Re-basin via implicit Sinkhorn differentiation

Fidel A. Guerrero Peña
fidel-alejandro.guerrero-pena@etsmtl.ca

Heitor Rapela Medeiros
heitor.rapela-medeiros.1@ens.etsmtl.ca

Thomas Dubail
thomas.dubail.1@ens.etsmtl.ca

Masih Aminbeidokhti
masih.aminbeidokhti.1@ens.etsmtl.ca

Eric Granger
eric.granger@etsmtl.ca

Marco Pedersoli
marco.pedersoli@etsmtl.ca

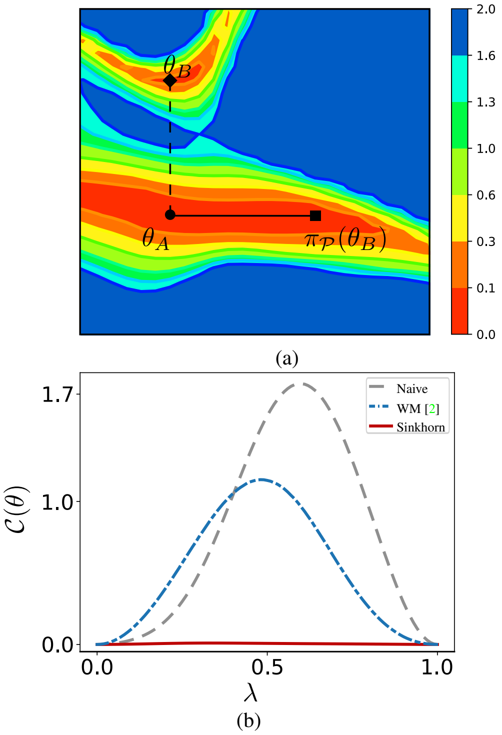
LIVIA, Dept. of Systems Engineering
ETS Montreal, Canada

Abstract

The recent emergence of new algorithms for permuting models into functionally equivalent regions of the solution space has shed some light on the complexity of error surfaces and some promising properties like mode connectivity. However, finding the permutation that minimizes some objectives is challenging, and current optimization techniques are not differentiable, which makes it difficult to integrate into a gradient-based optimization, and often leads to suboptimal solutions. In this paper, we propose a Sinkhorn re-basin network with the ability to obtain the transportation plan that better suits a given objective. Unlike the current state-of-art, our method is differentiable and, therefore, easy to adapt to any task within the deep learning domain. Furthermore, we show the advantage of our re-basin method by proposing a new cost function that allows performing incremental learning by exploiting the linear mode connectivity property. The benefit of our method is compared against similar approaches from the literature under several conditions for both optimal transport and linear mode connectivity. The effectiveness of our continual learning method based on re-basin is also shown for several common benchmark datasets, providing experimental results that are competitive with the state-of-art. The source code is provided at https://github.com/fagp/sinkhorn-rebasin.

1. Introduction

Despite the success of deep learning (DL) across many application domains, the loss surfaces of neural networks (NNs) are not well understood. Even for shallow NNs, the number of saddle points and local optima can increase exponentially with the number of parameters [4, 13]. The permu-



(a)

(b)

Figure 1. (a) Loss landscape for the polynomial approximation task [27]. θ_A and θ_B are models found by SGD. LMC suggests that re-basin the model θ_B would result in a functionally equivalent model $\pi_{\mathcal{P}}(\theta_B)$, with no barrier on its linear interpolation $(1-\lambda)\theta_A + \lambda\pi_{\mathcal{P}}(\theta_B)$. (b) Comparison of the cost in the linear path along λ before and after re-basin using weight matching (WM) [2] and our Sinkhorn. The dashed line in the figures corresponds with the naive path, and the solid line is the path after the proposed Sinkhorn re-basin. The blue line represents WM path.

tation symmetry of neurons in each layer allows the same function to be represented with many different parameter values of the same network. Symmetries imposed by these invariances help us to better understand the structure of the loss landscape [6, 11, 13].

Previous studies establish that minima found by Stochastic Gradient Descent (SGD) are not only connected in the network parameter's space by a path of non-increasing loss, but also permutation symmetries may allow us to connect those points linearly with no detriment to the loss [9, 11–13, 15, 24]. This phenomenon is often referred to as linear mode connectivity (LMC) [24]. For instance, Fig. 1a shows a portion of the loss landscape for the polynomial approximation task [27] using the method proposed by Li *et al.* [16]. θ_A and θ_B are two minima found by SGD in different basins with an energy barrier between the pair. LMC suggests that if one considers permutation invariance, we can teleport solutions into a single basin where there is almost no loss barrier between different solutions [2, 11]. In literature, this mechanism is called re-basin [2]. However, efficiently searching for permutation symmetries that bring all solutions to one basin is a challenging problem [11]. Three main approaches for matching units between two NNs have been explored in the literature [2]. Some studies propose a data-dependent algorithm that associates units across two NNs by matching their activations [2, 26]. Since activation-based matching is data dependent, it helps to adjust permutations to certain desired kinds of classes or domains [26]. Instead of associating units by their activations, one could align the weights of the model itself [2, 26], which is independent of the dataset, and therefore the computational cost is much lower. Finally, the third approach is to iteratively adjust the permutation of weights. In particular, Ainsworth *et al.* [2] have proposed alternating between models alignment and barrier minimization using a Straight-Through Estimator. Unfortunately, the proposed approaches so far are either non-differentiable [2, 11, 26] or computationally expensive [2], making the solution difficult to be extended to other applications, with a different objective. For instance, adapting those methods for incremental learning by including the algorithm for weight matching between two models trained on different domains is not trivial because of the difficulties in optimizing new objectives.

In this work, inspired by [21], we relax the permutation matrix with the Sinkhorn operator [1], and use it to solve the re-basin problem in a differentiable fashion. To avoid the high cost for computing gradients in the proposal of Mena *et al.* [21], we use the implicit differentiation algorithm proposed in [10], which has been shown to be more cost-effective. Our re-basin formulation allows defining any differentiable objective as a loss function.

A direct application of re-basin is the merger of diverse models without significantly degrading their performance [2, 5, 12, 13, 28]. Applications like federate learning [2], ensembling [12], or model initialization [5] exploit such a merger by selecting a model in the line connecting the models to be combined. To show the effectiveness of our approach, we propose a new continual learning algorithm that combines models trained on different domains. Our continual learning algorithm differs from previous state-of-art approaches [22] because it directly estimates a model at the intersection of previous and new knowledge, by exploiting the LMC property observed in SGD-based solutions.

Our main contribution can be summarized as follows:
(1) Solving the re-basin for optimal transportation using implicit Sinkhorn differentiation, enabling better differentiable solutions that can be integrated on any loss.
(2) A powerful way to use our re-basin method based on the Sinkhorn operator for incremental learning by considering it as a model merging problem and leveraging LMC.
(3) An extensive set of experiments that validate our method for: (i) finding the optimal permutation to transform a model to another one equivalent; (ii) linear mode connectivity, to linearly connect two models such that their loss is almost identical along the entire connecting line in the weights space; and (iii) learning new domains and tasks incrementally while not forgetting the previous ones.

2. Related work

Re-basin. Recently, in the NN community, re-basin has been demonstrating useful properties. The main goal of such re-basin approaches is to obtain functionally equivalent models in a different region of the weight space following some pre-defined objective. Permutation symmetries are a well-known example of transformations that allows performing re-basin. In particular, Entezari *et al.* [11] show that the invariances of NNs using random permutations on SGD solutions are likely to have almost zero barriers, and therefore the randomness in terms of permutations does not impact the quality of the final training result of the model. A simulated annealing-based algorithm was proposed for doing a re-basin with an elevated computational cost, making it impractical to use, especially for bigger models. Ainsworth *et al.* [2] proposed three new re-basin algorithms that rely on solving linear assignment problems to find permutation matrices that satisfy their encoded objective. The methods showed to perform well, especially in achieving linear mode connectivity. On the downside, new objectives are difficult to plug into their framework. Their solution uses greedy algorithms that do not guarantee finding the optimal solution, as shown in our experiments. Using a similar approach, Benzing *et al.* [5] found strong evidence that two random initialization of a NN after permutation can lead to a good performance, showing that the random initialization is already in the same loss valley during the initialization. Finally, [3] uses the concepts of

Wasserstein barycenter and Gromov-Wasserstein barycenter, offering a NN model fusion framework with insights about linear mode connectivity of SGD solutions. Even though the previous works presented solutions to perform re-basin by solving linear assignment problems, their approach fails to generalize well for other objectives. Using gradient descent-based algorithms seems to be a more suitable approach.

Mode connectivity. Mode connectivity is the task of finding low-barrier paths connecting models within the weight space. In their work, Garipov *et al.* [13] found that the local optima of deep learning models are connected by simple curves. The fast geometric ensembling method was proposed as an application for their proposal. Almost at the same time, [9] proposed a nudged elastic band-based method to construct continuous paths between minima of NN architectures. Finally, Frankle *et al.* [12] studied the sensitivity of different levels of SGD noise on NNs. These pioneering works are the basis for applications of mode connectivity, like [2, 22] and ours.

Incremental learning. Continual or incremental learning (IL) allows adapting models incrementally based on new training data without forgetting previous knowledge. Catastrophic forgetting [19, 25] occurs when a previously trained model that is fine-tuned on a new task loses information learned to perform well on previous datasets. To address this issue, Kirkpatrick *et al.* [14] proposes elastic weight consolidation (EWC), which reduces catastrophic forgetting by regularizing NN parameters with respect to the importance of the weights concerning the previous and actual tasks. Chaudhry *et al.* [8] proposed using a small number of samples for replay. Their experience replay (ER) helps to perform IL such that the performance on classification tasks is improved, even with a tiny episodic memory. Closer to our work, [22] proposes an LMC-based method with replay. Its solution is called mode connectivity SGD (MC-SGD), which relies on the assumption that there is always an existing solution that solves all seen tasks incrementally, and they are connected by a low-barrier and linear path. Furthermore, MC-SGD utilizes a replay buffer to remember previous tasks for IL. Its efficiency relies on exploring the linear path of low loss to constrain learning, thereby outperforming competitors like EWC when less data is available. However, such an approach has a high computational cost since it requires training independent models for the new task and merging them as a separate step with the model for previous knowledge. Also, the method has been shown to be difficult to reproduce or adapt to new benchmarks [20]. A compelling scenario for continual learning is using a linear mode connectivity path to keep learning and adapt the model without forgetting the previous knowledge. The trade-off between flatness of the LMC path and direction to adapt the loss can be tuned in a way that brings more stability or plasticity depending on the target final solution.

3. Re-basin via the Sinkhorn operator

Let $f_\theta(.)$ be a parameterized mapping where θ represents a vector of parameters within the solution space $\Theta \subset \mathbb{R}^d$, where d is the number of parameters in θ. In the deep learning context, f can be seen as a NN architecture, and f_θ is a model with weights θ. Here, we refer to θ as a model for simplicity. Consequently, the cost (or error) of a model for a given task can be defined as $\mathcal{C}(\theta) = \frac{1}{|\mathbb{T}|}\sum_{(x,y)\in\mathbb{T}} \mathcal{L}(f_\theta(x), y)$, where $(x, y) \in \mathbb{T}$ are input and expect output in the training set $\mathbb{T}$, and $\mathcal{L}$ is an appropriate supervised loss function.

A function f is invariant to a transformation if and only if the obtained transformed function is functionally equivalent to the original mapping. Note that such invariances can also be found between two functions within the family of parametric functions $\{f_\theta\}_{\theta\in\Theta}$. The permutation of neurons is a well-known example of such transformations applied to NNs that allow obtaining functionally equivalent models, i.e., $f_{\theta_A}(x) = f_{\theta_B}(x), \forall x$. These invariant models are obtained via the permutation transformation or re-basin function, here defined as $\pi\colon \Theta \to \Theta$, which shifts a model to a symmetric region of the loss landscape, $\mathcal{C}(\theta) = \mathcal{C}(\pi_{\mathcal{P}}(\theta))$. In this work $\mathcal{P} = (P_1, ..., P_h)$ is a transportation plan with P_i contained in the transportation polytope,

$$\Pi = \{P \in \mathbb{R}_+^{m\times n} | P\mathbb{1}_m = \mathbb{1}_n, P^T\mathbb{1}_n = \mathbb{1}_m\}, \tag{1}$$

where $\mathbb{1}_d = (1, ..., 1)^d$. Without loss of generality, let $f_\theta(x) = (\ell_h \circ ... \circ \ell_1)(x)$ be a NN defined as the composition of h layers such that $\ell_i(z) = \sigma(W_i z + b_i)$. Here, the weights W_i and biases b_i are the parameters of the network, $\theta = \{W_i, b_i\}_{i=1}^h$, and σ is a non-linear activation function. Then, the re-based model $\pi_{\mathcal{P}}(\theta)$ can be written as the functionally equivalent mapping:

$$\ell'_i(z) = \sigma(P_i W_i P_{i-1}^T z + P_i b_i), \tag{2}$$

where $P_i \in \Pi$ is a valid permutation matrix, and $P_h = P_0^T = \boldsymbol{I}$ is the identity matrix.

With regards to permutation invariance, Entezari *et al.* [11] conjectured that re-based SGD solutions are likely to have a low barrier within their linear interpolation $B(\theta_A, \theta_B) \approx 0$, where $B(.)$ is defined as:

$$\begin{aligned} B(\theta_A, \theta_B) = \sup_\lambda [\, [\mathcal{C}((1-\lambda)\theta_A + \lambda\theta_B)] - \\ [(1-\lambda)\mathcal{C}(\theta_A) + \lambda\mathcal{C}(\theta_B)]\,], \end{aligned} \tag{3}$$

and $\lambda \in (0, 1)$. This phenomenon is known as LMC [12], and it is a particular case of the widely studied mode connectivity [9, 13]. Notably, Ainsworth *et al.* [2] proposed three approaches that ratify the conjecture in [11] by finding a re-based model $\pi_{\mathcal{P}}(\theta_B)$ with LMC to a target model θ_A. Fig. 1 depicts the goal of such re-basin approaches. This figure shows two solutions for the task, θ_A and θ_B, found through SGD. As shown in Fig. 1b, the naive path between θ_A and θ_B has higher values of the barrier within the line $(1-\lambda)\theta_A+\lambda\theta_B, \lambda \in (0,1)$. On the other hand, and consistently with the results in [2,11], our re-based model $\pi_{\mathcal{P}}(\theta_B)$ achieves LMC by successfully finding a transportation plan $\mathcal{P}$ that shifts model θ_B to the same basin of model θ_A.

Although the seminal work by Ainsworth *et al.* [2] proposed a highly efficient approach for finding a permutation that minimizes the distances between models, their non-differentiable approach provides solutions that are difficult to be extended to other applications with a different objective. Specifically, their algorithms use a formulation based on the linear assignment problem (LAP) to find suitable permutations, meaning any new objective needs to be cast as a LAP which is a hard task in itself.

This work proposes a differentiable approach to perform a re-basin that defines any differentiable objective as a loss function. Here, we relax the rigid constraint of having a binary permutation matrix P, and consequently add an entropy regularizer $h(P) = -\sum P(\log P)$ to the original LAP as proposed by [21]. The final equation is then defined as:

$$S_\tau(X) = \arg\max_{P\in\Pi} \langle P, X\rangle_F + \tau h(P), \tag{4}$$

where is τ a factor that weights the strengths of the entropy regularization term.

The formulation in Eq. (4) is known as the Sinkhorn operator and can be efficiently approximated by:

$$S_\tau^{(0)}(X) = \exp\left(\frac{X}{\tau}\right),$$

$$S_\tau^{(t+1)}(X) = \mathcal{T}_c(\mathcal{T}_r(S_\tau^{(t)}(X))). \tag{5}$$

where $X \in \mathbb{R}^{m\times n}$ is a soft version of the permutation matrix, $\mathcal{T}_c(X) = X \oslash (\mathbb{1}_m \mathbb{1}_m^T X)$ and $\mathcal{T}_r(X) = X \oslash (X \mathbb{1}_n \mathbb{1}_n^T)$ are respectively the re-normalization of columns and rows of X, and $\oslash$ is the element-wise division. In their work, Mena *et al.* [21] proved that Eq. (5) converges to Eq. (4) when $t \to \infty$. However, in practice, only a finite number of iterations are needed to produce a suitable approximation.

Although the Sinkhorn operator is reasonably easy to implement within the NN layers, a significant drawback arises when considering the efficiency of its differentiation. We use the implicit differentiation algorithm proposed by Eisenberger *et al.* [10] to mitigate such an increase in the computational cost. Their method significantly increases the efficiency and also stability of the training process. The marginals of the generic formulation in [10] are defined as $\boldsymbol{a} = \mathbb{1}_m/m$ and $\boldsymbol{b} = \mathbb{1}_n/n$ to match the re-basin task.

Finally, our proposed Sinkhorn re-basin re-writes the original re-basin mapping in Eq. (2) as:

$$\ell_i'(z) = \sigma(S_\tau(P_i) W_i S_\tau(P_{i-1}^T) z + S_\tau(P_i) b_i). \tag{6}$$

where $P_i \in \mathbb{R}^{m\times n}$ is a differentiable permutation matrix.

To show the ability of our proposed Sinkhorn re-basin to minimize any differentiable objective, we provide three examples of cost functions that are minimized without changing the formulation in Eq. (6). These cost functions are used in the SGD framework to compute the gradient of the weights in our Sinkhorn re-basin network, and finally update the permutation matrices to minimize the objective. First, for a data-free objective like Weights Matching [2], we directly minimize the squared L2 distance defined as:

$$\mathcal{C}_{L2}(\mathcal{P}; \theta_A, \theta_B) = ||\theta_A - \pi_{\mathcal{P}}(\theta_B)||^2. \tag{7}$$

Given the ability of our method to use differentiable objectives, we introduce two other data-driven cost functions. Inspired by the Straight-Through Estimator in [2], we propose a differentiable midpoint cost function to minimize the barrier,

$$\mathcal{C}_{Mid}(\mathcal{P}; \theta_A, \theta_B) = \mathcal{C}\left(\frac{\theta_A + \pi_{\mathcal{P}}(\theta_B)}{2}\right). \tag{8}$$

Since minimizing the midpoint can lead to a multimodal cost path, i.e., lower cost value for $\lambda = 0.5$, and higher cost values elsewhere, we propose a cost function that minimizes the cost at random points within the line:

$$\mathcal{C}_{Rnd}(\mathcal{P}; \theta_A, \theta_B) = \mathcal{C}\left((1-\lambda)\theta_A + \lambda \pi_{\mathcal{P}}(\theta_B)\right), \tag{9}$$

with λ uniformly sampled at each iteration, $\lambda \sim U(0,1)$.

4. Re-basin incremental learning

A critical application of re-basin approaches is the ability to merge models without a significant performance reduction. Such merging is usually done by selecting a model in their connecting line. Applications like federate learning [2], ensembling [12], or model initialization [5] have been explored recently for merging models trained on the same task. Here we consider merging based on re-basin by proposing a new incremental learning approach that combines models trained on different domains or classes. Our proposed approach relies on a stability-plasticity hyperparameter α, allowing the user to choose the balance between forgetting and incorporating new knowledge.

Let θ_0 be an initial model trained over dataset $\mathbb{T}_0$, and $\mathcal{T} = \{\mathbb{T}_1, \mathbb{T}_2, ...\}$ be a stream of data, where $\mathbb{T}_i = \{(x_{ij}, y_{ij}) | x_{ij} \in \mathbb{X}_i, y_{ij} \in \mathbb{Y}_i\}, 1 \leq i \leq N_i$, is a supervised dataset with input $\mathbb{X}_i$, output $\mathbb{Y}_i$, and N_i data points. Note that the sets $\mathbb{T}_i$ are also known in incremental learning literature as tasks, but these are not limited to task incremental learning scenarios, and also include domain and

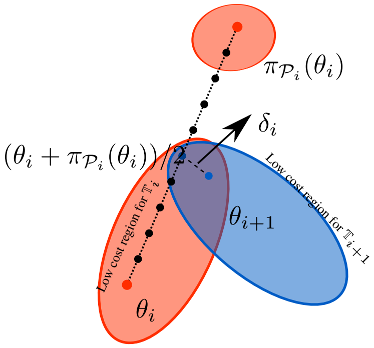

$\pi_{\mathcal{P}_i}(\theta_i)$
δ_i
$(\theta_i + \pi_{\mathcal{P}_i}(\theta_i))/2$
Low cost region for $\mathbb{T}_i$
Low cost region for $\mathbb{T}_{i+1}$
θ_{i+1}
θ_i

Figure 2. Graphical representation of the intersection of low curvature regions of the loss landscape for task $\mathbb{T}_i$ (red) and $\mathbb{T}_{i+1}$ (blue). The goal of our method is to find a re-basin $\pi_{\mathcal{P}_i}(\theta_i)$ that transverses the multitask region using LMC. The final model θ_{i+1} (Eq. (12)) is found in the surroundings of the line by adding a learnable residual δ_i.

class incremental learning. Incremental or continual learning seeks to incorporate the new knowledge $\mathbb{T}_{i+1}$ into the model θ_i without forgetting how to perform correctly in previous datasets $\mathbb{T}_0, ..., \mathbb{T}_i$. To this end, the continual learning community has proposed approaches that exploit the fact that multitask low curvature regions usually appear at the intersection of low curvature regions for individual tasks [14, 22] (see Fig. 2 for a visual reference). In particular, the Mirzadeh *et al.* [22] approach uses a two steps training, where the model θ_{i+1} is first trained over dataset $\mathbb{T}_{i+1}$, and then a mode connectivity-based merging finds the model with low loss value on both new and previous knowledge.

In contrast with [22], our approach directly estimates a model at the intersection of previous and new knowledge by exploiting our differentiable method to obtain the LMC observed in SGD-based solutions. Similarly to approaches introduced in the last section, our method looks for a re-basin of the given model that minimizes a given objective. For continual learning purposes, a new cost function is introduced such that, like in Eq. (8), the cost of the model in the middle of the line $(1-\lambda)\theta_i + \lambda\pi_{\mathcal{P}_i}(\theta_i), \lambda \in (0,1)$, is minimized for dataset $\mathbb{T}_{i+1}$ (see Fig. 2). In such a continual learning scenario, the middle point is the furthest model in the line from high stability points θ_i and $\pi_{\mathcal{P}_i}(\theta_i)$, noting that the endpoints are solutions for previous tasks, but not for the new one. The smooth nature of the loss landscape implies that neighboring models have similar costs for all tasks, thus it is important to select the farthest model from the extremes. However, constraining the solution space to models within the line yields a solution that performs well on the previous task, but does not allow optimal performance on the new task, thus affecting the training plasticity. Similarly to Kirkpatrick *et al.* [14], we find well-behaved models for $\mathbb{T}_{i+1}$ in the neighborhood of our optimization target. This is done by adding a regularization term that minimizes the l_2 norm of the residual vector δ_i. Finally, the proposed cost is calculated as:

$$\mathcal{C}_{CL}(\delta_i, \mathcal{P}_i; \theta_i) = \mathcal{C}\left(\frac{\theta_i + \pi_{\mathcal{P}_i}(\theta_i)}{2} + \delta_i\right) + \beta||\delta_i||^2. \quad (10)$$

During the learning phase, the underlying optimization problem finds:

$$\delta_i^*, \mathcal{P}_i^* = \underset{\delta_i, \mathcal{P}_i}{\arg\min}\, \mathcal{C}_{CL}(\delta_i, \mathcal{P}_i; \theta_i), \quad (11)$$

where δ_i^* and $\mathcal{P}_i^*$ are found at the same time. Note that the cost in Eq. (10) can be computed over any knowledge dataset. This work uses a replay method by taking the average of Eq. (10) for current and previous datasets.

Combining a model with a balance between previous and new knowledge should be obtained after minimizing the cost in Eq. (10). Here, the incremented model at episode $i+1$ is defined as:

$$\theta_{i+1} = (1-\alpha)\theta_i + \alpha\pi_{\mathcal{P}_i}(\theta_i) + \delta_i, \quad (12)$$

where α is a hyper-parameter controlling the balance between plasticity and stability. Note that values of α near 0.5 favor higher plasticity, while values around 0.0 and 1.0 give more importance to previous knowledge.

5. Experimental results and analysis

The experimental procedure for comparing re-basin approaches follows the same one used by Ainsworth *et al.* [2], while the continual learning experiments follow the standard experimental procedure in incremental learning literature [22]. For all experiments, the mean and standard deviation of results is reported and obtained over independent runs with different seeds. We used the original implementation provided by the authors in all cases. The effect of re-basin is studied for both classification and regression tasks. Mnist and Cifar10 datasets were used for image classification, while the polynomial approximation problems from [27] were used for regression. Feedforward NNs with 2 to 8 layers were explored as backbone architectures. In all our experiments, $t = 20$ and $\tau = 1.0$ were used, as proposed in [21] for the Sinkhorn operator. Furthermore, the corresponding performance measures and hyper-parameters are summarized in each subsection. Additional details on the experimental methodology, e.g., dataset, protocol, and performance measures, as well as other experimental settings, i.e., convolutional architectures, are provided in the supplementary materials.

5.1. Finding the optimal transport

In this experiment, we measure the ability of both Weights Matching (WM) [2] and our Sinkhorn re-basin to

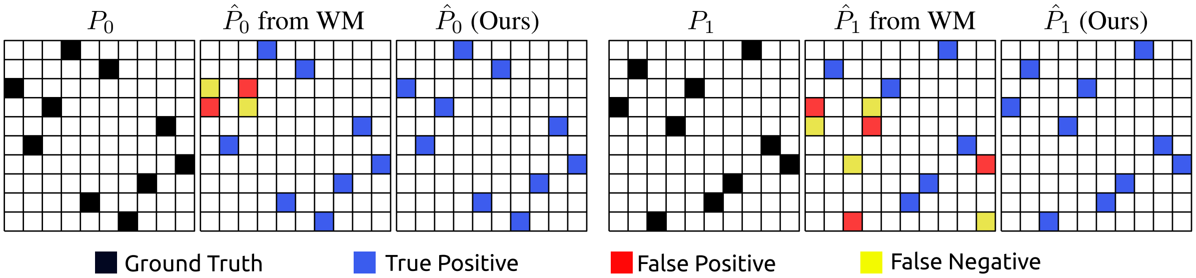


Figure 3. Estimated permutation matrices via weight matching (WM) [2] and the proposed Sinkhorn re-basin. P_i refers to the expected 10×10 permutation matrix with ones represented in black and zeros in white. The estimated permutations matrix $\hat{P}_i$ shows matching permutations as blue squares and miss-matchings in red and yellow. The permutation matrices $P_i \in \mathbb{R}^{10\times 10}$ correspond with transportation plans of layer i, with each layer containing 10 neurons. These matrices correspond with actual permutation matrices from the experiment with random initialization and 2 hidden layers.

find the optimal permutation. Similar to our $\mathcal{C}_{L2}$, the WM method minimizes the distance between the re-based and target models by solving a LAP. The cost function used for our re-basin is the squared L2 distance between models (Eq. (7)). Note that the objectives are not data-driven, and therefore we only measure the ability of each algorithm to reach the global minima without any context. Each method received a model and a randomly permuted version of it to find the permutation matrices that originated the target re-basin.

For our purposes, 9 datasets are created, each one containing 50 models and a random re-basin, $\mathbb{T}_i = \{(\theta_{ij}, \pi_{\mathcal{P}}(\theta_{ij})) \mid P_k \sim \mathrm{U}(\Pi), \forall P_k \in \mathcal{P}\}$, where $1 \leq j \leq 50$ is the index of the model within dataset $\mathbb{T}_i$ and $1 \leq k \leq h$ is the number of hidden layers in the NN. Note that we select random permutation matrices following a uniform distribution, $\mathrm{U}(\Pi)$. NNs with two, four, and eight hidden layers were used as base architecture. Additionally, we tested three types of initializations – random initialization with weights following a normal distribution $\mathcal{N}(0, 1)$, hereafter called *Rnd*, and models trained in a third and first-degree polynomial approximation problem, named *Pol3* and *Pol1* respectively. The 9 data set configurations lie within the combination $\{$Rnd, Pol3, Pol1$\}\times\{$2 hidden, 4 hidden, 8 hidden$\}$.

The Sinkhorn re-basin model was updated using the Adam optimizer with an initial learning rate of 0.1, and for a maximum of 5 iterations, using early stopping in case of convergence. Tab. 1 summarizes the L1 norm between weights after re-basin, $|\pi_{\hat{\mathcal{P}}}(\theta) - \pi_{\mathcal{P}}(\theta)|$, where $\hat{\mathcal{P}}$ and $\mathcal{P}$ are the estimated and optimal transportation plan, respectively. As expected, our proposal always finds the optimal permutation thanks to its ability to simultaneously look at all permutation matrices. In contrast, the WM results fall short for some scenarios. It is worth mentioning that these results match the ones obtained by Ainsworth *et al.* in [2]. In general, the WM algorithm seems to be affected by random initialization, while increasing the network's capacity improves its ability to reach the global minimum. We hypothesize that this is an effect of using a greedy algorithm that optimizes the objective for different layers at each iteration. A deeper inspection of the estimated permutation matrices shows that WM reaches local minima close to the expected re-basin, with only a few misplaced permutations (see Fig. 3). A more challenging alignment scenario and convergence analysis are shown in supplementary material.

5.2. Linear mode connectivity

We measure the ability of our method to find linear connectivity between SGD modes after re-basin one of them. For this experiment, four datasets are employed – first and third-degree polynomial regression tasks [27], along with the classical classification benchmarks, Mnist and Cifar10. Our experiment follows a similar setup to the one described by [2], i.e., two networks were trained over the same dataset, and we performed the re-basin of one of them, hoping to reach the same basin as the unchanged model. The experiment was repeated 50 times for every dataset, and each method saw the same two networks. To measure the ability of the different approaches to find LMC, we use the Barrier [12] (Eq. (3)) and Area Under the Curve (AUC) over

Method	Init	2 hidden ↓	4 hidden ↓	8 hidden ↓
WM [2]	Rnd	6.05±9.17	4.12±6.58	0.50±1.55
$\mathcal{C}_{L2}$ (Ours)		0.00±0.00	0.00±0.00	0.00±0.00
WM [2]	Pol3	0.57±2.84	0.07±0.46	0.01±0.10
$\mathcal{C}_{L2}$ (Ours)		0.00±0.00	0.00±0.00	0.00±0.00
WM [2]	Pol1	0.27±0.94	0.00±0.00	0.00±0.00
$\mathcal{C}_{L2}$ (Ours)		0.00±0.00	0.00±0.00	0.00±0.00

Table 1. L1 distance between the estimated and expected re-basin with different network initializations and depths. Distances are scaled $\times 10^3$.

Method	First degree polynomial		Third degree polynomial		Mnist		Cifar10	
	AUC ↓	Barrier ↓	AUC ↓	Barrier ↓	AUC ↓	Barrier ↓	AUC ↓	Barrier ↓
Naive	0.31±0.38	0.62±0.71	0.25±0.15	0.58±0.33	0.34±0.08	1.07±0.21	0.73±0.12	1.23±0.18
WM [2]	0.16±0.15	0.32±0.28	0.19±0.26	0.41±0.57	0.01±0.00	0.03±0.01	0.13±0.02	0.27±0.04
$\mathcal{C}_{L2}$ (Ours)	0.05±0.06	0.10±0.12	0.05±0.06	0.12±0.12	0.01±0.00	0.02±0.00	0.09±0.02	0.19±0.03
STE [2]	0.11±0.10	0.23±0.22	0.09±0.07	0.24±0.23	0.01±0.00	0.01±0.01	0.08±0.01	0.15±0.02
$\mathcal{C}_{Mid}$ (Ours)	0.03±0.02	0.07±0.05	0.05±0.04	0.17±0.17	**0.00±0.00**	**0.00±0.00**	**0.02±0.01**	**0.05±0.01**
$\mathcal{C}_{Rnd}$ (Ours)	**0.01±0.01**	**0.03±0.02**	**0.01±0.01**	**0.03±0.03**	**0.00±0.00**	0.01±0.00	**0.02±0.01**	0.06±0.01

Table 2. AUC and loss Barrier results of linear mode connectivity for regression datasets (first and third-degree polynomial approximation), and classification datasets (Mnist and Cifar10). The WM method and our Sinkhorn with $\mathcal{C}_{L2}$ belong to the data-free category, while STE, Sinkhorn with $\mathcal{C}_{Mid}$ and $\mathcal{C}_{Rnd}$ are data-driven.

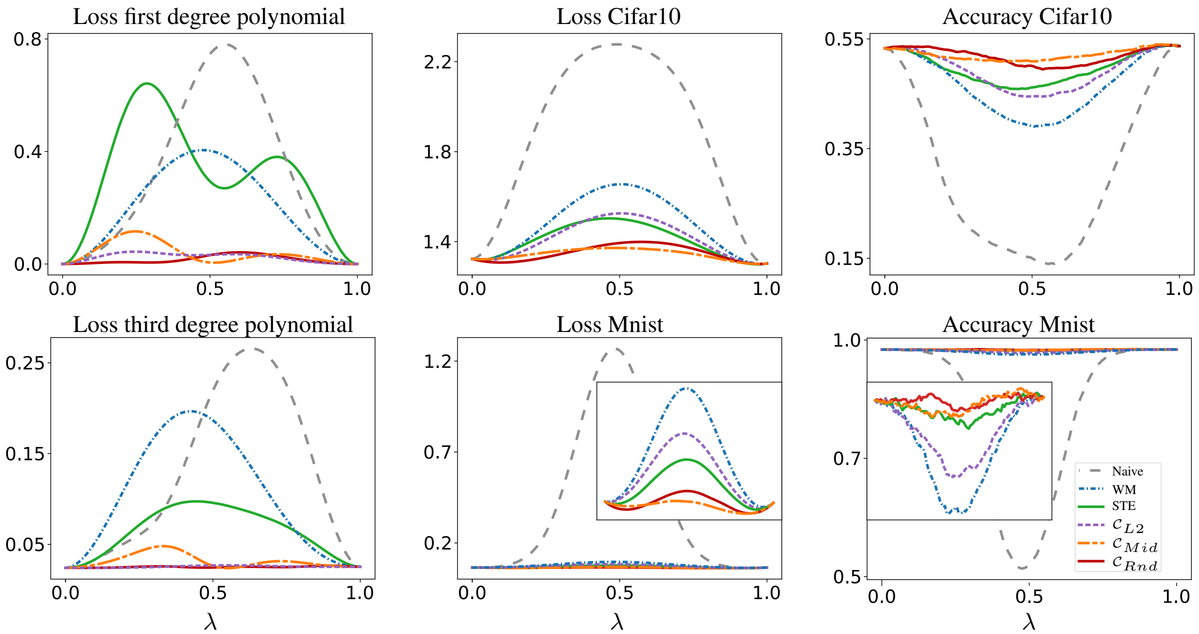


Figure 4. Example of linear mode connectivity achieved by WM [2], STE [2], and our Sinkhorn re-basin with $\mathcal{C}_{L2}$, $\mathcal{C}_{Mid}$, and $\mathcal{C}_{Rnd}$ costs for a NN with two hidden layers. Accuracy and loss are shown for the Mnist and Cifar10 classification, while only the L2 loss is shown for the regression tasks. For Mnist, we include an amplified version of the loss and accuracy for better comparison.

the estimated cost curve within the linear path. Both measures achieve the best performance when a method provides a low value, with a lower bound of 0.

We compare our approach with different objectives – L2 (Eq. (7)), Middle point (Eq. (8)), and Random point (Eq. (9)) with the recently introduced WM and Straight-Through Estimator (STE) [2]. Tab. 2 summarizes the performance of the methods for each dataset. All experiments used a NN with two hidden layers. The table shows that our methods outperform state-of-art methods with (the first three rows) and without considering the data (last three rows) during the re-basin. Specifically, our $\mathcal{C}_{L2}$ method exceeded WM for both AUC and Barrier measures for all datasets, except Mnist, where no significant differences were observed. Our other proposals outperform the state-of-art STE approach for AUC and Barrier in the data-driven category. In particular, our $\mathcal{C}_{Rnd}$ loss showed the best results, comparable to our $\mathcal{C}_{Mid}$ for more challenging scenarios like Cifar10. Note that STE and our $\mathcal{C}_{Mid}$ have the same objective, and their difference in performance lies in the solver. Additionally, our $\mathcal{C}_{L2}$, $\mathcal{C}_{Mid}$, and $\mathcal{C}_{Rnd}$ share the same architecture but with various loss functions, and their discrepancy in performance lies in the objective. As a general point, all methods significantly improve the naive path.

We show the obtained loss and accuracy curves over the linear path before doing a re-basin (naive) and after applying WM, STE, and Sinkhorn with $\mathcal{C}_{L2}$, $\mathcal{C}_{Mid}$, and $\mathcal{C}_{Rnd}$ methods in Fig. 4. Please note that our method can also

	Rotated Mnist		Split Cifar100	
Method	Accuracy ↑	Forgetting ↓	Accuracy ↑	Forgetting ↓
Finetune	46.28±1.01	0.52±0.01	35.41±0.95	0.49±0.01
EWC [14]	59.92±1.71	0.34±0.02	50.50±1.33	0.24±0.02
LwF [17]	61.86±3.66	0.29±0.06	41.43±4.06	0.51±0.01
A-GEM [7]	68.47±0.90	0.28±0.01	44.42±1.46	0.36±0.01
Rebasin /w replay (Ours)	**78.14±0.50**	**0.12±0.01**	**51.34±0.74**	**0.07±0.02**
Joint training	90.84±4.30	0.00	60.48±0.54	0.00

Table 3. Performance of our proposed and state-of-art methods on the continual learning benchmark datasets over 20 episodes.

perform re-basin of convolutional architectures with residual connections, as shown in supplementary materials.

5.3. Incremental learning application

This experiment seeks to compare our method with other well-known continual learning approaches from the literature. Since our proposal can fit the regularization techniques that use replay, we compare it with different algorithms within this category. In particular, we compared 3 regularization-based approaches – elastic weight consolidation (EWC) [14], learning without forgetting (LwF) [17], and average gradient episodic memory (A-GEM) [7]. The average accuracy was calculated to measure the overall performance of model θ_i in the first episodes $\mathbb{T}_j$. In addition, the forgetting measure averages the forgetting in terms of accuracy for each domain or task in the episode.

Given the variety of libraries and implementations, we limited our comparison to reproducible methods that could be used in the Avalanche environment [18]. All measurements, benchmarks, networks, and algorithms, including our own, were implemented using the framework. While we attempted to incorporate other recent approaches like MC-SGD [22] and Stable SGD [23] into Avalanche, a high discrepancy was observed w.r.t. their reported results and, therefore, we did not include them into our study. Difficulties in adapting MC-SGD to new conditions have also been observed by other authors [20].

We focused our experiments on low episodic memory scenarios, using only five examples per class for both benchmarks in methods that rely on memory replay (A-GEM and our method). We used a NN with one hidden layer and 256 neurons for the Rotated Mnist benchmark. For the Split Cifar100 benchmark, a multi-head ResNet18 was used following the settings in [22, 23]. In this experiment, we only apply the re-basin to linear layers.

Tab. 3 shows methods' accuracy and forgetting performance on the benchmarks Rotated Mnist and Split Cifar-100 datasets using 20 episodes. Our method outperforms the others and still performs better or comparable to those reported in [23]. The reader should pay special attention to the low values of forgetting achieved using our re-basin approach. This is a consequence of setting the value of $\alpha = 0.8$ when fusing the models (Eq. (12)). A-GEM is ranked second in accuracy and forgetting for Rotated Mnist. LwF showed a similar forgetting to A-GEM in this benchmark. On Split Cifar100, EWC ranked second for both measures. Despite having similar accuracy to our approach, the high forgetting value suggests stability issues. In general, the method showed to be robust to the values α and β. We included the corresponding ablation in supplementary materials.

6. Conclusion

This work proposes a new method based on the Sinkhorn operator to estimate permutation matrices that make two neural network models equivalent. With respect to previous work, such as weight matching, our method is (i) more flexible because it is differentiable and can be applied with any loss, (ii) estimates the permutations for all layers at the same time, avoiding getting stuck in local minima, (iii) more accurate, as shown in our experimental evaluation on well-known benchmarks. First, our experiments yielded perfect results when our approach was evaluated to produce the optimal permutation between a model and its artificially permuted transformation. We have also used our approach for linear mode connectivity, showing better connectivity (lower loss barrier) than weight matching. Finally, we showed that our efficient and differentiable approach for re-basin can easily be applied to the challenging task of continual learning, producing results comparable to, or better than state-of-art approaches. As a limitation to our work, we observed from our experiments and analysis of the literature that linear assignment problems solved with greedy Hungarian-based approaches are generally more efficient in terms of memory than the Sinkhorn operator.

Acknowledgements: This work was supported by Distech Controls Inc., the Natural Sciences and Engineering Research Council of Canada, and the Digital Research Alliance of Canada.

References

[1] Ryan Prescott Adams and Richard S Zemel. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, 2011. 2

[2] Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging Models modulo Permutation Symmetries. *arXiv preprint arXiv:2209.04836*, 2022. 1, 2, 3, 4, 5, 6, 7

[3] Aditya Kumar Akash, Sixu Li, and Nicolás García Trillos. Wasserstein Barycenter-based Model Fusion and Linear Mode Connectivity of Neural Networks. *arXiv preprint arXiv:2210.06671*, 2022. 2

[4] Peter Auer, Mark Herbster, and Manfred KK Warmuth. Exponentially many local minima for single neurons. *Advances in Neural Information Processing Systems*, 8, 1995. 1

[5] Frederik Benzing, Simon Schug, Robert Meier, Johannes von Oswald, Yassir Akram, Nicolas Zucchet, Laurence Aitchison, and Angelika Steger. Random initialisations performing above chance and how to find them. *arXiv preprint arXiv:2209.07509*, 2022. 2, 4

[6] Johanni Brea, Berfin Simsek, Bernd Illing, and Wulfram Gerstner. Weight-space symmetry in deep networks gives rise to permutation saddles, connected by equal-loss valleys across the loss landscape. *arXiv preprint arXiv:1907.02911*, 2019. 2

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations*, 2019. 8

[8] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 3

[9] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning*, pages 1309–1318. PMLR, 2018. 2, 3

[10] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, Florian Bernard, and Daniel Cremers. A Unified Framework for Implicit Sinkhorn Differentiation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 509–518, 2022. 2, 4

[11] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks. In *International Conference on Learning Representations*, 2022. 2, 3, 4

[12] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. 2, 3, 4, 7

[13] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in Neural Information Processing Systems*, 31, 2018. 1, 2, 3

[14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 3, 5, 8

[15] Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[16] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, 2018. 2

[17] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 8

[18] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas Tolias, Simone Scardapane, Luca Antiga, Subutai Amhad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: an End-to-End Library for Continual Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2nd Continual Learning in Computer Vision Workshop, 2021. 8

[19] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 3

[20] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. *arXiv preprint arXiv:2112.09153*, 2021. 3, 8

[21] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning Latent Permutations with Gumbel-Sinkhorn Networks. In *International Conference on Learning Representations*, 2018. 2, 4, 5

[22] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear Mode Connectivity in Multitask and Continual Learning. In *International Conference on Learning Representations*, 2021. 2, 3, 5, 8

[23] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020. 8

[24] Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[25] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990. 3

[26] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020. 2

[27] Johannes von Oswald, Christian Henning, Benjamin F. Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020. 1, 2, 5, 6

[28] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020. 2