by adding a simple loss term, which directly diversifies predictions. Figure 6.3 illustrates how StarSSE–CE changes the algorithm.

Table 6.4: Two versions of StarSSE-CE and StarSSE average accuracy, diversity and ensemble accuracy with different α.

StarSSE–CE		$\alpha = 0.00316$	$\alpha = 0.01$	$\alpha = 0.0316$	$\alpha = 0.1$	$\alpha = 0.316$	**StarSSE**
	avg. acc.	85.74±0.04	85.85±0.05	85.85±0.13	85.67±0.18	85.57±0.08	85.88±0.25
(1)	div.	67.44±2.07	67.01±1.58	66.91±1.72	69.45±2.48	70.18±4.07	68.12±1.15
	ens. acc.	87.25±0.06	87.35±0.07	87.35±0.15	87.29±0.04	87.26±0.35	**87.41**±0.12
	avg. acc.	85.79±0.17	85.77±0.15	85.7±0.21	85.4±0.49	85.22±0.5	85.88±0.25
(2)	div.	68.56±1.93	67.68±2.24	70.93±1.19	75.21±3.79	78.13±3.7	68.12±1.15
	ens. acc.	87.38±0.23	87.35±0.29	87.32±0.28	**87.44**±0.2	87.3±0.23	**87.41**±0.12

StarSSE–CE algorithm. Thus, the second phase loss is:

$$\begin{aligned}\text{phase II, training the i-th model: } \mathcal{L}_{\text{StarSSE–CE}}(x) &= \mathcal{L}_{\text{StarSSE}}(x) - \alpha \cdot \text{CE}(f_w(x), y) = \\ &= \text{CE}(f_w(x), y) - \alpha \cdot \text{CE}(f_w(x), f_{w_j}(x)) = \\ &= \text{CE}(f_w(x), y - \alpha \cdot f_{w_j}(x)),\end{aligned}$$

where all variables are the same as in the StarSSE–WO description.

StarSSE–CE experiments. We provide experiments with two options: regularize all labels (1) or regularize only incorrectly predicted labels (2). We consider $\alpha \in \{0.00316, 0.01, 0.0316, 0.1, 0.316\}$. Table 6.4 shows results across different α.

StarSSE–CE analysis. StarSSE–CE (1) performs worse than StarSSE, we suppose it is due to the high accuracy of individual models, thus, regularizing all labels, even correctly predicted, leads to quality degradation. Contrariwise, StarSSE–CE (2) shows similar to StarSSE accuracy, and information from Table 4.1 shows that these algorithms are different. StarSSE–CE (2) shows higher diversity, lower individual model accuracy, and similar ensemble accuracy, however, higher diversity can be beneficial for other tasks, for instance, knowledge distillation (Nam et al., 2021).

7 Conclusion

In this work, we study the effectiveness of cyclical learning rate methods for training ensembles from a single pre-trained checkpoint in the transfer learning setup. We demonstrate that SSE quality degradation could be partially fixed, and the modified methods produce a better ensemble. Our experiments show that StarSSE trains neural networks, which lie on the edge of the local regime, thus, obtaining maximal diversity possible with insignificant individual accuracy decrease. As a consequence, an attempt to improve StarSSE individual accuracy leads to ensemble degradation. Based on our analysis, we propose the hypothesis that, in the transfer learning from a pre-trained checkpoint, the potential of methods that fine-tune an ensemble with continuous trajectory is lower than the potential of the parallel scheme. As cyclical methods' superiority to non-cyclical methods was previously shown, it means that a parallel scheme like StarSSE is the best in this setup. Then we show how to save GPU time consumption during StarSSE training using PEFT and show that this approach cannot be applied to Local and Global DE. We also demonstrate StarSSE and Local DE high variance and significantly increase StarSSE diversity with no losing ensemble accuracy, though individual accuracy is decreased. We also show considering transfer learning setup importance as developed approaches for unsupervised setup may not give desired improvements. Our experiments and analysis prove that StarSSE is a strong algorithm for ensemble training from a single pre-trained checkpoint, and increasing individual accuracy leads to reducing diversity while improving diversity leads to individual accuracy drop. However, increasing ensemble accuracy in the transfer learning from a pre-trained checkpoint is still open.

Acknowledgements

I would like to thank Ildus Sadrtdinov for supervising me through this work. The empirical results were supported by the computational resources of HPC facilities at HSE University (Kostenetskiy et al., 2021).

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning, 2023.

[2] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning, 2021.

[3] Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. Representation topology divergence: A method for comparing neural network representations, 2022.

[4] Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation, 2021.

[5] Gregory W. Benton, Wesley J. Maddox, Sanae Lotfi, and Andrew Gordon Wilson. Loss surface simplexes for mode connecting volumes and fast ensembling. In *International Conference on Machine Learning (ICML)*, 2021.

[6] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022.

[7] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 18, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015432. URL https://doi.org/10.1145/1015330.1015432.

[8] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20s.html.

[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

[10] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, page 1–15, Berlin, Heidelberg, 2000. Springer-Verlag. ISBN 3540677046.

[11] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020. doi: 10.1007/s11023-020-09548-1. URL https://doi.org/10.1007/s11023-020-09548-1.

[12] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective, 2020.

[13] M.A. Ganaie, Minghui Hu, A.K. Malik, M. Tanveer, and P.N. Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, October 2022. ISSN 0952-1976. doi: 10.1016/j.engappai.2022.105151. URL http://dx.doi.org/10.1016/j.engappai.2022.105151.

[14] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992. doi: 10.1162/neco.1992.4.1.1.

[15] Raphael Gontijo-Lopes, Yann Dauphin, and Ekin D. Cubuk. No one representation to rule them all: Overlapping features of training methods, 2022.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[17] Fredrik K. Gustafsson, Martin Danelljan, and Thomas B. Schön. Evaluating scalable bayesian deep learning methods for robust computer vision, 2020.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[19] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models, 2023.

[20] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get m for free, 2017.

[21] Siddhartha Jain, Ge Liu, Jonas Mueller, and David Gifford. Maximizing overall diversity for improved uncertainty estimates in deep ensembles, 2020.

[22] Sanjay Kariyappa and Moinuddin K. Qureshi. Improving adversarial robustness of ensembles with diversity training, 2019.

[23] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Science*, 114(13):3521–3526, 2017.

[24] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited, 2019.

[25] P. S. Kostenetskiy, R. A. Chulkevich, and V. I. Kozyrev. Hpc resources of the higher school of economics. *Journal of Physics: Conference Series*, 1740(1):012050, jan 2021. doi: 10.1088/1742-6596/1740/1/012050. URL https://dx.doi.org/10.1088/1742-6596/1740/1/012050.

[26] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[27] Ludmila Kuncheva and Chris Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51:181–207, 05 2003. doi: 10.1023/A:1022859003006.

[28] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[29] Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10): 1399–1404, 1999. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(99)00073-8. URL https://www.sciencedirect.com/science/article/pii/S0893608099000738.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[31] Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[32] Andres R. Masegosa. Learning under model misspecification: Applications to variational and ensemble methods, 2020.

[33] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Deep ensembles for low-data transfer learning, 2020.

[34] Giung Nam, Jongmin Yoon, Yoonho Lee, and Juho Lee. Diversity matters when learning from ensembles, 2021.

[35] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning?, 2021.

[36] Luis A. Ortega, Rafael Cabañas, and Andrés R. Masegosa. Diversity and generalization in neural network ensembles, 2022.

[37] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

[38] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity, 2019.

[39] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. URL https://api.semanticscholar.org/CorpusID:49313245.

[40] Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization, 2023.

[41] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=Skdvd2xAZ.

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

[43] Ildus Sadrtdinov, Dmitrii Pozdeev, Dmitry P. Vetrov, and Ekaterina Lobacheva. To stay or not to stay in the pre-train basin: Insights on ensembling in transfer learning. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[44] Ilya Trofimov, Daniil Cherniavskii, Eduard Tulchinskii, Nikita Balabin, Evgeny Burnaev, and Serguei Barannikov. Learning topology-preserving data representations, 2023.

[45] Eli Verwimp, Rahaf Aljundi, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L. Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H. Lampert, Martin Mundt, Razvan Pascanu, Adrian Popescu, Andreas S. Tolias, Joost van de Weijer, Bing Liu, Vincenzo Lomonaco, Tinne Tuytelaars, and Gido M. van de Ven. Continual learning: Applications and the road forward, 2023.

[46] Ben Wang and Aran Komatsuzaki. Gpt-j-6b: A 6 billion parameter autoregressive language model, 2021.

[47] Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. Adamix: Mixture-of-adaptations for parameter-efficient model tuning, 2022.

[48] Danny Wood, Tingting Mu, Andrew Webb, Henry Reeve, Mikel Luján, and Gavin Brown. A unified theory of diversity in ensemble learning, 2024.

[49] Mitchell Wortsman, Maxwell Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. Learning neural network subspaces, 2021.

[50] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022.

[51] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment, 2023.

[52] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks, 2020.

[53] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2020.