Model Merging for Continual Learning

Louis Barinka   Tanguy Dieudonné   Clotilde Laval   Lars Schuster
ETH ZÜRICH

Abstract

Adapting neural networks to sequential tasks without succumbing to catastrophic forgetting is a core challenge in continual learning (CL). Inspired by mode connectivity, we introduce **Mode Path Fusion (MPF)**, a novel model merging method that constructs a low-loss path in parameter space to merge models trained on different tasks. Unlike traditional methods relying on linear averaging, MPF better preserves task-specific performance while integrating multiple models. Comprehensive experiments on MNIST and CIFAR-10 demonstrate that MPF consistently outperforms baseline methods. This method opens new avenues for robust and scalable continual learning frameworks. https://github.com/tanguy8001/continual-learning-via-model-merging

1. Introduction

Under ever-changing real-world conditions, intelligent systems must continuously adapt by learning from new experiences while retaining previously acquired knowledge. This process, referred to as *continual learning (CL)*, is crucial for developing robust and autonomous AI systems capable of performing diverse tasks across time.

In the general framework, a model is presented with a series of tasks, each consisting of datasets, with the model having access to only one task at a time (Wang et al., 2024). The objective is to effectively learn and retain knowledge from all previous tasks. A central difficulty in CL lies in balancing plasticity for learning new tasks with preserving performance on older tasks. A trade-off often disrupted by *catastrophic forgetting* —a phenomena where the model's performance on earlier tasks excessively degrades when having to learn new ones.

Early efforts in CL explored methods for preserving prior knowledge, however, these approaches often come with trade-offs. Regularization-based methods like Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) focus on stabilizing task-critical parameters but can struggle with highly heterogeneous tasks. Replay-based strategies, such as Generative Replay (Shin et al., 2017), mitigate forgetting by revisiting prior task data, yet are limited in scenarios requiring data privacy or constrained memory. Architectural methods like Progressive Neural Networks (PNNs) (Rusu et al., 2022) avoid interference by allocating separate parameters for each task but face scalability challenges due to unbounded growth.

Recently, model merging strategies, such as Fisher-weighted averaging (Marouf et al., 2024) or selective weight deviation storage (Marczak et al., 2024), have shown promise in preserving knowledge in CL while enabling task adaptation.

The concept of *mode connectivity* has emerged as a key phenomenon in understanding the optimization landscapes of neural networks. Mode connectivity suggests that the local minima found by stochastic gradient descent (SGD) are not isolated points in the parameter space but are instead connected by continuous low-loss paths. First explored by Goodfellow et al. (2015), mode connectivity has since been extensively studied in the literature (Keskar et al., 2017), (Venturi et al., 2020), (Webson et al., 2023).

Notably, Draxler et al. (2019) demonstrated that loss minima form a connected manifold, enabling continuous paths with low loss values between different minima. This finding challenges the traditional view of minima as isolated valleys in the parameter space. In a related approach, Garipov et al. (2018) further showed that these minima could be connected via simple parametric curves, such as Bezier curves, with near-constant loss across training and test sets. Their work provides a practical procedure to identify these paths efficiently. Entezari et al. (2022) extend this understanding by introducing *permutation invariance*, conjecturing that stochastic gradient descent (SGD) solutions lie within the same basin of the loss landscape after applying appropriate permutations. This conjecture is supported by empirical studies that show that aligning neurons through permutation yields linear paths with near-zero loss barriers (Akash et al., 2022).

Inspired by mode connectivity, which posits that local minima found by stochastic gradient descent are connected by continuous low-loss paths (Garipov et al., 2018), we propose

Mode Path Fusion (MPF). MPF leverages Bezier curves to identify low-loss paths that unify knowledge from multiple models while minimizing task-specific degradation. Linear averaging often fails to adequately capture the subtleties of the loss landscape, especially when the solution manifolds of two models are misaligned. Prior works ((Draxler et al., 2019); (Garipov et al., 2018)) highlight that low-loss paths exhibit non-linear structures, suggesting the potential for improved solutions via curved trajectories. Building on this insight, MPF constructs carefully optimized curves—designed to minimize loss—that better preserve task-specific performance while integrating knowledge across models. By addressing the limitations of linear averaging and improving over OT-based merging, MPF provides a robust and flexible approach for continual learning.

Contributions. Our work builds on the theoretical and empirical foundations laid by these studies. Our contributions include:

- Proposing *Mode Path Fusion (MPF)*, a novel method leveraging Bezier curves to identify low-loss paths for model merging.
- Demonstrating MPF's effectiveness on MNIST and CIFAR-10 and comparing it against baselines.
- Extending the theoretical understanding of mode connectivity under permutation invariance and its application to neural network fusion.

2. Models and Methods

In this section, we first explain the merging methods, followed by their application within the Continual Learning framework.

2.1. Model Merging Procedures

We now detail the three model merging procedures employed.

Problem Formulation. Let $W_1, W_2 \in \mathbb{R}^{|network|}$ represent the weight vectors of neural networks M_1 and M_2, trained independently on datasets D_1 and D_2, respectively. Here, $|network|$ denotes the total number of weights in the neural network, and $\mathcal{L}(\cdot)$ is the loss function. The objective is to merge these networks into a new model $M_{\mathcal{F}}$ with weights $W_{\mathcal{F}}$, sharing the same architecture, while preserving high accuracy on both D_1 and D_2.

2.1.1. Naive Averaging

Naive averaging (AVG) simply averages model weights for each layer ℓ from 2 to L:

$$\boldsymbol{W}_{\mathcal{F}}^{(\ell,\ell-1)} \leftarrow \frac{1}{2}(\boldsymbol{W}_1^{(\ell,\ell-1)} + \boldsymbol{W}_2^{(\ell,\ell-1)}) \qquad (1)$$

2.1.2. Adding neuron alignment via Optimal Transport

We adopt the method proposed by Singh & Jaggi (2023), which introduces a novel layer-wise fusion strategy that leverages Optimal Transport (OT) to align neurons based on the similarity of their activations or incoming weights.

Consider n pre-trained models $\theta_1, ..., \theta_n$ to merge, each with L layers. Let us assume that we have already aligned the neurons in the previous layers and are now at some later ℓ. We denote by W_i^ℓ and $W_{\mathcal{F}}^\ell$ the weights of the ℓ-th layer in neural network θ_i and target model $\theta_{\mathcal{F}}$ respectively.

Aligning Incoming Edges via Transport Map The incoming edges are aligned by post-multiplying the weight matrix W_i^ℓ with the transportation matrix $\left(\Pi_i^{\ell-1}\right)^*$ obtained from the previous layer, normalized by the probability measure of the ℓ-th layer in the target model.

Cost Matrix for Neuron Alignment For each model θ_i, the neurons of the ℓ-th layer are aligned with respect to the target model $\theta_{\mathcal{F}}$ by solving OT problems using the cost matrix:

$$C_i^{\ell,\mathrm{OT}} := \left[\|w_g - \widehat{w}_{j,i}\|_2^2\right]_{g,j}, \qquad (2)$$

where w_g is the g-th row vector of weight matrix $W_{\mathcal{F}}^\ell$ and $\widehat{w}_{j,i}$ the j-th row vector of W_i^ℓ.

We repeat this alignment procedure layer-by-layer across the network.

OT fusion inherently addresses the permutation invariance problem by aligning neurons based on their functional similarity.

2.1.3. Leveraging Linear Mode Connectivity

We introduce our Mode Path Fusion (MPF) procedure that leverages the LMC property. Here, we specifically aim to merge the two networks into new model $M_{\mathcal{F}}$ with weights $W_{\mathcal{F}}$ by identifying a low-loss path between W_1 and W_2 in the parameter space.

Constructing a Low-Loss Path. We parametrize the path as a quadratic Bezier curve given by :

$$\phi_\theta(t) = (1-t)^2 W_1 + 2t(1-t)\theta + t^2 W_2 \qquad (3)$$

where parameter θ of the curve parametrization corresponds to the bend of the chain which we optimize and such that

$\phi_\theta(0) = W_1$ and $\phi_\theta(1) = W_2$

Given the curve parametrization $\phi_\theta(t) : [0,1] \rightarrow \mathbb{R}^{|network|}$, we search for parameter θ that minimize the expectation over a uniform distribution on $t \in [0,1]$.

$$\ell(\theta) = \int_0^1 \mathcal{L}(\phi_\theta(t))\, dt = \mathbb{E}_{t \sim U(0,1)}[\mathcal{L}(\phi_\theta(t))] \quad (4)$$

where $U(0,1)$ is the uniform distribution on $[0,1]$.

In practice, the loss is minimized by following the following procedure. At each iteration, sample $\tilde{t} \sim U(0,1)$ and compute an unbiased estimate of the gradient of $\ell(\theta)$ to make a gradient step for θ. The optimization continues until convergence, ensuring the path is of consistently low loss.

Deducing the Fused Model Weights Once the low-loss path is constructed, the final weights of the fused model are obtained by selecting the weights at the midpoint of the curve, $W_{\mathcal{F}} = \phi_\theta(0.5)$. Selecting the midpoint along the curve ensures that the merged model retains a balanced representation of the knowledge from both models, while minimizing loss across the training and test sets.

2.1.4. Base Models and General Setup

We conducted experiments using a suite of multi-layer perceptron (MLP) models, evaluating their performance on the sequential MNIST and CIFAR-10 datasets. These experiments were designed to systematically investigate the efficacy of merging and fusion strategies across diverse model architectures and dataset characteristics:

- **Architectures:** MLPNET (400, 200, 100), MLPLARGE (800, 400, 200), MLPHUGE (1024, 512, 256).
- **Optimization:** SGD with momentum (0.9) and weight decay (5×10^{-4}).
- **Learning Rate Schedule:** Constant for 50% of epochs, linearly decaying to 0.00007 between 50% and 90%, and stabilizing for the final 10%.
- **Evaluation Metric:** Accuracy on test datasets, averaged over five random seeds.

For both MNIST and CIFAR-10, each model was trained for 10 epochs.

Mode Path Fusion was implemented using Bézier parameterization with three bends. The endpoints of the Bézier curve correspond to the parameters of the two models being fused. The curve-finding process was optimized over 10 epochs using SGD with an initial learning rate of 0.07. This approach provides a continuous trajectory between the two models, enabling smooth transitions in parameter space.

OT-based merging employed the Earth Mover's Distance (EMD) exact solver to align and merge model weights. The EMD solver, configured without regularization, used model weights as the cost metric to compute optimal transport between model parameter distributions.

All datasets were normalized to have zero mean and unit variance to standardize input distributions and improve model training stability. For both MNIST and CIFAR-10, the default training and testing splits were utilized to ensure consistency with established benchmarks.

2.2. Continual Learning Procedure

For the methods, we focus on Class-Incremental Learning (CIL) which is a subfield of CL. In CIL the tasks consist of samples with a specific subset of all the classes.

The algorithm 1 describes the general procedure of the merging setup. The input is a sequence of tasks and a base-model. First the base-model is finetuned on task 1. The current model is initialzed to be model finetuned on task 1. For every new task the base-model is again finetuned to on the tasks data. The new model is then merged with the current model through the respective merging scheme. Finally the current model is updated to be the merged model.

For both naive merging and OT merging, we use the coefficient of $\frac{1}{t}$ for every new model. Through this all the models are weighed equally. Since MPF is not data-free we need to deploy a replay buffer to train the curve-model on. The base model used is MLPNet as described in section 2.1.4. The training procedure is also the same as described above.

Since the MPF has access to previous data we also investigate if the other schemes would also improve by this. We modify the algorithm to maintain a certain number of samples per class in a replay buffer. At the end of each iteration in the algorithm 1, the current model is finetuned on the data in the replay buffer.

3. Experiments & Results

We first analyze the performance of various model fusion strategies, specifically focusing on models trained under heterogeneous data distributions. We then present the results of the general continual learning procedure.

3.1. Fusion Under Heterogeneous Data Distributions

Setup: This setup follows the approach outlined in Singh & Jaggi (2019). To simulate a heterogeneous data split for MNIST digit classification, model A M_A is trained to specialize in recognizing a particular digit (e.g., digit 4) that is not part of the training data for the other model M_B. M_B is trained on 90% of the training data for all digits except the one that M_A specializes in, while M_A is also trained using the remaining 10% of the data. Both models have different initializations.

Quantitative Results: In Table 1, we present a comparison of the performance of the three fusion procedures. Our results show that Mode Path Fusion consistently outperforms the other methods across all models and datasets considered.

Figure 1 contains the visualizations of MPF fusing two MLPNet trained on MNIST. Unlike vanilla averaging, which results in a fused model located along a higher-loss path directly connecting the two models, MPF identifies a lower-loss trajectory between them, yielding a fused model that lies along this optimized path.

MNIST	MLPNet	MLPLarge	MLPHuge
Joint model	96.97 ± 0.20	97.44 ± 0.18	97.22 ± 0.21
Model A	91.68 ± 0.65	92.11 ± 0.36	91.92 ± 0.66
Model B	87.56 ± 0.21	87.67 ± 0.27	87.81 ± 0.15
AVG	81.30 ± 1.87	85.75 ± 0.41	86.19 ± 0.27
OT	80.27 ± 2.06	85.42 ± 0.56	85.96 ± 0.36
MPF (ours)	**97.32** ± 0.07	**97.52** ± 0.11	**97.76** ± 0.07

CIFAR-10	MLPNet	MLPLarge	MLPHuge
Joint model	34.12 ± 1.29	33.94 ± 1.37	35.22 ± 0.66
Model A	34.52 ± 2.45	35.27 ± 2.04	34.32 ± 1.98
Model B	33.42 ± 0.40	35.95 ± 0.88	35.02 ± 1.45
AVG	23.93 ± 0.60	27.06 ± 2.32	27.96 ± 1.89
OT	25.29 ± 1.04	27.78 ± 1.78	27.87 ± 0.90
MPF (ours)	**48.02** ± 0.25	**48.93** ± 0.43	**49.62** ± 0.39

Table 1: Performance comparison (test accuracy ± standard deviation %) of different fusion methods on the MNIST and CIFAR-10 datasets.

3.2. Continual Learning using Model Merging

To assess the performance of the merging schemes in continual learning (CL), we utilize the MNIST dataset to train five distinct submodels, each specialized in learning two different digits. After executing the continual learning algorithm, we extract the final model and evaluate its performance on the entire MNIST test set (see Table 2).

To examine whether other methods could similarly benefit from access to data, we evaluate the modified algorithm with a replay buffer. For this analysis, we explore buffer sizes of 16, 32, 64, 128, 256, and 512. The corresponding results are provided in Appendix C.

While MPF surpasses the baselines without fine-tuning, it fails to outperform them when all methods are provided access to the replay buffer.

SeqMNIST	MLPNet
AVG	36.34 ± 9.19
OT	36.48 ± 9.16
MPF: 500 samples per class (≈ 4%)	91.80 ± 0.41
Joint Model	98.56 ± 0.06

Table 2: Comparison of Accuracies [%] of different model merging schemes in continual learning.

4. Discussion

Mode Path Fusion (MPF) demonstrates superior performance in model merging compared to baselines, albeit with increased computational complexity due to minimum-loss curve computation. However, for general continual learning, this overhead is mitigated when compared to AVG or OT combined with fine-tuning. MPF's reliance on data for curve computation diverges from the goal of a fully data-free approach, raising concerns in scenarios where data storage is constrained or privacy is critical. Additionally, the use of a replay buffer partially conflicts with the strict class-incremental learning (CIL) assumption of non-shared data between tasks. While this design choice is effective for model merging, it reveals limitations in broader continual learning applications.

Future work should explore ways to reduce MPF's data dependency, such as leveraging compressed representations or sufficient statistics inspired by GEM (Lopez-Paz & Ranzato, 2017) and A-GEM (Chaudhry et al., 2018). Extending MPF to architectures like CNNs, transformers, and recurrent networks will further test its scalability and adaptability across diverse tasks and modalities.

5. Summary

We have demonstrated that merging independently trained models can effectively retain task-specific knowledge while supporting adaptation in a continual learning context.

Our experiments reveal that naive averaging often fails when data distributions diverge, whereas optimal transport (OT) alignment mitigates misalignment by aligning corresponding neurons across networks.

Most importantly, the proposed Mode Path Fusion (MPF) exploits a low-loss trajectory in parameter space to merge models without compromising their individual performance, outperforming baseline methods on both heterogeneous data splits and class-incremental tasks.

In CL, while MPF can surpass data-free baselines when using a replay buffer, it does not yet outperform other methods that also have access to data. Future work could explore strategies to minimize this dependency on external data.

References

Akash, A. K., Li, S., and Trillos, N. G. Wasserstein barycenter-based model fusion and linear mode connectivity of neural networks, 2022. URL https://arxiv.org/abs/2210.06671.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially no barriers in neural network energy landscape, 2019. URL https://arxiv.org/abs/1803.00885.

Entezari, R., Sedghi, H., Saukh, O., and Neyshabur, B. The role of permutation invariance in linear mode connectivity of neural networks, 2022. URL https://arxiv.org/abs/2110.06296.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns, 2018. URL https://arxiv.org/abs/1802.10026.

Goodfellow, I. J., Vinyals, O., and Saxe, A. M. Qualitatively characterizing neural network optimization problems, 2015. URL https://arxiv.org/abs/1412.6544.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima, 2017. URL https://arxiv.org/abs/1609.04836.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114 (13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Marczak, D., Twardowski, B., Trzciński, T., and Cygert, S. Magmax: Leveraging model merging for seamless continual learning, 2024. URL https://arxiv.org/abs/2407.06322.

Marouf, I. E., Roy, S., Tartaglione, E., and Lathuilière, S. Weighted ensemble models are strong continual learners, 2024. URL https://arxiv.org/abs/2312.08977.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks, 2022. URL https://arxiv.org/abs/1606.04671.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay, 2017. URL https://arxiv.org/abs/1705.08690.

Singh, S. P. and Jaggi, M. Model fusion via optimal transport, 2019. URL https://arxiv.org/abs/1910.05653.

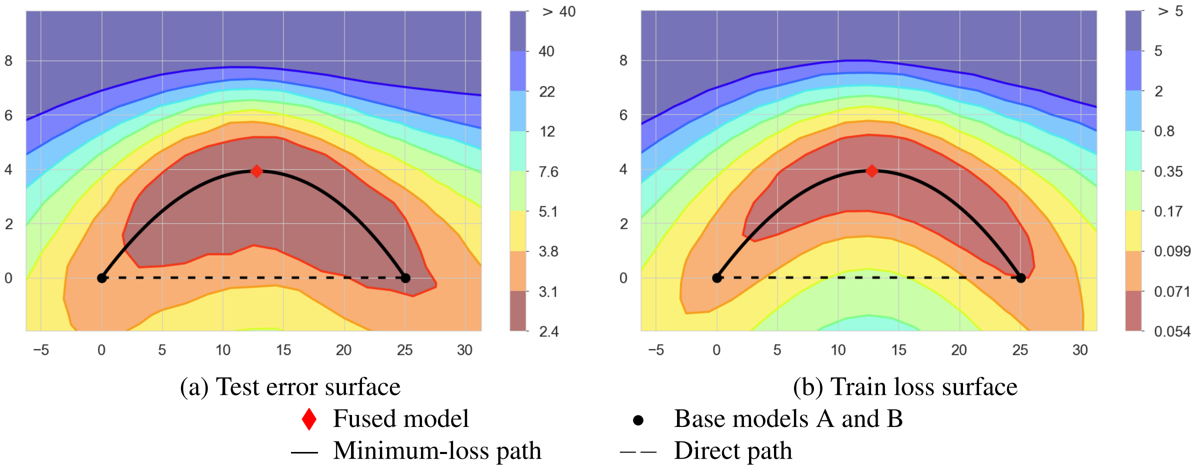
Singh, S. P. and Jaggi, M. Model fusion via optimal transport, 2023. URL https://arxiv.org/abs/1910.05653.

Venturi, L., Bandeira, A. S., and Bruna, J. Spurious valleys in two-layer neural network optimization landscapes, 2020. URL https://arxiv.org/abs/1802.06384.

Wang, L., Zhang, X., Su, H., and Zhu, J. A comprehensive survey of continual learning: Theory, method and application, 2024. URL https://arxiv.org/abs/2302.00487.

Webson, A., Loo, A. M., Yu, Q., and Pavlick, E. Are language models worse than humans at following prompts? it's complicated, 2023. URL https://arxiv.org/abs/2301.07085.

A. Loss landscape visualizations



(a) Test error surface

(b) Train loss surface

Figure 1: (a) Visualization of the test error surface, illustrating the fusion results of two MLPNet models trained on MNIST under the setup described in 3.1. (b) Visualization of the train loss surface as a function of network weights in a two-dimensional subspace, illustrating the fusion results of two MLPNet models trained on MNIST under the setup described in 3.1.

B. CL Merging algorithm

Algorithm 1 General Procedure of the Merging Setup

```
Input: Sequence of tasks $\{D_t\}_{t=1}^{N}$ and base-model $\theta_0$
Output: $\theta_{\text{final}}$
Initialization: $\theta_{\text{current}} \leftarrow \text{finetune}(\theta_0, D_1)$
for each new task $D_t$ for $t = 2, \ldots, N$ do
    $\theta_t \leftarrow \text{finetune}(\theta_0, D_t)$
    $\theta_{\text{merged}} \leftarrow \text{merge}(\theta_{\text{current}}, \theta_t)$
    $\theta_{\text{current}} \leftarrow \theta_{\text{merged}}$
end for=0
```

C. CL with Replay Buffer

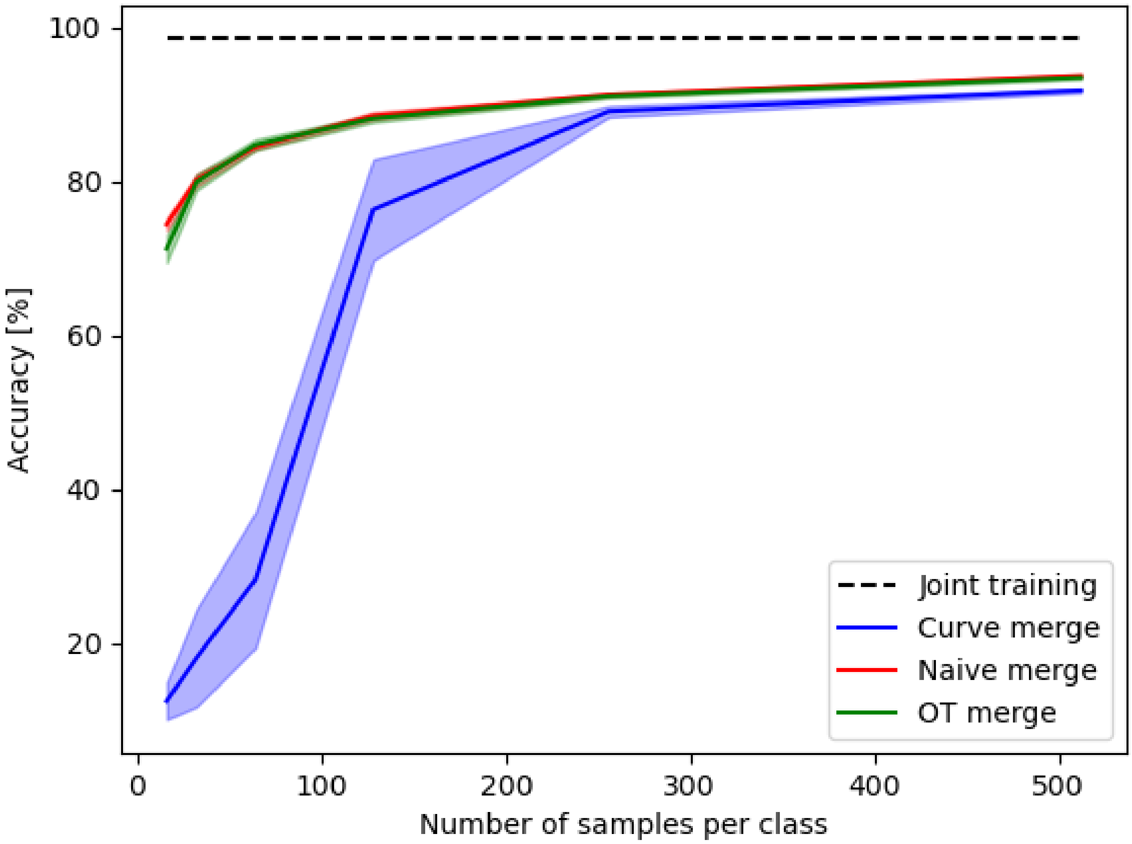


Figure 2: Performance of Mode Path Fusion on Sequential MNIST in relation to the number of samples saved per task. Mean and standard deviation from 5 different seeds.