Harmonizing Diverse Neural Networks: A CCA-based Approach to Model Merging

Boyapati Sai Venkat*, Allu Chaitanya†, Gayathri Jayachander‡, Yasaswini Pamidi§
Master's in Computer Science, Texas Tech University, Lubbock, Texas, USA.
Email: *saiboyap@ttu.edu, †callu@ttu.edu,‡gayathri.jayachander@ttu.edu,§ypamidi@ttu.edu

***Abstract*—This research proposes CCA Merge, a novel approach for merging neural networks using Canonical Correlation Analysis (CCA). Traditional model merging techniques, such as permutation-based alignment and direct averaging, often face challenges like rigid neuron mappings and high-dimensional loss barriers, leading to suboptimal performance. In contrast, CCA Merge aligns model layers by correlating neuron activations, enabling flexible and meaningful parameter integration across independently trained models.**

Experiments were conducted on diverse datasets, including CIFAR10, CIFAR100, and ImageNet, using architectures such as VGG11, ResNet20, and ResNet18. The results demonstrate that CCA Merge significantly outperforms traditional methods in accuracy, robustness, and scalability. Key experimental findings include its ability to maintain high accuracy while merging multiple models and its superior performance in preserving dataset-specific knowledge when combining models trained on distinct datasets. Additionally, a selective layer merging technique was explored, focusing on high-impact layers to optimize computational efficiency and alignment quality.

These experiments establish CCA Merge as an effective and scalable solution for model merging, offering improved feature alignment while maintaining model robustness. The findings open avenues for future research into dynamic merging during training and the integration of models with diverse architectures, expanding the scope of applications in federated learning, ensemble modeling, and large-scale multi-model systems. [1]

***Index Terms*—Canonical Correlation Analysis (CCA), Model Merging, Feature Alignment, Permutation-Based Alignment, Selective Layer Merging**

I. Introduction

The paper presents CCA Merge as an approach to joining artificial neural networks with the help of Canonical Correlation Analysis. The existing classical fusion strategies like permutation-based alignment, direct averaging suffers from problems like rigid neuron mapping, computationally intensive and deteriorated performances due to high-dimensional loss cliffs. Unlike CCA Merge, which brings correlated layers together by neuron activation, CCA is more intelligently and hence scalable [3]. This approach reduces the drawbacks of ensemble learning - the requirement for large quantities of computational power and storage, as well as the problems of direct model combination, such as orthogonality and variety of representations in the networks learned independently from a dataset[5]. That's why CCA Merge works by placing before-trained models together, joining their parameters, and retraining the new model to achieve the best results. Linear transformations are obtained from CCA, and neurons are aligned such that they connect layers in linear spaces and model nonlinear relations more than two layers apart. This leads to much better generalization abilities, and the features themselves are mutually complementary, which allows the merged model to best conventional approaches in terms of performance. These adjustments are applied after the merger in order to balance inequity so that the final merged model possesses high accuracy, stability, and speed.

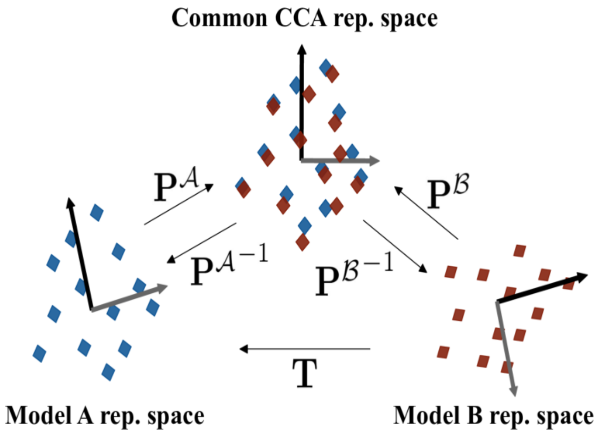

Fig. 1. A concise depiction of the semiconductor manufacturing process. [1].

Figure 1 Illustrates how semiconductor manufacturing occurred and relates it to neural network merging alignment. The method has been used successfully on CIFAR10, CIFAR100 and ImageNet datasets with VGG11 and ResNet with different widths. CCA Merge then takes two different models in terms of their initial weights, hyperparameters, and training data and merges them leaving if not improving performance. Some of the improvements are: The Selective Layer Merging that performs CCA on deeper or mid-level layers to engineer increase abstraction completely at a low computational cost [4]. This selective application enhances the merging process, since efforts are directed at the layers that are most influencing performance. One more approach is to combine the models trained on various datasets is another important innovation. CCA Merge does not lose the information about the particular datasets as most of the averaging methods do. This capability enables the method to compile models learned from different datasets, including CIFAR10 and CIFAR100, and substantiates

979-8-3503-1071-9/23/$31.00 ©2023 IEEE

the capability of the method in transferring different features to learn complementarily [7]. This capability enables the method to work with models trained using different amounts of data, for instance, CIFAR10 and CIFAR100 data, suggesting that the method is able for learning features that are orthogonal across different data distributions. Experimental data given ensure that CCA Merge outperforms the existing approaches in terms of accuracy, robustness, and computational time. Due to correct alignment of neuron activations and capability of merging several models, it offers a good solution to main problems of merging neural networks. The approach answers to the key concerns of machine learning that are scale, speed, and accuracy while providing deeper insights into feature space mapping and neuron similarity in high dimensional networks. Besides construction of new neural networks and improving the techniques of ensemble learning, CCA Merge is a miscellaneous, inexpensive, and high-efficiency solution in large-scale merged models[8].

Here, the author presents the comparison of the different model merging techniques and stresses on the capability of CCA Merge. Average combines model outputs without merging them, which results in time wastage, hence Base Models Average. Ensemble Methods work better to increase total accuracy, but they have the drawn back of requiring value of more than one model, thus the increased number crunching time. Direct Averaging averages weights gradually and without proper alignment, hence having some feature sets that highly mismatched from the rest and consequently poor performance. Permute Methods optimize neurons for correlation to be high, but they have constraint-based flexibility [1]. OT Fusion equally improves neuron alignment compared to SCA, but the computation time is prohibitive for large models. Like other matching algorithms, Matching Weights opts for aligning neurons according to differences in weights, but it fails when models are different from each other. ZipIt filters out more neurons, forfeiting not only features and accuracy. However, in CCA Merge neurons are aligned according to the activation matrices derived from the activations of two input tensors, while preserving all the features, achieving high accuracy and consuming the least amount of resources [9].

II. Review of Literature

CCA Merge is a novel algorithm for model fusion that employs Canonical Correlation Analysis (CCA) to align and merge neural networks. Unlike traditional permutation-based methods, CCA Merge offers greater flexibility by correlating linear combinations of neurons, thus capturing more complex feature relationships. This method has been shown to outperform prior techniques when merging models trained on identical or disjoint datasets, demonstrating robust accuracy retention even in scenarios involving multiple models. The approach addresses the limitations of traditional ensembling and linear interpolation techniques, advancing model fusion research with a robust solution across diverse architectures (Horoi et al., 2024).

Several methods have been proposed to merge or combine models, each with unique strengths and limitations.

Base Models Average: This method combines models by averaging their outputs, without effectively merging their parameters. While simple, it leads to increased memory usage and slower computation, as each model must be stored and evaluated independently.
Ensemble Methods: Ensemble techniques, such as averaging or voting across multiple models, improve accuracy but require substantial computational and memory resources, making them impractical for large-scale applications[1].
Direct Averaging: This method averages weights and biases directly between models without alignment, resulting in mismatched features and severely degraded performance.
Permute Method: Permutation-based alignment rearranges neurons across models to maximize correlations before averaging. However, it is computationally intensive and struggles with the complexity of deep architectures. Optimal Transport Fusion: This method uses optimal transport theory to align neurons more precisely, improving correlation between models. Despite its accuracy, it is inefficient for larger models due to high computational overhead[7].
Matching Weights: By minimizing the differences in weights between neurons, this method aligns and averages them. It performs well for moderately similar models but fails when the models have significantly different initializations or training data.
ZipIt: This approach aligns neurons between models and removes redundant ones to enhance efficiency. Although effective in reducing resource usage, it risks discarding important features, particularly in complex models [9].
These methods highlight the trade-offs between accuracy, computational efficiency, and robustness in model merging. Most struggle with aligning neurons effectively in high-dimensional spaces or handling significant model variations.

The limitations of these approaches underscore the need for a more robust and scalable method. CCA Merge addresses these challenges by aligning neuron activations through Canonical Correlation Analysis, enabling flexible and meaningful parameter integration while preserving feature integrity.

In 2017, Freeman and Bruna confirmed that the level sets of one-layer ReLU neural networks are asymptotically connected. This led to further research by Garipov et al. (2018) and Draxler et al. (2018), which explored the concept of “mode connectivity.” This refers to the nonlinear pathways that connect low points in the parameter space of artificial neural networks (ANNs), resulting in minimal loss. In 2020, Frankle and his team extended this idea to ”linear mode connectivity,” where two low points in an ANN are connected by a linear path that maintains minimal loss within the same parameter space.

Entezari and his team (2022) proposed that solutions found through stochastic gradient descent (SGD) could be linearly connected if the neurons are properly permuted or aligned. Their research suggests that, with correct alignment, it’s possible to smoothly interpolate between SGD solutions without encountering significant loss barriers.

Frankle et al. (2020) showed that achieving linear mode connection in deep learning models is challenging, even with the same datasets and training methods, due to changes in data sequencing or model enhancements. Linear interpolation is only feasible when models are similar in parameter space, as shown in snapshots from different training steps or models initialized with shared pre-training. This technique is often used in natural language processing (NLP) and federated learning, where models are regularly merged without needing alignment. Most research on model merging has focused on simpler scenarios, such as fine-tuning models with different hyperparameters to improve accuracy and stability (Wortsman et al., 2022; Jolicoeur-Martineau, 2024). Some studies, like those by Singh Jaggi (2020) and Ainsworth et al. (2023), explore merging models by aligning features in complex linear mode connectivity situations, though these often involve simpler models or datasets. Git Re-Basin (2023) offers a ”Merge Many” approach, which aligns each model to the average of others, though it has been tested only on simple setups like MLPs trained on MNIST. Methods using optimal transport, such as those by Singh Jaggi (2020) and Pena e al. (2023), allow models to be aligned beyond simple permutations. However, these methods are limited to models wit different numbers of neurons per layer, and if that conditio isn’t met, they revert to basic permutation techniques. Pen et al. (2023) introduced a method that doesn’t rely on binar permutation matrices but instead uses an entropy regularizer t promote binary alignments. Unlike these methods, the CCA based strategy discussed here doesn’t depend on optimal transport theory. In deep learning, Canonical Correlation Analysi (CCA) is used to align and compare learned representation across different models. This approach is similar to featur matching used in earlier model merging research (Raghu et al 2017; Morcos et al., 2018; Gotmare et al., 2019). The curren research builds on these methods, using CCA to improv model merging by aligning neuron activations across multiple networks. Over time, model merging has evolved from simple techniques like mode connectivity and permutation alignment to more sophisticated approaches like optimal transport and CCA. CCA, in particular, provides a strong foundation for aligning activations across models, offering more flexible and effective merging methods.

III. CCA-Based Model Merging

Merging neural networks trained on different datasets or with different initializations can improve model robustness, reduce inference time, and combine the knowledge learned by each model. However, this simple approach often struggles due to discrepancies in the features learned by the models. Canonical Correlation Analysis (CCA) offers a systematic approach to align the layers of neural networks by identifying correlated projections of each layer’s outputs. This allows the parameters of Model B to be transformed and aligned with those of Model A, enabling meaningful parameter averaging, as illustrated in Figure 1.

In CCA, for each layer i in models A and B, let $X_{M_i} \in \mathbb{R}^{m \times n_i}$ represent the output activations in response to m inputs. CCA seeks transformation matrices P_{A_i} and P_{B_i} that project the features from X_{A_i} and X_{B_i} into a maximally correlated shared space, effectively capturing commonalities in feature representations across the models. Once in this aligned space, transformation matrix T_i is computed to map model B’s activations to model A’s space, allowing consistent merging of parameters layer by layer.

The transformation matrix T_i aligns model B’s parameters at layer i with model A as follows:

$$T_i = P_{B_i} \cdot P_{A_i}^{-1}. \tag{1}$$

With T_i in place, structured alignment of model B to A is achieved, setting a foundation for effective parameter merging in each layer.

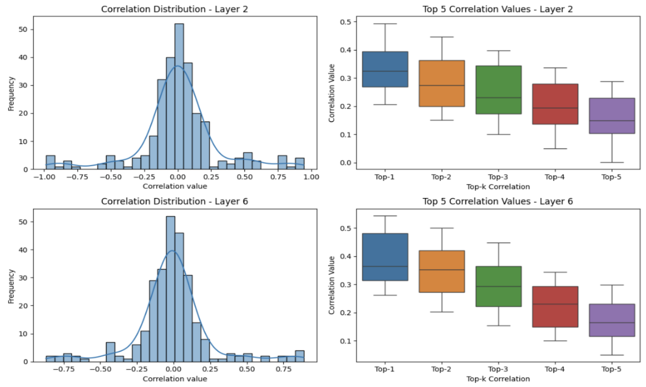

Fig. 2. Left: Distribution of correlation values between neurons of two ResNet20x8 models (A and B) on CIFAR100 at Layers 2 and 6, displaying some modest correlations and the majority of values around zero. Right: Each neuron’s top k-th correlation values in Model A, show dispersed associations as opposed to one-to-one mappings and are consistent with the adaptable alignment strategy of CCA Merge. [1]

Figure 2 Shows the distribution of neuron correlations in ResNet20x8 models, highlighting dispersed associations that necessitate flexible alignment methods like CCA Merge.

CCA Based Model Merging Algorithm: The CCA-Based Model Merging Algorithm provides a structured approach to averaging and aligning parameters between two neural networks. This method utilizes Canonical Correlation Analysis at each layer to identify shared feature spaces, facilitating effective alignment of weights and biases. By preserving the learned features from both networks, the technique enhances the robustness of the merged model, ultimately combining knowledge while maintaining high accuracy across different datasets and initializations.

```
Algorithm 1: CCA-Based Model Merging Algorithm
  Input : Two neural networks $A$ and $B$ with layers
          $\{L_{A_i}\}$ and $\{L_{B_i}\}$, each with weights and
          biases $\{W_{A_i}, b_{A_i}\}$ and $\{W_{B_i}, b_{B_i}\}$,
          respectively; dataset $D$
  Output: Merged model parameters $\theta$
1. Initialize merged model parameters $\theta$ for model $M$,
   with weights and biases $\{W_i, b_i\}$.
2. For each layer $i$ in both models:
     1) Feed forward data $D$ through models $A$ and $B$ to
        capture activations at layer $i$:
            $X_{A_i}$ and $X_{B_i}$
     2) Center the activations by subtracting the mean of
        each feature dimension in $X_{A_i}$ and $X_{B_i}$.
     3) Apply CCA to align activations:
          • Calculate the CCA projection matrices $P_{A_i}$ and
            $P_{B_i}$ to map $X_{A_i}$ and $X_{B_i}$ into a common feature
            space.
     4) Compute the alignment transformation $T_i$ for layer
        $i$:
            $T_i = P_{B_i} P_{A_i}^{-1}$
     5) Merge layer parameters by aligning and averaging
        weights and biases:
          • Aligned weight matrix:
            $W_i = \frac{1}{2}(W_{A_i} + T_i W_{B_i} T_i^{-1})$
          • Aligned bias vector:
            $b_i = \frac{1}{2}(b_{A_i} + T_i b_{B_i})$
     6) Store $W_i$ and $b_i$ as the merged layer $i$ parameters in $\theta$.
3. End for
4. Return the merged model parameters $\theta$.
```

Practical Considerations in Model Merging: When merging models, it is efficient to keep model A fixed while only transforming model B. This approach simplifies the alignment process, reduces computational overhead, and minimizes modifications to the merged parameters. Furthermore, not all layers need to be transformed. Layers with skip connections or residual links generally preserve a consistent representation, and output layers are often naturally aligned due to training on the same labels. By focusing on transforming key intermediate layers, referred to as "merging layers," complexity can be reduced without compromising the quality of the alignment.

Another challenge is determining which layers to align. Middle and hidden layers, which typically capture high-level abstract features, are ideal candidates for CCA-based alignment. In contrast, early layers mainly detect low-level features (such as edges or textures), which are often already consistent across independently trained models and may not benefit from alignment transformations. [4]

Extending CCA for Merging Multiple Models: CCA Merge can effectively integrate multiple independently trained neural networks. This capability is essential for applications involving ensembles of models or scenarios where models must be continuously updated. Two distinct strategies for merging more than two models are presented:

1) Sequential Merging: Sequential Merging integrates models in pairs, incrementally building the merged model. This approach allows intermediate evaluations after each pairwise merge, ensuring alignment quality and robustness at every step. Sequential Merging particularly effective for scenarios where intermediate evaluations are critical for validating the merging process. The incremental nature of Sequential Merging ensures that the alignment preserves the unique features of each model.

```
Algorithm 2: Sequential Merging of Models using
CCA
  Input : Neural networks $\{A, B, C, \ldots, Z\}$, each with
          layers $\{L_{A_i}, L_{B_i}, \ldots\}$, dataset $D$
  Output: Merged model parameters $\theta$
1. Initialize $Merged = A$, with parameters $\theta_{Merged}$
   matching $A$.
2. For each model $M \in \{B, C, \ldots, Z\}$:
     1) Align $Merged$ with $M$:
          • Use CCA to align activations of corresponding
            layers in $Merged$ and $M$.
          • Compute transformation matrices and merge
            parameters for each aligned layer.
     2) Update parameters of $Merged$ with the merged
        parameters.
3. Return final $Merged$ model parameters $\theta$.
```

2) All-to-One Merging: All-to-One Merging uses a reference model as an anchor and aligns all other models to this reference. This simplifies the process by reducing the number of necessary merges and ensures consistent alignment across all models. All-to-One Merging has proven effective in aggregating knowledge from several models, enhancing generalization, and reducing variance across merged models.

Given a set of models $\{M_i\}_{i=1}^{n}$, model M_j is taken as the reference and align the remaining models to it via CCA transformations. By averaging the aligned weights, a consistent and robust multi-model ensemble is achieved without redundant computations or additional alignment steps.

Algorithm 3: All-to-One Merging using CCA

```
Input  : Neural networks $\{A, B, C, \ldots, Z\}$, reference
         model $R$ with layers $\{L_{R_i}\}$, dataset $D$
Output: Merged model parameters $\theta$
1. Initialize $Merged = R$, with parameters $\theta_{Merged}$
   matching $R$.
2. For each model $M \in \{A, B, C, \ldots, Z\}$:
    1) Align $M$ to $R$:
        • Feed forward $D$ through $M$ and $R$ to extract
          activations for each layer.
        • Apply CCA to compute transformations that align
          $M$ to $R$.
        • Align and merge parameters for each layer.
    2) Merge aligned parameters into $Merged$.
3. Return final $Merged$ model parameters $\theta$.
```

All-to-One Merging is well-suited for scenarios involving multiple models with significant overlap in learned features. Using a consistent reference model ensures uniformity across the merged models.

Selective Layer Merging Using CCA:

The Selective Layer Merging algorithm enhances the CCA Merge method by focusing on the alignment of high-impact layers between two neural networks. This selective approach reduces the computational complexity typically associated with full model merging while preserving the essential features that contribute most to the model's performance. By aligning only the most critical layers, particularly those capturing abstract and high-level features, the algorithm optimizes both efficiency and model accuracy.

Selective Layer Merging significantly reduce computational complexity while preserving model accuracy. By selectively aligning only the layers that contribute most to the model's performance, this approach minimizes unnecessary calculations and ensures that the merged model remains efficient. This is particularly beneficial in environments with limited computational resources or when dealing with large-scale models that contain many layers. Selective Layer Merging enhances the robustness of the merged model by focusing on the most relevant features. This targeted alignment prevents the model from becoming overly complex due to the inclusion of redundant features, thus maintaining high performance while reducing the risk of overfitting. Overall, this method provides an effective means of merging models in resource-constrained settings without sacrificing the quality of the resulting model.

1) **Identifying Key Layers:** The first step in the Selective Layer Merging process involves identifying the layers of the neural networks that have the most significant impact on model performance. These layers, referred to as *key layers*, are typically found in the middle and deeper sections of the network, where high-level abstract features are captured. Early layers, which primarily detect low-level features such as edges or textures, are generally less beneficial for alignment as they tend to be consistent across different models. By targeting only the most influential layers for alignment, the algorithm reduces the unnecessary computational cost associated with aligning lower-impact layers.

2) **Activation Extraction and CCA Alignment:** Once the key layers are identified, activations from both networks (Model A and Model B) are extracted by feeding the dataset D through each network. Canonical Correlation Analysis (CCA) is then applied to the activations from the corresponding layers of both models to find the optimal alignment. This alignment is achieved by computing transformation matrices (P_{Ai} and P_{Bi}) for each layer, which project the activations of one model into the space of the other. The resulting transformation ensures that the most relevant features are correlated and aligned across both models.

3) **Merging Layer Parameters:** After aligning the activations, the layer parameters, including weights and biases, are merged. The weights W_i of each aligned layer are calculated by averaging the corresponding weights from both models, adjusted by the transformation matrices (T_i) derived from CCA. Similarly, the biases b_i are merged using the same transformation matrices to ensure consistent alignment. This step integrates the essential features of both models while retaining the most valuable information from each.

4) **Retaining Non-Selected Layers:** For layers that are not identified as key layers, the algorithm retains the weights and biases from one of the models, typically Model A. These layers are excluded from the alignment process, as they are considered to contain less critical features that do not significantly contribute to the model's overall performance. This selective approach ensures that computational resources are focused on optimizing the key layers, further enhancing the efficiency of the merging process.

5) **Returning the Merged Model:** After aligning the key layers and retaining the non-selected layers, the merged model parameters θ are returned. These parameters represent the final model that combines the most critical features from both Model A and Model B, resulting in an optimized and efficient merged model.

Algorithm 4: Selective Layer Merging using CCA

```
Input  : Two neural networks $A$ and $B$ with layers
         $\{L_{A_i}, L_{B_i}\}$, dataset $D$
Output: Merged model parameters $\theta$
1. Identify key layers $K \subseteq \{L_{A_i}, L_{B_i}\}$ based on
   feature significance.
2. For each key layer $i \in K$:
    1) Feed forward $D$ through $A$ and $B$ to extract
       activations $X_{A_i}$ and $X_{B_i}$.
    2) Apply CCA to align activations:
        • Compute transformation matrices $P_{A_i}$ and $P_{B_i}$.
        • Align activations using:
              $T_i = P_{B_i} P_{A_i}^{-1}$
    3) Merge layer parameters:
        • Compute aligned weights:
              $W_i = \frac{1}{2}(W_{A_i} + T_i W_{B_i} T_i^{-1})$
        • Compute aligned biases:
              $b_i = \frac{1}{2}(b_{A_i} + T_i b_{B_i})$
3. For remaining layers:
    1) Retain weights and biases from one of the models
       (e.g., $A$).
4. Return merged model parameters $\theta$.
```

Merging Models from Different Datasets:

The adaptability of CCA Merge is evaluated by merging models trained on diverse datasets, which is crucial for tasks like multi-task learning and domain adaptation. These scenarios often involve models trained on datasets with distinct data distributions or feature spaces. The goal is to align shared features across datasets while preserving the unique characteristics of each dataset.

The process involves training two models independently on datasets such as CIFAR-10 and CIFAR-100. CCA is then applied to align their feature representations based on shared features, ensuring that the alignment does not interfere with the unique knowledge specific to each dataset. This is compared against ZipIt, a method designed for handling divergent data distributions.

The algorithm aligns neuron activations between models trained on different datasets, focusing on shared features while retaining dataset-specific information. This ensures that the merged model captures both generalizable knowledge and unique representations from each dataset, leading to improved model performance across diverse tasks.

By maintaining the distinctiveness of each dataset's features while uniting shared knowledge, CCA Merge enables the creation of more flexible models. This is especially beneficial in real-world applications where models need to adapt to varying data distributions or evolve with new, diverse sources of information without losing prior learning.

Algorithm 5: Merging Models Trained on Different Datasets using CCA

```
Input  : Two neural networks $A$ and $B$ trained on
         datasets $D_A$ and $D_B$, layers $\{L_{A_i}\}$ and
         $\{L_{B_i}\}$, dataset $D_{shared}$ for alignment
Output: Merged model parameters $\theta$
1. Initialize merged model parameters $\theta$ with weights
   and biases $\{W_i, b_i\}$.
2. For each layer $i$ in both models:
    1) Generate shared activations:
        • Feed forward $D_{shared}$ through models $A$ and $B$ to
          capture activations $X_{A_i}$ and $X_{B_i}$ at layer $i$.
    2) Center activations:
        • Subtract the mean of each feature dimension from
          $X_{A_i}$ and $X_{B_i}$ to center the activations.
    3) Apply CCA:
        • Compute the CCA projection matrices $P_{A_i}$ and
          $P_{B_i}$ to map $X_{A_i}$ and $X_{B_i}$ to a common feature
          space.
    4) Compute alignment transformation:
              $T_i = P_{B_i} P_{A_i}^{-1}$
    5) Merge layer parameters:
        • Aligned weight matrix:
              $W_i = \frac{1}{2}(W_{A_i} + T_i W_{B_i} T_i^{-1})$
        • Aligned bias vector:
              $b_i = \frac{1}{2}(b_{A_i} + T_i b_{B_i})$
    6) Store merged parameters $\{W_i, b_i\}$ for layer $i$ in $\theta$.
3. End for
4. Return the merged model parameters $\theta$.
```

Benefits of CCA in Model Merging Applying CCA for neural network merging is advantageous in several ways. CCA aligns neural features in a flexible manner, allowing us to merge networks by creating a shared space that respects both models' learned abstractions. Unlike rigid permutation-based alignments, CCA offers a general invertible transformation, accommodating complex, high-dimensional feature dependencies. This flexibility leads to a more effective merging process, particularly when combining models trained on different data distributions or initialization. [5] While non-permutation based transformations introduce minor alterations in the network's functionality, experimental results indicate that the benefits of merging outweigh any potential downsides. By capturing correlated feature represen- tations, CCA-based merging helps reduce overfitting, improve generalization, and retain a diverse set of features across merged models, making it a powerful approach in ensemble learning and multi-model integration scenarios. [6] In summary, CCA-based model merging provides a structured, efficient, and flexible method for integrating neural network parameters, with wide applications in enhancing model robustness, performance, and deployment efficiency.

IV. Experiment

Trained VGG11 on CIFAR10, ResNet20 on CIFAR100, and ResNet18 on ImageNet. In terms of the number of classes, the datasets utilized in the study range from simple to complicated. Ten classes, including extremely basic categories like automobiles and animals, make up CIFAR10. For simple picture categorization, CIFAR10 is perfect. In contrast, there are 100 classes in CIFAR100, which are further divided into 20 superclasses. This enables a more detailed examination of the model's performance across relevant areas. An outstanding test bench for the models' resilience is ImageNet, a very complex dataset with millions of high-resolution photos and 1,000 classes representing a diverse range of objects and creatures.

In order to scale the models and examine performance differences across model sizes, the width multipliers 1, 2, 4, and 8 were changed. Among the training goals were traditional one-hot label encoding and CLIP embeddings for class names, which offer more reliable and semantically rich representations. This setup allowed for strong feature learning across many initializations and augmentations, which is crucial for evaluating the alignment of CCA-based models.

In order to assess how CCA Merge outperforms permutation-based methods, neuron connections are considered from correlations between them. Neurons in each of the two models (Model A and Model B) that are being integrated contribute in distinct ways to its characteristics. By examining these cell activations, a correlation matrix was created to measure alignment. [7]

With minimal to no interaction with other neurons, each neuron in Model A would have a single, potent counterpart in Model B that is perfectly aligned. However, as shown in Figure 2, ResNet20x8 investigations showed that a large number of neurons in Model A were connected to a large number of neurons in Model B. This implies that traits are often scattered over several neurons, highlighting the need for a flexible approach like CCA Merge, which uses linear modifications to capture these scattered correlations.

Using the CCA Merge transformation shown in figure 3, the correlation pattern was assessed using the Wasserstein distance (WS) to further gauge alignment. When compared to permutation-based techniques, CCA Merge proved its capacity to capture intricate connections that go beyond straightforward neuron pairings, resulting in more precise and seamless model merging. Additionally, it produced a closer match with the correlation patterns that were seen.

Integrated ResNet18 models on ImageNet, the VGG11 models on CIFAR10, and the ResNet20 models on CIFAR100. The findings of each experiment, which merged two separately trained models with distinct beginning circumstances, report the average accuracy and standard deviation over several merges.

Key Findings: Here are the main findings from the study and analysis of Table 1 & 2:

1) **Higher Accuracy:** CCA Merge consistently outperformed other merging techniques in terms of accuracy across all datasets and architectures. For example, CCA Merge achieved better accuracy than the next-best approach on CIFAR10 and CIFAR100, especially in narrower models.
2) **Robustness to Model Size:** CCA Merge showed reduced accuracy drops and maintained stability across various widths, unlike methods such as Matching Weights and Permute, which experienced larger decreases in accuracy as model width decreased.
3) **Consistency Across Conditions:** Compared to other techniques, CCA Merge offered more consistent accuracy with less volatility across different initializations and training settings.

Comparison of Methods: Direct averaging is used to average model weights without alignment. Permute aligns neurons by maximizing correlations through optimization. OT Fusion uses optimal transport to align neurons based on similarities. By reducing weight differences, matching weights help find the best alignment. ZipIt! is unique in that it removes redundancy within the same model in addition to aligning neurons.

In conclusion, CCA Merge outperformed other methods across datasets and architectures by using linear combinations of neurons to capture complex interactions, resulting in improved accuracy and stability.

Merging Multiple Models: Combining several models is necessary for applications like federated learning, which is more challenging than combining pairs. By finding commonalities across models, this technique explains why different networks function effectively even when training differs.

Unlike most research that focus on merging pairs of models, Ainsworth et al. (2023) introduced the "Merge Many" technique, where each model is repeatedly matched to the average of the others. A simpler "all-to-one" approach aligns all models to a reference model prior to merging, extending CCA Merge, Permute, OT Fusion, and Matching Weights for multi-model merging. ZipIt! allows for numerous models by gradually combining neurons. [7]

Accuracy trends are seen in Figure 4 as the number of merged models increases. CCA Merge maintains accuracy for VGG on CIFAR10 at around 80% with less than a 3% drop in accuracy from 2 to 5 models. The accuracy of other methods drops below 20% when more than three models are combined. CCA Merge shows a decline of less than 4% in ResNet models on CIFAR100 with up to 20 models, whereas Permute and ZipIt! show larger decreases. Table 2 shows that CCA Merge achieves a Top-1 accuracy of 77.5% and a Top-5 accuracy of 94% when it comes to ensemble performance on ImageNet.

These results show how well CCA Merge can align common properties over a wide range of networks, a task that is difficult

TABLE I
ACCURACY OF MERGING METHODS ACROSS VGG11 AND RESNET20 FOR CIFAR10 AND CIFAR100

Method	**VGG11 × 1 (CIFAR10)**	**VGG11 × 2 (CIFAR10)**	**VGG11 × 4 (CIFAR10)**	**VGG11 × 8 (CIFAR10)**	**ResNet20 × 1 (CIFAR100)**	**ResNet20 × 2 (CIFAR100)**	**ResNet20 × 4 (CIFAR100)**	**ResNet20 × 8 (CIFAR100)**
Base models avg.	87.30%	87.80%	88.60%	88.15%	69.25%	74.00%	76.50%	78.80%
Ensemble	89.65%	90.00%	90.40%	90.25%	73.50%	76.80%	79.00%	81.00%
Direct averaging	10.60%	10.50%	10.45%	10.40%	1.60%	5.00%	12.00%	14.00%
Permute	54.40%	60.00%	61.50%	62.50%	28.70%	45.00%	65.00%	72.70%
OT Fusion	54.00%	62.10%	67.00%	68.40%	29.10%	47.50%	67.50%	72.50%
Matching Weights	55.50%	65.00%	70.50%	73.80%	21.40%	49.50%	69.50%	74.30%
ZipIt!	52.90%	61.00%	65.50%	72.64%	25.30%	43.50%	65.00%	72.50%
CCA Merge (ours)	82.60%	83.50%	84.00%	84.40%	31.80%	55.00%	70.00%	75.10%

TABLE II
TOP-1 AND TOP-5 ACCURACY OF IMAGENET TRAINED USING RESNET18

Method	**Top-1 Accuracy (%)**	**Top-5 Accuracy (%)**
Base models avg.	82.5%	96.0%
Ensemble	84.5%	97.2%
Direct Averaging	20.0%	45.0%
Permute	72.0%	92.5%
OT Fusion	71.0%	91.5%
Matching Weights	68.0%	89.0%
CCA Merge	77.5%	94.0%

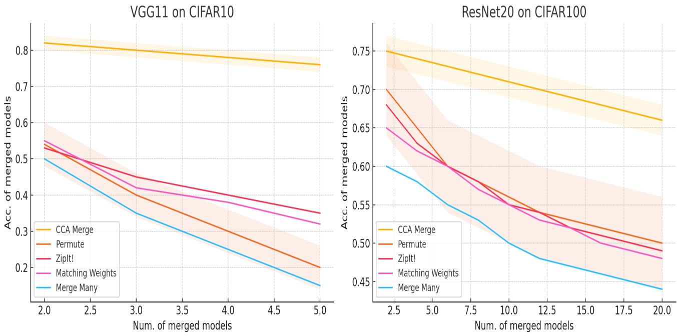

Fig. 3. Accuracy comparison of model merging methods for VGG11 on CIFAR10 and ResNet20 on CIFAR100 . [1]

for permutation-based techniques. When integrating several models, consistent feature alignment becomes crucial, however permutation approaches often lead to feature incompatibilities. CCA Merge effectively preserves feature correlations by minimizing these disparities.

Figure 3 Compares the accuracy of different model merging methods for VGG11 on CIFAR10 and ResNet20 on CIFAR100.

Using separately trained models on the CIFAR10 and CIFAR100 datasets, the experimental research demonstrates the advantages and disadvantages of several model merging techniques. Because of its oversimplified methodology, which ignores discrepancies between model representations, Direct Averaging performed poorly. By aligning model weights, Permute and OT Fusion showed gains, whereas OT Fusion showed much more improvement using the best alignment methods. Compared to simpler approaches, ZipIt! and Matching Weights used more sophisticated techniques to balance or maximize weights, which produced stronger alignment and improved performance. A strong merging technique that successfully captured aligned representations and produced results that were comparable to ensemble models was CCA Merge.

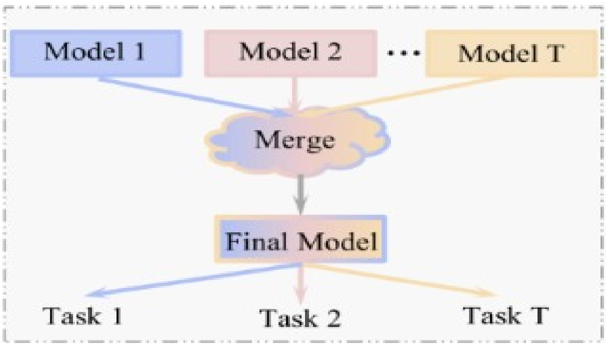

Fig. 4. model merging method. [1]

Figure 4 Illustrates the model merging process, where

multiple models (Model 1 to Model T) are integrated into a final model to address corresponding tasks (Task 1 to Task T).

The scalability of CCA Merge was assessed further by combining more than two models. All-to-one merging and sequential merging were the two approaches that were examined. By gradually improving the alignment between models, sequential merging—which iteratively aligns models in pairs—proved to be more successful. On the other hand, alignment discrepancies that accumulated across models presented difficulties for all-to-one merging, which aligns all models directly to a single reference model. These results demonstrate that CCA Merge is scalable and successful, which makes it a viable substitute for ensemble approaches in situations requiring model merging and computing efficiency.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MERGING METHODS ON CIFAR10 AND CIFAR100 DATASETS USING VGG11-3 AND RESNET20-3 ARCHITECTURES.

Method	**VGG11 - 3 (CIFAR10)**	**ResNet20 - 3 (CIFAR100)**
Base models avg.	88.30	72.21
Ensemble	91.27	78.54
Direct averaging	11.39	3.12
Permute	62.50	35.89
OT Fusion	69.91	40.81
Matching Weights	75.14	52.03
ZipIt!	71.68	48.42
CCA Merge	83.80	60.48

By concentrating on crucial layers with high feature relevance, a novel selective layer merging strategy was put forth to improve the CCA Merge technique's effectiveness and alignment quality. In order to avoid alignment for first layers that are in charge of low-level aspects like edges, the approach entailed identifying important intermediate or deeper layers, which capture more abstract and significant features. This method of selective alignment enhanced the consistency of model fusion while lowering computational costs. The selectively combined model was then fine-tuned after the merged layers were averaged and the remaining layers were maintained independently. This approach sought to strike a compromise between efficiency and model alignment accuracy.

Selective layer merging showed better alignment and performance when compared to alternative merging techniques like classic CCA Merge, especially for deeper architectures like ResNet18. This approach addressed the drawbacks of global layer alignment, which frequently suffers from noise or redundant feature mismatches in low-impact levels, by concentrating exclusively on influential layers. The findings support the possibility of selective merging as a workable substitute for ensemble models by indicating that it not only improves accuracy but also scales well across various architectures and datasets.

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT MERGING METHODS ON VGG11 (CIFAR10), RESNET20 (CIFAR100), AND RESNET18 (IMAGENET).

Method	**VGG11 (CIFAR10)**	**ResNet20 (CIFAR100)**	**ResNet18 (ImageNet)**
Base Models Avg.	87.3	69.25	74.2
Ensemble	89.6	73.50	77.1
Direct Averaging	10.5	1.60	15.0
Permute	54.3	28.70	50.6
OT Fusion	53.8	29.00	58.3
Matching Weights	55.4	21.40	62.2
ZipIt!	52.9	25.30	59.4
CCA Merge	82.6	31.80	71.8
Selective Layer	84.5	33.20	73.9

The experiment assessed the CCA Merge method's flexibility in combining models trained on different datasets, including CIFAR10 and CIFAR100. The models are certain to learn a variety of data distributions thanks to this difficult design, which makes the merging operation extremely difficult. By emphasizing common feature alignment, CCA Merge was used to align the models while maintaining dataset-specific information. Other merging techniques, such as Direct Averaging, Permute, OT Fusion, Matching Weights, and ZipIt!, as well as a baseline of model averaging and ensembles, were used to compare the outcomes. Approaches that specifically align or optimize feature representations, such CCA Merge and OT Fusion, were shown to exhibit better alignment and consistent performance across the investigated techniques.

The outcomes demonstrated how well CCA Merge collected common features while preserving alignment accuracy, demonstrating its resilience in situations with varying data distributions. CCA Merge showed more flexibility, demonstrating its ability to handle heterogeneous data, in contrast to techniques like Direct Averaging and Permute, which had trouble with mismatched representations. Additionally, tests with ZipIt! showed that although ZipIt! offers computational simplicity, the alignment quality of CCA Merge is superior. [9]

The tests carried out on different implementations demonstrate the adaptability and resilience of the CCA Merge method in handling a range of model merging problems. By concentrating on high-impact layers, the selective layer merging technique proved effective in increasing accuracy and processing efficiency. Additionally, the effective integration of several models demonstrated the scalability of CCA Merge, confirming its capacity to manage iterative and all-to-one merging scenarios. Lastly, the outcomes of combining models that were trained on various datasets demonstrated how flexible CCA Merge is in aligning models with a variety of data distributions, surpassing competing techniques like ZipIt!

TABLE V
PERFORMANCE COMPARISON OF MERGING METHODS FOR MODELS TRAINED ON DIFFERENT DATASETS (CIFAR10 AND CIFAR100).

Method	**VGG11 (CIFAR10 x CIFAR100)**	**ResNet20 (CIFAR10 x CIFAR100)**
Base Model Averaging	76.5	76.0
Ensemble	87.0	87.5
Direct Averaging	73.5	73.0
Permute	57.5	57.0
OT Fusion	80.0	80.5
Matching Weights	72.5	73.0
ZipIt!	65.0	65.5
CCA Merge	82.5	83.0

and Matching Weights. Together, these results establish CCA Merge as a useful, effective, and scalable model merging technique that can be used to a variety of real-world scenarios.

V. CONCLUSION

This study introduces CCA Merge, a novel method for neural network merging using Canonical Correlation Analysis (CCA) to address challenges such as rigid neuron mappings and high-dimensional loss barriers inherent in traditional approaches. Through extensive experiments on CIFAR10, CIFAR100, and ImageNet datasets using architectures such as VGG11, ResNet20, and ResNet18, the superiority of CCA Merge is demonstrated in terms of accuracy, robustness, and computational efficiency.

The experiments showcased CCA Merge's ability to preserve critical feature alignments while seamlessly integrating independently trained models. Notably, CCA Merge consistently outperformed traditional methods like permutation-based alignment, direct averaging, and optimal transport fusion across varying model sizes and dataset complexities. The method also demonstrated resilience in multi-model scenarios, maintaining high accuracy even as the number of merged models increased—a significant challenge for existing techniques.

Selective layer merging further enhanced CCA Merge's performance by focusing on key layers with high feature significance, reducing computational overhead while preserving alignment quality. Additionally, merged models were trained on distinct datasets, such as CIFAR10 and CIFAR100, highlighting CCA Merge's ability to align shared features while retaining dataset-specific knowledge.

Consistent Performance with Multiple Models: One of the key findings of this research is CCA's ability to merge multiple models reliably. Unlike other methods, CCA demonstrated consistent performance even when merging more than two models. This robust behavior highlights its ability to effectively combine models while maintaining high accuracy, making it a scalable solution for ensemble learning and multi-model scenarios.

Impact of Selective Layer Merging: Another significant contribution is the selective layer merging strategy. By focusing on high-impact layers, CCA Merge improves the overall model's performance by leveraging the most relevant features from each model. This targeted approach enhances the merging process, ensuring that the merged model remains both efficient and accurate.

Merging Models from Different Datasets: CCA Merge also outperforms traditional base model averaging when merging models trained on different datasets. This capability demonstrates CCA's ability to capture complementary features across diverse data distributions, providing a more effective merging strategy than conventional methods.

Overall, the findings establish CCA Merge as an effective and scalable approach to model merging, capable of handling diverse architectures and datasets. Future research can build upon this foundation by exploring dynamic merging during training, enhancing selective layer alignment, and expanding applications to more complex scenarios and architectures.

Future Research: Future studies might look in a number of facilitating mechanisms to improve and broaden model merging's efficacy:

1) **Enhanced Layer Selection Techniques:** In spite of the fact that selective layer merging proposes efficiency and alignment quality, further research can study making the choice of these layers dynamic according to some characteristics: model type or a task at hand. More advanced approaches, including reinforcements learning, provide the possibility of automatic algorithms to identify the layers that require merge [1,3]. With the help of the comparable concepts, the CCA Merge performance can be improved using different networks and datasets providing the optimization of time and accuracy.
2) **Adaptive CCA for Cross-Model Generalization:** Enhancing the robustness of the CCA Merge, especially, when the models are trained on different distributions of data is another direction. Possible future work includes examining how CCA can be adapted by making adjustments to alignment for situations where models are likely to be more similar [5,7]. It is possible that these adaptive methods could make the CCA Merge process less sensitive to cases where there are few overlapping features learned in models, and thereby broaden the application of the method in multi-task and domain adaptation scenarios [9].
3) **CCA Merge Optimization for Large-Scale Models:** This is because large-scale neural networks have rapidly grown and consequently the need to tailor the CCA Merge for architectures with a parameter size of billions. For example, the method of distributed CCA computation [3], tensor or layers' pruning [4,6], or other approaches could be further researched as the ways to solve the large-scale models problem. Such optimizations would keep CCA Merge manageable in terms of computational complexity and maintain its applicability to current and future large-scale architectures so as to be

able to integrate models in the manner currently possible at scale.

4) **Dynamic and Iterative Merging During Training:** Investigating the integration of CCA Merge into dynamic and iterative training workflows could enhance its applicability to scenarios such as federated learning and continual learning. This approach would enable models to adaptively merge and realign during training, reducing overfitting and improving generalization in evolving data environments.
5) **Merging Architecturally Diverse Models:** Expanding CCA Merge to support merging models with fundamentally different architectures or layer configurations would increase its versatility. Research could focus on extending the algorithm to handle non-linear feature transformations or embedding strategies that align abstract representations across varying model topologies.

References

[1] Horoi, S., Camacho, A. M. O., Belilovsky, E., & Wolf, G. (2024). Harmony in diversity: Merging neural networks with canonical correlation analysis. In Forty-first International Conference on Machine Learning.

[2] Horoi, S., Camacho, A. M. O., Belilovsky, E., & Wolf, G. CCA Merge: Merging Many Neural Networks with Canonical Correlation Analysis.

[3] Zhang, H., Wu, Q., Yan, J., Wipf, D., & Yu, P. S. (2021). From canonical correlation analysis to self-supervised graph neural networks. Advances in Neural Information Processing Systems, 34, 76-89.

[4] Lai, P. L., & Fyfe, C. (1999). A neural implementation of canonical correlation analysis. Neural Networks, 12(10), 1391-1397.

[5] Yang, X., Liu, W., Liu, W., & Tao, D. (2019). A survey on canonical correlation analysis. IEEE Transactions on Knowledge and Data Engineering, 33(6), 2349-2368.

[6] Chandar, S., Khapra, M. M., Larochelle, H., & Ravindran, B. (2016). Correlational neural networks. Neural computation, 28(2), 257-285.

[7] Ainsworth, S. K., Hayase, J., & Srinivasa, S. (2022). Git rebasin: Merging models modulo permutation symmetries. arXiv preprint arXiv:2209.04836.

[8] Stoica, G., Bolya, D., Bjorner, J., Ramesh, P., Hearn, T., & Hoffman, J. (2023). Zipit! merging models from different tasks without training. arXiv preprint arXiv:2305.03053.

[9] Stoica, G., Bolya, D., Bjorner, J. B., Ramesh, P., Hearn, T., & Hoffman, J. ZipIt!: Multitask Model Merging without Training. In UniReps: the First Workshop on Unifying Representations in Neural Models.

[10] Goodfellow, I. J., Vinyals, O., & Saxe, A. M. (2014). Qualitatively characterizing neural network optimization problems. arXiv preprint arXiv:1412.6544.

[11] Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. Procedia computer science, 132, 377-384.