Optimizing Mode Connectivity for Class Incremental Learning

Haitao Wen [1] **Haoyang Cheng** [1] **Heqian Qiu** [1] **Lanxiao Wang** [1] **Lili Pan** [1] **Hongliang Li** [1]

Abstract

Class incremental learning (CIL) is one of the most challenging scenarios in continual learning. Existing work mainly focuses on strategies like memory replay, regularization, or dynamic architecture but ignores a crucial aspect: mode connectivity. Recent studies have shown that different minima can be connected by a low-loss valley, and ensembling over the valley shows improved performance and robustness. Motivated by this, we try to investigate the connectivity in CIL and find that the high-loss ridge exists along the linear connection between two adjacent continual minima. To dodge the ridge, we propose parameter-saving OPtimizing Connectivity (OPC) based on Fourier series and gradient projection for finding the low-loss path between minima. The optimized path provides infinite low-loss solutions. We further propose EOPC to ensemble points within a local bent cylinder to improve performance on learned tasks. Our scheme can serve as a plug-in unit, extensive experiments on CIFAR-100, ImageNet-100, and ImageNet-1K show consistent improvements when adapting EOPC to existing representative CIL methods. Our code is available at https://github.com/HaitaoWen/EOPC.

1. Introduction

Continual learning is essential for intelligent machines to achieve dynamic adaptation (Ashfahani & Pratama, 2019; Lesort et al., 2020), knowledge accumulation (Caccia et al., 2020; Jin et al., 2021), and analogical reasoning (Hayes & Kanan, 2021). Class incremental learning (CIL) as one of the most challenging scenarios requires the model incrementally learn a sequence of new tasks without the information of task identification (Van de Ven & Tolias, 2019; Delange et al., 2021). However, when adapting an unpolished model to the above scenario, the performance of previous tasks will significantly degrade, i.e., catastrophic forgetting phenomenon (McCloskey & Cohen, 1989; Robins, 1995).

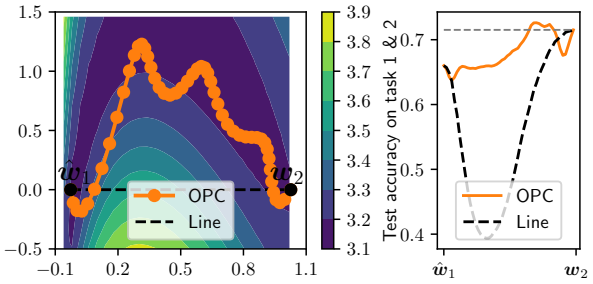

Figure 1. **Left:** The path found by OPC and the linear path connect minima $\hat{\boldsymbol{w}}_1$ and $\boldsymbol{w}_2$ in training loss landscape of task 1. **Right:** Testing accuracy curves along OPC and the linear path on task 1 and task 2. It can be seen that OPC locates in a lower-loss region and dodges the high-loss ridge compared with the linear path. Because of this low-loss retention property, an interval along OPC can achieve a higher performance than endpoints (the part above the gray dashed line). Besides, the ensembling method EOPC works in this local interval. Results are obtained from the first two of six tasks on CIFAR-100 with PODNet (Douillard et al., 2020).

To achieve CIL, different methods have been proposed in recent years. *Memory replay* uses a tiny episodic memory(Rebuffi et al., 2017; Liu et al., 2020) or the generative network (Kemker & Kanan, 2017; van de Ven et al., 2020) to replay samples from the distribution of previous tasks when learning a new task. *Regularization* is based on the idea of constraining important parameters to change (Kirkpatrick et al., 2017; Zenke et al., 2017) or distilling features from the previous model (Hou et al., 2019; Douillard et al., 2020; Simon et al., 2021; Kang et al., 2022). However, the parameter regularization methods rely on the approximation of local region around minimum and they are limited by the shift of distribution between tasks, which usually can not achieve good performance in CIL. *Dynamic architecture* follows two paradigms: one is parameter isolation (Serra et al., 2018) and the other is architecture expansion (Liu et al., 2021). Although remarkable advances have been achieved, none of the above work explores the aspect of mode connectivity in CIL (Garipov et al., 2018).

[1]University of Electronic Science and Technology of China. Correspondence to: Haitao Wen <haitaowen@std.uestc.edu.cn>, Heqian Qiu <hqqiu@std.uestc.edu.cn>, Lanxiao Wang <lanxiao.wang@std.uestc.edu.cn>, Hongliang Li <hlli@uestc.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

In contrast to the traditional view of loss landscape that regions of low loss are disconnected (Choromanska et al., 2015), mode connectivity (Garipov et al., 2018; Draxler et al., 2018) is the phenomenon that different minima (a.k.a., modes) can be connected by low-loss paths. The connectivity is modeled by parametric paths or a multi-dimensional volume between modes, such as Bezier curve (Garipov et al., 2018), nudged elastic band (NEB) (Henkelman & Jónsson, 2000; Draxler et al., 2018), and simplicial complexes (Benton et al., 2021). The reason why we choose mode connectivity as the breakthrough point of CIL is the low-loss retention property, *when parameters travel along the path, performance will not change too much compared with the endpoints, this is what continual learning needs!* Therefore, some pioneers have begun to cultivate this field. (Mirzadeh et al., 2020) found that there exists linear mode connectivity between continual minima and multitask minima under the scenario of task incremental learning (TIL). However, they are limited to the TIL scenario and linear connection, which may not scale to CIL.

In this paper, we try to investigate the connectivity between two adjacent continual minima in CIL and try to find a low-loss path between them for improving the performance of learned tasks. Our contributions can be summarized as follows:

- Linear interpolation experiments with different ways of initialization between minima are carried out and find that the high-loss ridge exists along the linear connection;
- A low-loss path finding method OPC based on the Fourier series and gradient projection is proposed, Figure 1 and experiments in Section E show OPC is an efficient and parameter-saving method;
- A performance-boosting method EOPC based on ensembling parameters within a local bent cylinder is proposed;
- Extensive experiments on CIFAR-100, ImageNet-100, and ImageNet-1K show significant improvements when adapting EOPC to representative CIL methods, e.g., boosting PODNet by 1.39%, 2.22%, and 3.73% for 5, 10, and 25 steps of increments on ImageNet-1K respectively.

2. Related Work

Class Incremental Learning as a challenging scenario in continual learning has gained increasing attention in recent years. According to the taxonomy in (Delange et al., 2021), we can divide existing methods into three categories. *Regularization-based* methods commonly use distillation techniques to constrain output activations or intermediate features consistent with the previous model. iCaRL (Rebuffi et al., 2017), LUCIR (Hou et al., 2019), PODNet (Douillard et al., 2020), GeoDL (Simon et al., 2021), and AFC (Kang et al., 2022) respectively constrains output possibilities, input embeddings, pooled features, projected features, and weighted features. *Memory replay* methods rehearse previous samples against distribution shift between tasks. iCaRL (Rebuffi et al., 2017) replays samples that are close to the mean feature of each class. Mnemonics (Liu et al., 2020) replays parameterized exemplars that mostly approximate previous tasks. *Dynamic architecture* methods try to isolate parameters or representations of each task. AANet (Liu et al., 2021) designs stable blocks and plastic blocks for previous tasks and the new task respectively and learns weights to adaptively aggravate representations.

Mode Connectivity is a phenomenon that different minima can be connected by low-loss paths in parameter space (Garipov et al., 2018). This is an innovative view that minima optimized by SGD or other optimizers are points on the same connected multi-dimensional manifold of low-loss (Draxler et al., 2018; Benton et al., 2021), which advances our understanding of neural network optimization. Generally, the existence of mode connectivity depends on two aspects. First, *connectivity condition*, (Garipov et al., 2018) showed that high loss exists along the linear connection between two minima which are trained with different random initialization. (Frankle et al., 2020) found that linear connectivity exists when two minima are trained from the same initialization, which should be stable to SGD noise. Therefore, the initialization of minima is an important factor for the existence of mode connectivity. Second, *connectivity finding* tries to find a lower-loss path or more general form of connection. The key factor of this aspect is to model connectivity properly, such as polygonal chain, Bezier curve (Garipov et al., 2018), elastic band (Draxler et al., 2018), and simplicial complexes (Benton et al., 2021). In addition, mode connectivity brings convenience to other research, such as loss landscape analysis (Garipov et al., 2018; Draxler et al., 2018; Fort & Jastrzebski, 2019; Czarnecki et al., 2019), weight pruning analysis (Frankle et al., 2020), and model ensembling (Fort & Jastrzebski, 2019; Fort et al., 2019; Benton et al., 2021; Wortsman et al., 2021).

Discussion. In this paper, we introduce mode connectivity into CIL and study from the following three aspects. First, considering the impact of initialization on connectivity, we conduct linear interpolation experiments with different ways of initialization to understand the basic situation of connectivity in CIL. Second, we propose a connectivity finding method OPC based on the Fourier series and gradient projection, which is novel and parameter-saving compared with existing connectivity finding methods (Garipov et al., 2018; Benton et al., 2021). Third, we further propose a performance-boosting method EOPC, which is orthogonal to existing CIL work (Rebuffi et al., 2017; Douillard et al., 2020; Liu et al., 2021) and can be easily plugged into them in a way of post-processing.

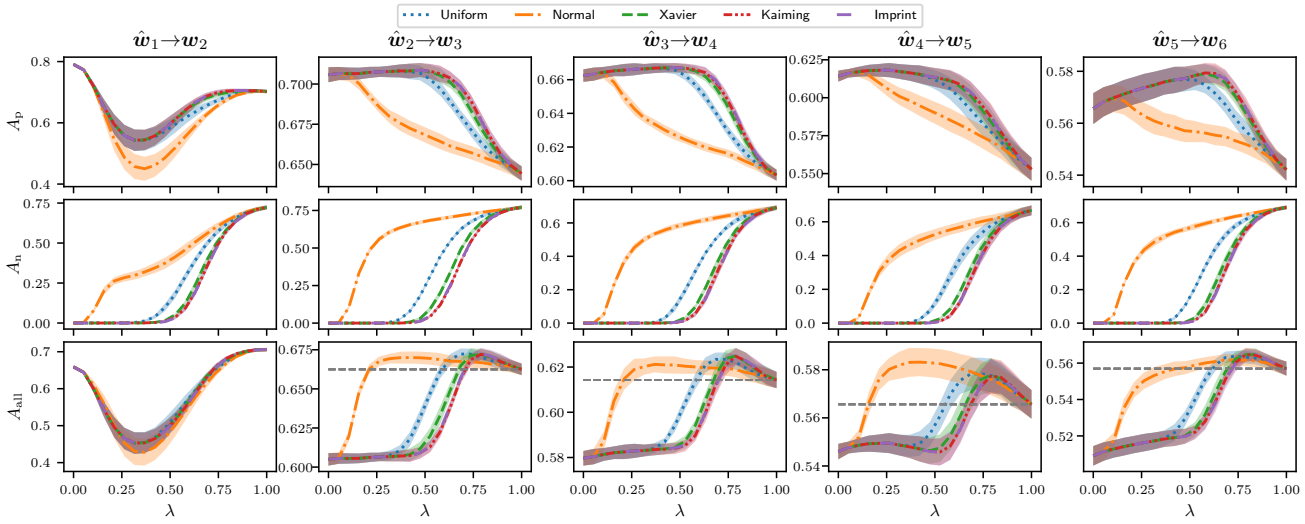

Figure 2. Testing accuracy curves along the linear connection between two adjacent continual minima of PODNet (Douillard et al., 2020) for 5 steps of increments (i.e., 6 tasks in total) on CIFAR-100. A_p, A_n, and A_all denote accuracy on previous tasks, on the new task, and on all learned tasks respectively. λ is the interpolation factor. Results are the mean and standard deviation of 10 experiments under different random seeds. Taking the testing accuracy as the measure of connectivity is because it is more sensitive to moving along the path than training loss.

3. Analyzing Linear Connectivity in CIL

We first give some notations and concepts for convenient subsequent description. Continual learning requires a model parameterized with $\boldsymbol{w} \in \mathbb{R}^n$ to learn a sequence of T tasks $\{\mathcal{D}_t\}_{t=1}^T$. For the scenario of class incremental learning (Van de Ven & Tolias, 2019), data of task t is a set of tuples and classes between different tasks are not overlapped, i.e., $\mathcal{D}_t = \{(\boldsymbol{x}_t, y_t)\}$ and $\{y_t\} \cap \{y_{k \neq t}\} = \emptyset$, where $\boldsymbol{x}$ is the input, y is the corresponding label. We denote the parameter vector $\boldsymbol{w}$ of a L layers model after learning task t as $\boldsymbol{w}_t = \{\boldsymbol{w}_{t,l}\}_{l=1}^L$. Most frontier CIL work adopts memory replay (Rebuffi et al., 2017; Hou et al., 2019; Simon et al., 2021; Liu et al., 2021; Kang et al., 2022), we denote the memory of all learned tasks as $\mathcal{M}_t$, which is incrementally constructed by selecting a small number of representative samples for each task.

3.1. Linear Interpolation Experiments

Given minima of two adjacent tasks $\boldsymbol{w}_{t-1}$ and $\boldsymbol{w}_t$. Under the scenario of CIL, before learning the new task t, a new parameter vector $\boldsymbol{z}_t$ parameterizes the new classifier should be added to $\boldsymbol{w}_{t-1}$ for learning new classes, and $\boldsymbol{z}_t$ is initialized by a distribution P. Therefore, the minimum is expanded as $\hat{\boldsymbol{w}}_{t-1} = \boldsymbol{w}_{t-1} \oplus \boldsymbol{z}_t$, where $\oplus$ is the concatenation operator. Generally, $\hat{\boldsymbol{w}}_{t-1}$ is taken as the initial parameters of task t. The expanded minimum of task $t-1$ and the minimum of task t are interpolated by $\boldsymbol{w} = (1-\lambda)\hat{\boldsymbol{w}}_{t-1} + \lambda \boldsymbol{w}_t, \lambda \in [0, 1]$. Here we only focus on the connectivity between minima after learning task $t-1$ and t and keep the original way of initialization on $\boldsymbol{z}_t$ invariant during the incremental learning.

Figure 2 shows testing accuracy curves along the linear path between two adjacent minima with five ways of initialization, including Uniform, Normal, Xavier-Normal (Glorot & Bengio, 2010), Kaiming-Normal (He et al., 2015), and Imprint (Hou et al., 2019) that commonly used in CIL. From this figure, we can see that:

- There is an interval on the linear path $\hat{\boldsymbol{w}}_1 \to \boldsymbol{w}_2$ incurs significant testing accuracy degradation of task 1, which indicates that a high-loss ridge possibly exists along this path in training loss landscape of task 1. Therefore, the overall accuracy on tasks 1 and 2 inevitably deteriorates.
- There is an interval on the linear path $\hat{\boldsymbol{w}}_{t-1} \to \boldsymbol{w}_t, t \geq 3$ achieves a higher accuracy on all learned tasks than endpoints (the part above the gray dashed line), which indicates that points sampled within this interval can replace $\boldsymbol{w}_t$ as a better minimum of task t.
- Xavier, Kaiming, and Imprint can achieve higher accuracy on all learned tasks along the linear path compared with Uniform and Normal (except $\hat{\boldsymbol{w}}_4 \to \boldsymbol{w}_5$), by further comparing A_p and A_n, the accuracy of this three initialization on previous tasks can maintain higher over a longer distance starting from $\hat{\boldsymbol{w}}_{t-1}$, which indicates that paths of this three initialization locate in a lower-loss region in training loss landscape of previous tasks.

Therefore, if we pull the path from the high-loss ridge to the low-loss region, we can get an interval on the path that has higher accuracy than endpoints. In the next section, we will propose a connectivity optimization method to dodge this high-loss ridge.

4. OPC: Optimizing Connectivity between Minima

To find a low-loss path between expanded minimum $\hat{\boldsymbol{w}}_{t-1}$ and minimum $\boldsymbol{w}_t$, let $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda) : [0,1] \to \mathbb{R}^n$ be the parameterized arbitrary path between them, such that

$$\boldsymbol{p}_{\boldsymbol{\theta}}(0) = \hat{\boldsymbol{w}}_{t-1} \quad \text{and} \quad \boldsymbol{p}_{\boldsymbol{\theta}}(1) = \boldsymbol{w}_t \tag{1}$$

and $\boldsymbol{\theta}$ is the parameters of path. We commonly use the expected loss $\hat{\ell}(\boldsymbol{\theta})$ along the path to characterize its quality (Garipov et al., 2018), i.e.,

$$\hat{\ell}(\boldsymbol{\theta}) = \int_0^1 \mathcal{L}(\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda))d\lambda = \mathbb{E}_{\lambda\sim U(0,1)}[\mathcal{L}(\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda))], \tag{2}$$

where $\mathcal{L}$ is the task loss, such as cross-entropy loss, NCA loss (Douillard et al., 2020), or embedding loss (Hou et al., 2019), $U(0,1)$ is the uniform distribution on the interval $[0,1]$. Under general settings (Garipov et al., 2018; Draxler et al., 2018), if minima are trained on the same data but from different initialization or the same initialization that is stable to SGD noise (Frankle et al., 2020), there exists mode connectivity between minima. Hence, we can randomly sample points λ between $[0,1]$ and minimize loss $\mathcal{L}(\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda))$ with respect to $\boldsymbol{\theta}$ to optimize the path, i.e.,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)), \lambda \sim U[0,1], \tag{3}$$

where γ is the learning rate of the path. However, for continual learning, the new minimum is trained starting from the previous minimum on the new task data, which will inevitably rise two problems: general connectivity conditions are not met between two continual minima, and catastrophic forgetting occurs in the new minimum. Therefore, we can not directly establish a low-loss path between two adjacent continual minima whether in terms of previous tasks or the new task. Multitask learning learns all task data simultaneously and its minimum is capable of all tasks. We hypothesize that there is a point on the optimal path that has properties similar to the multitask minimum and term it the switching point (SP). Then, we can redefine a low-loss path taking SP as a bridge, where the part between the previous minimum and SP is for previous tasks, and the part between SP and the new minimum is for the new task. Let λ^* be the point corresponding to SP in the interval $[0,1]$. We can reformulate Equation (2) for continual learning as follows,

$$\ell(\boldsymbol{\theta}) = \int_0^{\lambda^*} \mathcal{L}_{1:t-1}(\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda))d\lambda + \int_{\lambda^*}^1 \mathcal{L}_t(\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda))d\lambda, \tag{4}$$

where $\mathcal{L}_{1:t-1}$ is loss on previous tasks and $\mathcal{L}_t$ is loss on the new task. With this reasonable criterion, we can effectively evaluate the quality of the path $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)$ between continual minima.

4.1. Connectivity Modeling

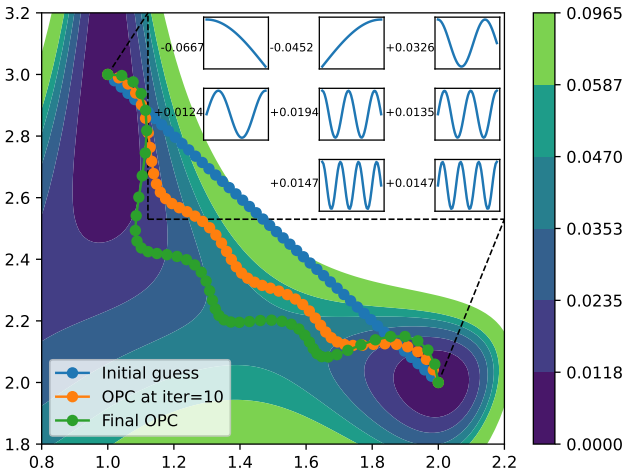

Figure 3. A toy example of optimizing connectivity between minima in the 2-dimensional plane. The initial guess of the low-loss path is set to the linear connection. We adopt the Fourier series to approximate arbitrary perturbation between the linear path and the ideal low-loss path. The coefficients of trigonometric functions are optimized by gradient projection to minimize loss along the path, i.e., Equation (2) or Equation (4).

Existing connectivity modeling depends on polygonal chain (Garipov et al., 2018), Bezier curve (Garipov et al., 2018), elastic band (Draxler et al., 2018), and simplicial complexes (Benton et al., 2021), different from these, we propose a novel modeling based on the Fourier series. There are infinite forms of paths and the basic requirement of a path is to pass through two adjacent continual minima. Inspired by solving the Brachistochrone Problem (Haws & Kiser, 1995), our main idea is to take linear connection as the basic form, and then add perturbation $\boldsymbol{\delta}_{\boldsymbol{\theta}}(\lambda)$ on this basis to approximate any form of the path, i.e.,

$$\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda) = (1-\lambda)\hat{\boldsymbol{w}}_{t-1} + \lambda\boldsymbol{w}_t + \boldsymbol{\delta}_{\boldsymbol{\theta}}(\lambda), \lambda \in [0,1]. \tag{5}$$

This requires $\boldsymbol{\delta}_{\boldsymbol{\theta}}(\lambda)$ should meet the following conditions,

$$\boldsymbol{\delta}_{\boldsymbol{\theta}}(0) = \mathbf{0} \quad \text{and} \quad \boldsymbol{\delta}_{\boldsymbol{\theta}}(1) = \mathbf{0}. \tag{6}$$

The property of zeroing at endpoints is similar to that perturbation takes the linear connection as the horizontal axis and fluctuates up and down around it between the interval $[0,1]$. The family of trigonometric functions also has this property. Therefore, it drives us to model the perturbation as a periodic curve, and use trigonometric functions paired

with series to approximate it. Fourier series approximates arbitrary curves by weighting trigonometric functions (basis functions) with multiple frequencies, i.e.,

$$\hat{\delta}_{\boldsymbol{\theta}}(\lambda) = \frac{\alpha_0}{2} + \sum_{i=1}^{N} \alpha_i \cos(\omega_i \lambda + \varphi_i), \tag{7}$$

where $\alpha_0/2$ is the DC component, α_i, ω_i, and φ_i are coefficients, frequencies, and phases of basis functions respectively, let $\boldsymbol{\theta} = [\alpha_1, \alpha_2, ..., \alpha_N]$. In our case, we set α_0 and φ_i to 0. The fundamental angular frequency of $\hat{\delta}_{\boldsymbol{\theta}}(\lambda)$ is ω_1, we only set $[0, 1]$ as the 1/4 fundamental period to reduce the fluctuation of the path for more convenient optimization, thus $\omega_1 = \frac{2\pi}{4} = \frac{\pi}{2}$. In addition, the behaviors of standard cosine and sine functions in the first 1/4 period (i.e., $[0, \pi/2]$) are opposite, cosine starts from 1 to 0 and sine starts from 0 to 1. Therefore, we can multiply $\cos$ with $\hat{\boldsymbol{w}}_{t-1}$ and multiply $\sin$ with $\boldsymbol{w}_t$ to construct a curve in 2-dimensional plane, i.e,

$$\hat{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda) = \cos(\frac{\pi}{2}\lambda)\hat{\boldsymbol{w}}_{t-1} + \sin(\frac{\pi}{2}\lambda)\boldsymbol{w}_t, \tag{8}$$

where 1/4 period is mapped from $[0, \pi/2]$ to $[0, 1]$ by adjusting the angular frequency to $\pi/2$. This curve can meet Condition (1), i.e., pass through $\hat{\boldsymbol{w}}_{t-1}$ at point 0 and $\boldsymbol{w}_t$ at point 1. To form a more complex curve, we can add trigonometric functions with higher frequencies into Equation (8), the angular frequency of i-th term trigonometric function must be $\omega_i = \pi/2 + 2\pi(i-1) = (4i-3)\pi/2$ so that the i-th term is the same as the 1-th term at points 0 and 1, then

$$\begin{aligned}\hat{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda) = &\sum_{i=1}^{N} \alpha_i \cos(\frac{(4i-3)\pi}{2}\lambda)\hat{\boldsymbol{w}}_{t-1} + \\ &\sum_{i=1}^{N} \beta_i \sin(\frac{(4i-3)\pi}{2}\lambda)\boldsymbol{w}_t,\end{aligned} \tag{9}$$

where $\boldsymbol{\theta} = [[\alpha_1, ..., \alpha_N]^{\mathrm{T}}, [\beta_1, ..., \beta_N]^{\mathrm{T}}]^{\mathrm{T}}$, in this case, α_i and β_i should meet $\sum_{i=1}^{N} \alpha_i = 1$ and $\sum_{i=1}^{N} \beta_i = 1$ to make $\hat{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda)$ still meet Condition (1). Under fixed group of basis functions, substitute Equation (9) into Equation (4), we can optimize parameters $\boldsymbol{\theta}$ according to $\ell(\boldsymbol{\theta})$ to obtain the low-loss path. However, this formulation does not have good initialization, in practice, it will make $\ell(\boldsymbol{\theta})$ unstable, and we can only constrain α_i and β_i within a small boundary for optimization (Rosen, 1960). Similar to AutoNEB sets the initial guess as the linear connection(Draxler et al., 2018), we formulate the path as Equation (5). Here we let $\boldsymbol{\delta}_{\boldsymbol{\theta}}(\lambda) = \hat{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda)$, and accordingly, the constraints of parameters should be changed to $\sum_{i=1}^{N} \alpha_i = 0$ and $\sum_{i=1}^{N} \beta_i = 0$ to meet Condition (6). Connections in a 2-dimensional plane may not be optimal, in addition, (Benton et al., 2021) shows that the low-loss valley is a multi-dimensional manifold. Therefore, we connect minima in a layer-wise manner to construct a space curve $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)$ in subspace $\mathbb{R}^{L+1}$, and write it in the form of matrix operations,

$$\begin{aligned}\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda) =&(\boldsymbol{A}\boldsymbol{C} + (1-\lambda)\mathbf{1}_L) \cdot \hat{\boldsymbol{w}}_{t-1} \\ &+ (\boldsymbol{B}\boldsymbol{S} + \lambda\mathbf{1}_L) \cdot \boldsymbol{w}_t,\end{aligned} \tag{10}$$

where $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{L\times N}$ and $\boldsymbol{C}, \boldsymbol{S} \in \mathbb{R}^{N}$, specifically,

$$\begin{aligned}\boldsymbol{A} = \begin{bmatrix} \alpha_{1,1} & \dots & \alpha_{1,N} \\ & \vdots & \\ \alpha_{L,1} & \dots & \alpha_{L,N} \end{bmatrix} \boldsymbol{C} = \begin{bmatrix} \cos(\frac{\pi}{2}\lambda) \\ \vdots \\ \cos(\frac{(4N-3)\pi}{2}\lambda) \end{bmatrix} \\ \boldsymbol{B} = \begin{bmatrix} \beta_{1,1} & \dots & \beta_{1,N} \\ & \vdots & \\ \beta_{L,1} & \dots & \beta_{L,N} \end{bmatrix} \boldsymbol{S} = \begin{bmatrix} \sin(\frac{\pi}{2}\lambda) \\ \vdots \\ \sin(\frac{(4N-3)\pi}{2}\lambda) \end{bmatrix}\end{aligned} \tag{11}$$

$\mathbf{1}_L \in \mathbb{R}^L$ is the notation of all-ones vector, $\cdot$ is the notation of element-wise multiplication between vectors or matrices, we split $\boldsymbol{w}$ into L blocks for simplicity, i.e., $\boldsymbol{w} = [\boldsymbol{w}_1^{\mathrm{T}}, \dots, \boldsymbol{w}_L^{\mathrm{T}}]^{\mathrm{T}}$. Correspondingly, $\boldsymbol{\theta} = [\boldsymbol{A}^{\mathrm{T}}, \boldsymbol{B}^{\mathrm{T}}]^{\mathrm{T}}$, and the constraints of parameters should be,

$$\boldsymbol{\theta}\mathbf{1}_N = \mathbf{0}_{2L}, \tag{12}$$

where $\mathbf{0}_{2L} \in \mathbb{R}^{2L}$ is a all-zeros vector. This is a constrained optimization, next, we will adopt gradient projection to cope with this problem.

4.2. Connectivity Optimizing

Our objective function is shown in Equation (4), which requires all previous tasks data to compute loss $\mathcal{L}_{1:t-1}$. Considering the setups of continual learning, we can only use the memory $\mathcal{M}_t$ to evaluate the path $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)$ after learning task t. However, the size of memory is commonly set very small, i.e., $|\mathcal{M}_t| \ll |\mathcal{D}_{1:t}|$, it may not appropriately describe the original distribution of previous tasks and incurs overfitting on it (Rebuffi et al., 2017; Liu et al., 2020). A flat minimum tends to have better generalization performance (Hochreiter & Schmidhuber, 1997; Keskar et al., 2016). This drives us to pull the path to a flatter region. We try to construct a bent cylinder along the path and randomly evaluate points on its surface. It is essentially equivalent to adding fixed amplitude noise orthogonal to the tangent direction of the path. According to Equation (10), the tangent is

$$\boldsymbol{p}'_{\boldsymbol{\theta}}(\lambda) = (\boldsymbol{A}'\boldsymbol{S} - \mathbf{1}_L) \cdot \hat{\boldsymbol{w}}_{t-1} + (\boldsymbol{B}'\boldsymbol{C} + \mathbf{1}_L) \cdot \boldsymbol{w}_t, \tag{13}$$

where $\boldsymbol{A}', \boldsymbol{B}' \in \mathbb{R}^{L\times N}$, specifically,

$$\begin{aligned}\boldsymbol{A}' = \begin{bmatrix} -\frac{\pi}{2}\alpha_{1,1} & \dots & -\frac{(4N-3)\pi}{2}\alpha_{1,N} \\ & \vdots & \\ -\frac{\pi}{2}\alpha_{L,1} & \dots & -\frac{(4N-3)\pi}{2}\alpha_{L,N} \end{bmatrix} \\ \boldsymbol{B}' = \begin{bmatrix} \frac{\pi}{2}\beta_{1,1} & \dots & \frac{(4N-3)\pi}{2}\beta_{1,N} \\ & \vdots & \\ \frac{\pi}{2}\beta_{L,1} & \dots & \frac{(4N-3)\pi}{2}\beta_{L,N} \end{bmatrix}\end{aligned} \tag{14}$$

Then, randomly sample noise from the normal distribution and orthogonalize it with the tangent direction, i.e.,

$$\boldsymbol{\epsilon} = \hat{\boldsymbol{\epsilon}} - \frac{\boldsymbol{p}'_{\boldsymbol{\theta}}(\lambda)^{\mathrm{T}}\hat{\boldsymbol{\epsilon}}\boldsymbol{p}'_{\boldsymbol{\theta}}(\lambda)}{\|\boldsymbol{p}'_{\boldsymbol{\theta}}(\lambda)\|^2}, \quad \hat{\boldsymbol{\epsilon}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{15}$$

where $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ is the identity matrix. Next, add normalized noise scaled by radius r to the path to obtain the point on the surface of the cylinder,

$$\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda) = \boldsymbol{p}_{\boldsymbol{\theta}}(\lambda) + r\frac{\boldsymbol{\epsilon}}{\|\boldsymbol{\epsilon}\|}. \tag{16}$$

Replace $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)$ with $\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda)$ in Equation (3), we can get a flatter path for traditional connectivity modeling. However, in our case, directly using gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda))$ will make the updated path violate constraints, i.e., Equation (12). We adopt gradient projection to cope with this problem. For Equation (12), which is an equality constraint, it is feasible to update parameters along the direction orthogonal to the normal of equation, i.e.,

$$\begin{aligned}\boldsymbol{\Delta}(\lambda) &= \nabla_{\boldsymbol{\theta}}\mathcal{L}(\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda))(\boldsymbol{I}_{N\times N} - \mathbf{1}_N(\mathbf{1}_N^{\mathrm{T}}\mathbf{1}_N)^{-1}\mathbf{1}_N^{\mathrm{T}}) \\ &= \nabla_{\boldsymbol{\theta}}\mathcal{L}(\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda)) - \frac{1}{N}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda))\mathbf{1}_N\mathbf{1}_N^{\mathrm{T}}.\end{aligned} \tag{17}$$

This operation is essentially equivalent to subtracting the mean of gradient from original gradient in each layer. Replace $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda))$ with $\boldsymbol{\Delta}(\lambda)$ in Equation (3), we can get the iterative rule for parameters of the path,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma\boldsymbol{\Delta}(\lambda), \quad \lambda \sim U[0,1]. \tag{18}$$

4.3. EOPC: Ensembling with OPC

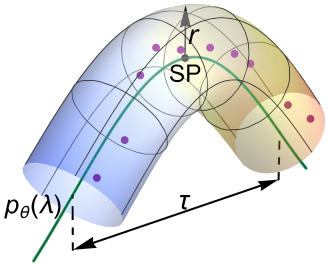

Figure 4. The green curve is the optimized low-loss path $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)$, the surface with gradient color around the path forms the local bent cylinder with radius r, τ is the width of the interval, and "SP" is the switching point. EOPC randomly sample (purple) points within this cylinder and average them for ensembling.

The optimized path provides infinite low-loss solutions on both sides of the switching point (SP), i.e., $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda^*)$. To further improve performance on learned tasks, we propose EOPC to ensemble points within a local bent cylinder around SP. The cylinder is constructed according to the tangent of the path, let S be the set of points within this cylinder and can be formulated as follows,

$$\begin{aligned}S = \{\boldsymbol{w} | (\boldsymbol{w} - \boldsymbol{p}_{\boldsymbol{\theta}}(\lambda))^{\mathrm{T}}\boldsymbol{p}'_{\boldsymbol{\theta}}(\lambda) = 0, \|\boldsymbol{w} - \boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)\|_2 \le r; \\ \lambda \in [\lambda^* - \tau/2, \lambda^* + \tau/2]\},\end{aligned} \tag{19}$$

where τ is the width of the interval. There are generally two roadmaps of ensembling, one is ensembling in output space, and the other is ensembling in parameter space. Considering the requirement of parameter efficiency in CIL, we adopt the latter scheme by averaging points within S, i.e.,

$$\overline{\boldsymbol{w}} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{w}_i, \quad \boldsymbol{w}_i \sim S, \tag{20}$$

where M is the number of total sampling points. The operation of $\boldsymbol{w}_i \sim S$ can be performed by replacing r in Equation (16) with a random factor $k \sim U[0, r]$. We take $\overline{\boldsymbol{w}}$ as the minimum of the current task and the initial parameters of model in the next task.

5. Experiments

In this section, we will adapt EOPC to several existing representative CIL methods in a post-processing manner on several benchmarks for comparisons. Then, extensive analytical experiments are conducted to validate the effectiveness and scalability of EOPC. Next, we introduce the basic settings of experiments.

Benchmarks. Three different datasets are used in our experiments: **1) CIFAR-100** contains 100 classes, each class has 500 training samples and 100 testing samples with image size 32×32 (Krizhevsky et al., 2009). **2) ImageNet-1K** contains 1000 classes, each class has about 1300 training samples and 50 validation samples (Deng et al., 2009). **3) ImageNet-100** consists of 100 classes and is randomly extracted from ImageNet-1K with a fixed random seed 1993. We split these datasets into a sequence of tasks, the first task contains half of the classes, e.g., 50 classes for CIFAR-100, then the rest of the classes are equally assigned to 5, 10, and 25 steps for incremental learning.

Baselines and Evaluation Metrics. Two representative CIL methods are chosen as our adaptation baselines: PODNet (Douillard et al., 2020) and AANet (Liu et al., 2021). For comprehensive comparisons, iCaRL (Rebuffi et al., 2017), BiC (Wu et al., 2019), LUCIR (Hou et al., 2019), Mnemonics (Liu et al., 2020), GeoDL (Simon et al., 2021), and AFC (Kang et al., 2022) are chosen as our comparison baselines. We use two metrics for evaluation, one is the average incremental accuracy (Rebuffi et al., 2017), i.e., $\mathcal{A} = \frac{1}{T}\sum_{t=1}^{T}A_t$, where A_t is the testing accuracy on all learned tasks after learning task t. The other is the average forgetting of previous tasks, $\mathcal{F} = \frac{1}{T-1}\sum_{i=1}^{T-1}\max_{t\in\{1,\dots,T-1\}}(a_{t,i} - a_{T,i})$, where $a_{t,i}$ is the

Method	CIFAR-100			ImageNet-100			ImageNet-1K		
$\mathcal{A}$ (%) ↑	5	10	25	5	10	25	5	10	25
iCaRL (Rebuffi et al., 2017)	57.83	52.63	49.02	64.75	58.80	52.46	51.60	47.42	41.03
BiC† (Wu et al., 2019)	59.36	54.20	50.00	70.07	64.96	57.73	62.65	58.72	53.47
LUCIR (Hou et al., 2019)	63.62	60.95	57.79	71.93	69.43	63.51	66.13	61.63	54.05
Mnemonics† (Liu et al., 2020)	63.34	62.28	60.96	72.58	71.37	69.74	64.63	63.01	61.00
GeoDL† (Simon et al., 2021)	65.14	65.03	**63.12**	73.87	73.55	71.72	65.23	64.46	62.20
AFC (Kang et al., 2022)	65.87	64.45	62.05	77.27	**75.47**	**72.41**	69.07	66.85	**63.40**
PODNet (Douillard et al., 2020)	65.47	63.13	59.85	76.32	73.54	63.05	68.33	65.35	58.62
w/ EOPC	66.68	64.94	62.36	77.12	74.53	68.18	**69.72**	**67.57**	62.35
AANet (Liu et al., 2021)	66.53	64.63	61.05	77.98	74.70	68.65	68.87	65.65	60.07
w/ EOPC	**67.55**	**65.54**	61.82	**78.95**	74.99	70.10	69.47	67.35	62.20
$\mathcal{F}$ (%) ↓	5	10	25	5	10	25	5	10	25
iCaRL (Rebuffi et al., 2017)	25.16	26.57	29.83	24.22	29.63	32.58	16.66	15.94	18.91
LUCIR (Hou et al., 2019)	19.58	19.79	20.31	20.56	25.97	28.55	13.68	26.99	37.73
AFC (Kang et al., 2022)	12.86	14.16	19.37	11.82	12.81	21.47	12.76	14.72	21.69
PODNet (Douillard et al., 2020)	19.26	25.01	28.55	13.72	18.41	29.11	13.88	17.97	28.81
w/ EOPC	7.68	8.64	12.02	6.16	4.15	8.3	11.68	15.75	25.58
AANet (Liu et al., 2021)	25.74	30.31	32.15	18.08	22.17	28.66	16.50	19.88	28.14
w/ EOPC	20.96	23.22	25.17	8.3	9.66	12.06	15.14	18.45	26.70

Table 1. The adaptation results of EOPC and comparison results with existing incremental learning methods on CIFAR-100, ImageNet-100, and ImageNet-1K. The upper part is the average incremental accuracy results and the lower part is the average forgetting results. † represents results referenced from (Simon et al., 2021).

testing accuracy of the model on task i after learning task t. A better CIL method should have a higher $\mathcal{A}$ and a lower $\mathcal{F}$.

Model Architectures. We use two types of model architectures: ResNet-32 (He et al., 2016) for CIFAR-100 and ResNet-18 for both ImageNet-100 and ImageNet-1K. Although these two types of architectures are widely adopted in contemporary CIL methods, there are a few differences between them, e.g., AANet adopts a dual-branch ResNet architecture that is different from the single-branch ResNet in PODNet. This difference can validate the scalability of EOPC in the width of the model to a certain extent. For a more comprehensive validation, we take iCaRL as the adaptation baseline and use different architectures, including CNN, ResNet, and DenseNet (Huang et al., 2017), with varying widths and depths in our analytical experiments.

5.1. Adaptation and Comparison Results

Results on CIFAR-100. Table 1 summarizes the adaptation results of EOPC and comparison results with representative CIL methods on CIFAR-100. EOPC improves PODNet by 1.21%, 1.81%, and 2.51%, and improves AANet by 1.02%, 0.91%, and 0.77% for 5, 10, and 25 steps respectively. In addition, EOPC can significantly reduce forgetting, such as reducing forgetting of PODNet by 11.58%, 16.37%, and 16.53% for 5, 10, and 25 steps respectively.

Results on ImageNet. Table 1 also provides results on ImageNet-100 and ImageNet-1K. It can be seen that EOPC improves PODNet by 0.8%, 0.99%, and 5.13% for 5, 10, and 25 steps respectively, and improves AANet by 0.97%, 0.29%, and 1.45% for 5, 10, and 25 steps respectively on ImageNet-100. EOPC improves PODNet by 1.39%, 2.22%, and 3.73%, and improves AANet by 0.6%, 1.7%, and 2.1% for 5, 10, and 25 steps respectively on ImageNet-1K. Similar to results on CIFAR-100, EOPC can also reduce forgetting on ImageNet.

5.2. Analytical experiments

Ablation of OPC. In this part, we try to study the effectiveness of each component in EOPC. The first thing to figure out is whether the path found by OPC is more beneficial than the linear connection for CIL, we donate these two schemes as "OPC" and "Line" respectively. The switching point (SP) is directly taken as the initial parameters of the next task (i.e., without flattening and ensembling), where SP is $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda^*)$ for the "OPC" scheme and $(1-\lambda^*)\hat{\boldsymbol{w}}_{t-1}+\lambda^*\boldsymbol{w}_t$ for the "Line" scheme. Table 2 shows the results of these two schemes adapting to PODNet for 10 and 25 steps. As shown in Figure 1 and 2, the high ridge loss exists along the linear connection. It can be seen that the accuracy of PODNet is significantly decreased by SP in the linear connection. In contrast, OPC optimizes the path to dodge the high-loss ridge of previous tasks. Therefore, SP in OPC optimized path can achieve higher accuracy than endpoints and improves overall performance in CIL, especially for a long sequence of tasks. These results confirm that the "OPC"

scheme is an effective post-processing procedure for CIL.

Method	CIFAR-100		ImageNet-100	
$\mathcal{A}$ (%) ↑	10	25	10	25
PODNet	63.13	59.85	73.54	63.05
w/ Line	61.03	56.59	45.29	39.80
w/ OPC	64.04	62.21	74.11	67.88

Table 2. Adaptation results of the "OPC" scheme and the "Line" scheme to PODNet on CIFAR-100 and ImageNet-100.

Ablation of Flattening and Ensembling. Next, we try to validate the effectiveness of the flattening (in Section 4.2) and ensembling techniques. The key hyperparameter of these two techniques is the radius of the cylinder. We combine these two techniques to form three schemes. The left of Figure 5 shows the accuracy curves of these schemes when added to the "OPC" scheme under different radii. It can be seen that the individual "Ensembling" scheme can not

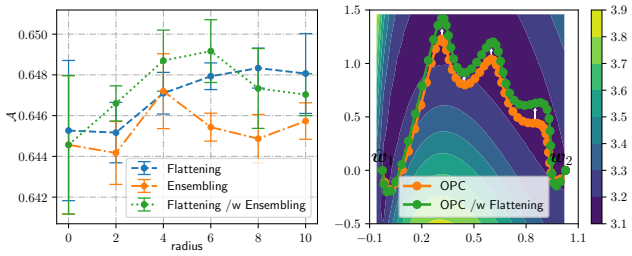

Figure 5. **Left:** The accuracy curves of three schemes when added to the "OPC" scheme on CIFAR-100 for 10 steps under different radii. **Right:** The path found by OPC and the path found by OPC with flattening in training loss landscape of task 1 on CIFAR-100 for 5 steps.

improve accuracy continuously with the increase of radius. This is because the roadmap of ensembling in parameter space brings weak benefits (Frankle et al., 2020). In contrast, the individual "Flattening" scheme can consistently improve accuracy with the increase of radius. The right of Figure 5 shows the path found by OPC and the path found by OPC with flattening, where the path found with flattening is farther away from the high loss region. Therefore, these results confirm that pulling the path to a flatter region will have better generalization performance. The third scheme is the "Ensembling /w Flattening" scheme which can achieve higher accuracy compared with its two components. This result indicates that it is necessary to consider the geometric structure of the ensembling area.

Validation of Scalability on Different Architectures. We adopt similar configurations of architectures to (Draxler et al., 2018), including CNN, ResNet, and DenseNet. For CNN, the size of the filter is set to 3×3 and "CNN-W×D" means CNN has D convolution layers (Depth) with W channels (Width). We consider 12×8, 24×8, 36×8, 48×8, and 96×8 to validate in different widths, and consider 48×6, 48×8, and 48×10 to validate in different depths. In addition, different depths of ResNet (-8, -20, -32, -44, and -56) and DenseNet (-40, -100) are also considered. Figure 6 shows the results of EOPC adapting to iCaRL with different architectures. It can be seen that EOPC can significantly improve the performance in a wide range of architectures, except the narrow CNN-12×8 and the shallow ResNet-8. We think that this is because these two architectures are too weak for CIL on CIFAR-100, the poor results of iCaRL when using CNN-12×8 and ResNet-8 indicate that continual minima of these architectures tend to stay in the high-loss region after each task. Therefore, it is difficult to find a low-loss path between their continual minima, which deteriorates the overall performance in CIL. We also provide adaptation results to the transformer architecture (Douillard et al., 2022) in Section D.

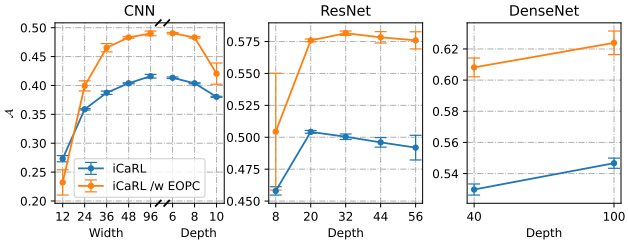

Figure 6. The results of EOPC adapting to iCaRL with different model architectures on CIFAR-100 for 5 steps.

6. Conclusion

In this paper, we studied the connectivity between two adjacent continual minima in CIL and found that the high-loss ridge exists along the linear connection. If we pull the path from the high-loss ridge to the low-loss region, we can get an interval on the path that has higher accuracy than endpoints. Therefore, we proposed optimizing connectivity (OPC) to find the low-loss path between minima. OPC models the connectivity by the Fourier series in a layer-wise manner to construct a space curve between minima in a multi-dimensional subspace. We further proposed the flattening scheme to pull the path to a flatter region and use the gradient projection to make the updated path still pass through two endpoints. With the path in a flat region, we proposed EOPC to ensemble points within a local bent cylinder for further improving the performance. Extensive experiments show that the adaptation of EOPC to representative CIL methods can significantly improve performance.

We think that the ideal minimum of multitasks is located

somewhere in the region of low-loss connecting continual minima. In the future, we will continue to leverage the theory of mode connectivity for a more accurate finding of the multitask minimum.

Acknowledgements

This work was supported in part by National Key R&D Program of China (2021ZD0112001) and National Natural Science Foundation of China (No. 61831005, No. 62171111).

We thank all reviewers for taking the time to review our paper and give valuable suggestions.

References

Ashfahani, A. and Pratama, M. Autonomous deep learning: Continual learning approach for dynamic environments. In *Proceedings of the 2019 SIAM international conference on data mining*, pp. 666–674. SIAM, 2019.

Benton, G., Maddox, W., Lotfi, S., and Wilson, A. G. G. Loss surface simplexes for mode connecting volumes and fast ensembling. In *International Conference on Machine Learning*, pp. 769–779. PMLR, 2021.

Caccia, M., Rodriguez, P., Ostapenko, O., Normandin, F., Lin, M., Caccia, L., Laradji, I., Rish, I., Lacoste, A., Vazquez, D., et al. Online fast adaptation and knowledge accumulation: a new approach to continual learning. *arXiv preprint arXiv:2003.05856*, 2020.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pp. 192–204. PMLR, 2015.

Czarnecki, W. M., Osindero, S., Pascanu, R., and Jaderberg, M. A deep neural network's loss surface contains every low-dimensional pattern. *arXiv preprint arXiv:1912.07559*, 2019.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pp. 86–102. Springer, 2020.

Douillard, A., Ramé, A., Couairon, G., and Cord, M. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9285–9295, 2022.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Fort, S. and Jastrzebski, S. Large scale structure of neural network loss landscapes. *Advances in Neural Information Processing Systems*, 32, 2019.

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8803–8812, 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Haws, L. and Kiser, T. Exploring the brachistochrone problem. *The American Mathematical Monthly*, 102(4):328–336, 1995.

Hayes, T. L. and Kanan, C. Selective replay enhances learning in online continual analogical reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3502–3512, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pp. 770–778, 2016.

Henkelman, G. and Jónsson, H. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *The Journal of chemical physics*, 113(22):9978–9985, 2000.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Jin, X., Lin, B. Y., Rostami, M., and Ren, X. Learn continually, generalize rapidly: Lifelong knowledge accumulation for few-shot learning. *arXiv preprint arXiv:2104.08808*, 2021.

Kang, M., Park, J., and Han, B. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16071–16080, 2022.

Kemker, R. and Kanan, C. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

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. *arXiv preprint arXiv:1609.04836*, 2016.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *https://www.cs.toronto.edu/kriz/cifar.html*, 2009.

Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., and Díaz-Rodríguez, N. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020.

Liu, Y., Su, Y., Liu, A.-A., Schiele, B., and Sun, Q. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 12245–12254, 2020.

Liu, Y., Schiele, B., and Sun, Q. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2544–2553, 2021.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Mirzadeh, S. I., Farajtabar, M., Gorur, D., Pascanu, R., and Ghasemzadeh, H. Linear mode connectivity in multitask and continual learning. *arXiv preprint arXiv:2010.04495*, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Robins, A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

Rosen, J. B. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the society for industrial and applied mathematics*, 8(1):181–217, 1960.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pp. 4548–4557. PMLR, 2018.

Simon, C., Koniusz, P., and Harandi, M. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1591–1600, 2021.

Van de Ven, G. M. and Tolias, A. S. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

van de Ven, G. M., Siegelmann, H. T., and Tolias, A. S. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020.

Wortsman, M., Horton, M. C., Guestrin, C., Farhadi, A., and Rastegari, M. Learning neural network subspaces. In *International Conference on Machine Learning*, pp. 11217–11227. PMLR, 2021.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017.

A. Algorithm

Algorithm 1 Adapting EOPC to CIL methods

Input: a sequence of tasks data $\{\mathcal{D}_t\}_{t=1}^{T}$, parameter vector of the model $\boldsymbol{w}$, existing CIL method $\mathscr{A}$, memory of previous tasks $\mathcal{M}$, learning rate of the path γ, and predetermined λ^*.
Random initialize $\boldsymbol{w}_0$.
Train $\boldsymbol{w}_0$ on $\mathcal{D}_1$ and obtain $\boldsymbol{w}_1$.
Store representative samples of $\mathcal{D}_1$ into $\mathcal{M}_1$.
for $t = 2, \ldots, T$ **do**
  Incremental learning with existing method, $\boldsymbol{w}_t = \mathscr{A}(\mathcal{D}_t, \mathcal{M}_{t-1}, \boldsymbol{w}_{t-1})$.
  Store representative samples of $\mathcal{D}_t$ and update the memory as $\mathcal{M}_t$.
  Expand parameter vector $\boldsymbol{w}_{t-1}$ by Kaiming initialization (He et al., 2015) and obtain $\hat{\boldsymbol{w}}_{t-1}$.
  Connect $\hat{\boldsymbol{w}}_{t-1}$ and $\boldsymbol{w}_t$ using $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)$ according to Equation (10).
  Initialize the path $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda)$ as the linear connection, i.e., $\boldsymbol{\theta} = \mathbf{0}$.
  repeat
    for $(\boldsymbol{x}, y) \sim \mathcal{M}_t$ **do**
      Sample $\lambda \in [0, 1]$ uniformly.
      Obtain points on the cylinder $\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda)$ according to Equation (16).
      Compute loss $\mathcal{L}(\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda))$ according to Equation (4) with predetermined λ^* and the batch sample $(\boldsymbol{x}, y)$.
      Project gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\widetilde{\boldsymbol{p}}_{\boldsymbol{\theta}}(\lambda))$ according to Equation (17) and obtain the feasible update $\boldsymbol{\Delta}$.
      Update parameters of the path, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma\boldsymbol{\Delta}$.
    end for
  until required epochs
  Ensembling within the local bent cylinder according to Equation (20) and obtain $\overline{\boldsymbol{w}}$.
  Assign the value of $\overline{\boldsymbol{w}}$ to $\boldsymbol{w}_t$.
end for

B. Batch Normalization

The batch normalization (BN) layer (Ioffe & Szegedy, 2015) is widely used in modern neural networks for accelerating training. It maps the input $\boldsymbol{x}$ to a unified distribution,

$$\hat{\boldsymbol{x}} = \boldsymbol{\gamma}\frac{\boldsymbol{x} - \boldsymbol{\mu}(\boldsymbol{x})}{\sqrt{\boldsymbol{\sigma}(\boldsymbol{x}) + \epsilon}} + \boldsymbol{\beta}, \tag{21}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are learnable parameters, $\boldsymbol{\mu}(\boldsymbol{x})$ and $\boldsymbol{\sigma}(\boldsymbol{x})$ are the mean and the standard deviation of the input, and $\epsilon > 0$ is for numerical stability. At the training stage, $\boldsymbol{\mu}(\boldsymbol{x})$ and $\boldsymbol{\sigma}(\boldsymbol{x})$ are the statistics of the current batch samples, we keep two additional variables $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ for storing the global statistics updated by the momentum of $\boldsymbol{\mu}(\boldsymbol{x})$ and $\boldsymbol{\sigma}(\boldsymbol{x})$. At the testing stage, the input is normalized by the global statistics $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, and mapped by the learned $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$.

Connectivity finding usually involves interpolation of parameters between minima, which will cause the mismatch between learnable parameters ($\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$) and the statistics ($\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$) in the BN layer. Existing work of mode connectivity deals with this problem by updating the statistics as usual at the training stage and updating the statistics with an additional forward pass on training data before being applied to the testing data (Draxler et al., 2018; Garipov et al., 2018). However, this operation is too time-consuming for class incremental learning (CIL), especially on a large dataset, such as ImageNet-1K. Furthermore, if the additional updating of statistics is on the tiny memory of previous tasks, it will bias the statistics. We deal with this problem in OPC for CIL by directly interpolating the global statistics of the previous model and the global statistics of the new model, i.e,

$$\begin{aligned} \boldsymbol{\mu} &= (1-\lambda)\boldsymbol{\mu}_{t-1} + \lambda\boldsymbol{\mu}_t \\ \boldsymbol{\sigma} &= (1-\lambda)\boldsymbol{\sigma}_{t-1} + \lambda\boldsymbol{\sigma}_t \end{aligned} \tag{22}$$

We apply this interpolation of statistics at both the training stage and the testing stage.

C. Implementation Details

Hyperparameters of all baselines are the same as the original implementations in their work. We use PyTorch (Paszke et al., 2019) to reimplement iCaRL (Rebuffi et al., 2017), LUCIR (Hou et al., 2019), PODNet (Douillard et al., 2020), AANet (Liu et al., 2021), and AFC (Kang et al., 2022) in the same environment for fair comparisons. For the hyperparameters of EOPC, we choose the SGD optimizer with an initial learning rate of 0.1, which is decayed by a factor of 0.1 at 10 and 15 epochs. The path between continual minima is optimized for 20 epochs with a batch size of 128. The maximum order of the Fourier series (i.e., N in Equation (10)) is set to 4, and the radius of the cylinder is chosen from $\{2, 4, 6\}$. We select an appropriate λ^* from $\{0.75, 0.85, 0.9\}$ for OPC and uniformly sample 10 points in the interval $[0.1, 0.95]$ and take their average loss as the loss of each iteration. Kaiming initialization is used to initialize the new parameter vector $\boldsymbol{z}_t$ in OPC. The number of total sampling points in ensembling is set to 10 and the interval τ is set to 0.1. The 10 random seeds used for results of Figure 2 are from 1991 to 2000. Consistent with existing CIL methods, the random seed used for results in Tables 1 and 2 is 1993. The random seeds used for all repeated experiments are from $\{1993, 1994, 1995\}$. In addition, we want to note that the cross-entropy (CE) loss is adopted for drawing the loss landscapes (except Figure 3) and for OPC because a higher performance can be achieved compared with the NCA loss (Douillard et al., 2020) (PODNet is taken as the testbed), this will lead to the loss $\mathcal{L}_{\mathrm{p}}$ and $\mathcal{L}_{\mathrm{n}}$ (colorbar range) larger than the usual value, but the OPC is functioning properly with the CE loss.

D. Repeated Experiments

To further demonstrate the robust effectiveness of EOPC in repeated experiments, we adapt it to existing CIL methods for 3 runs with different random seeds and report their mean performance and standard deviation in Table 3. DyTox (Douillard et al., 2022) is additionally chosen as the adaptation baseline, as it has a strong performance and is based on the recently prevalent transformer architecture (Dosovitskiy et al., 2020). It can be seen that EOPC can still consistently improve adaptation baselines in repeated experiments.

Method	CIFAR-100			ImageNet-100		
$\mathcal{A}$ (%) ↑	5	10	25	5	10	25
AFC (Kang et al., 2022)	65.94(±0.07)	64.29(±0.31)	62.33(±0.34)	77.25(±0.05)	75.45(±0.01)	72.65(±0.23)
PODNet (Douillard et al., 2020)	65.07(±0.28)	62.93(±0.14)	59.45(±0.28)	76.46(±0.14)	73.52(±0.06)	64.86(±1.31)
w/ EOPC	66.58(±0.01)	64.92(±0.20)	62.13(±0.09)	76.88(±0.18)	74.39(±0.10)	68.57(±0.53)
AANet (Liu et al., 2021)	65.97(±0.40)	64.08(±0.44)	60.44(±0.45)	77.97(±0.05)	74.98(±0.21)	68.51(±0.10)
w/ EOPC	67.04(±0.36)	65.17(±0.26)	61.69(±0.16)	78.77(±0.14)	74.82(±0.19)	70.15(±0.14)
DyTox (Douillard et al., 2022)	69.48(±0.12)	65.90(±0.14)	58.82(±0.18)	74.45(±0.10)	70.67(±0.09)	64.61(±0.22)
w/ EOPC	70.04(±0.09)	66.77(±0.11)	60.31(±0.08)	75.21(±0.05)	71.95(±0.05)	65.91(±0.10)
DyTox-2K	71.47(±0.08)	68.37(±0.07)	63.59(±0.14)	75.95(±0.22)	73.24(±0.17)	69.75(±0.12)
w/ EOPC	72.02(±0.09)	69.47(±0.08)	64.74(±0.07)	76.40(±0.02)	74.14(±0.03)	70.42(±0.07)

Table 3. Repeated adaptation and comparison results of EOPC on CIFAR-100 and ImageNet-100. Results are reported with mean and standard deviation. We allocate 2 GPUs for the distributed training of DyTox and use the distributed memory option. For a sequence of tasks containing 100 classes, the total memory size of existing CIL methods is commonly set to 2K (i.e., 20 representative samples for each class). To keep fairness, DyTox leaves a total memory size of 1K for each GPU process. We also provide the results of 2K for each GPU process, denoted by DyTox-2K.

E. Companions with Existing Connectivity Finding Methods

E.1. Experimental Scheme

We compare Fourier-based OPC with three kinds of existing non-linear mode connectivity finding methods, including polygonal chain (Garipov et al., 2018), Bezier curve (Garipov et al., 2018), and simplicial complexes (Benton et al., 2021). We use these methods to find a low-loss path between two adjacent continual minima according to Equation (4). Therefore, a switching point (SP) $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda^*)$ needs to be appointed on the path for bridging previous tasks and the new task. Different from the original definition of the polygonal chain, we reformulate it to take λ^* as the SP and the parameterized path is,

$$\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda) = \begin{cases} (1 - \frac{\lambda}{\lambda^*})\hat{\boldsymbol{w}}_{t-1} + \frac{\lambda}{\lambda^*}\boldsymbol{\theta}, & 0 \leq \lambda \leq \lambda^* \\ (1 - \frac{\lambda - \lambda^*}{1 - \lambda^*})\boldsymbol{\theta} + \frac{\lambda - \lambda^*}{1 - \lambda^*}\boldsymbol{w}_t, & \lambda^* \leq \lambda \leq 1, \end{cases} \tag{23}$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$. The simplicial complexes is a generalization of polygonal chain, which needs multiple iterations of polygonal chain to build a low-loss volume, we use two connectors to connect adjacent continual minima, and it eventually constructs

Method	Params	CIFAR-100		
$\mathcal{A}$ (%) ↑		5	10	25
PODNet	0	65.07(±0.28)	62.93(±0.14)	59.45(±0.28)
w/ Line	0	64.85(±0.30)	60.72(±0.37)	56.63(±0.06)
w/ Polygonal Chain	5×10^5	64.41(±0.18)	60.02(±0.10)	52.40(±0.18)
w/ Bezier Curve	5×10^5	**66.81**(±0.08)	64.06(±0.07)	57.59(±0.12)
w/ Second-order Bezier Curve	10×10^5	66.56(±0.11)	62.69(±0.04)	55.02(±0.23)
w/ Simplicial Complexes	10×10^5	66.73(±0.06)	63.71(±0.01)	57.33(±0.23)
w/ Fourier-based OPC	$2 \times 96 \times 4$	66.53(±0.06)	**64.53**(±0.34)	**62.00**(±0.16)

Table 4. Results of comparing Fourier-based OPC with existing connectivity finding methods on CIFAR-100. We run each experiment 3 times and report their mean and standard deviation. "Params" is the number of parameters that need to be learned for each method. The continual model used for the classification task of CIFAR-100 contains approximately 5×10^5 parameters and 96 learnable layers.

two triangles with these four points (two connectors and two minima) in parameter space. The Bezier curve is the same as the original definition, i.e., $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda) = (1-\lambda)^2\hat{\boldsymbol{w}}_{t-1} + 2\lambda(1-\lambda)\boldsymbol{\theta} + \lambda^2\boldsymbol{w}_t$. We also consider the second-order Bezier curve with two bends, its formulation is,

$$\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda) = (1-\lambda)^3\hat{\boldsymbol{w}}_{t-1} + 3\lambda(1-\lambda)^2\boldsymbol{\theta}_1 + 3(1-\lambda)\lambda^2\boldsymbol{\theta}_2 + \lambda^3\boldsymbol{w}_t, \tag{24}$$

where parameters of the path $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2] \in \mathbb{R}^{n\times 2}$. The proposed Fourier-based OPC is defined in Equation (10). For polygonal chain, Bezier curve, and OPC, we use SP $\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda^*)$ as the substitute for endpoints. For simplicial complexes, we take the center of two connectors as the substitute for endpoints. Table 4 shows the results of these non-linear mode connectivity finding methods when adapting to PODNet. It can be seen that OPC achieves comparable performance with Bezier curve for 5 steps, and get the best results for 10 and 25 steps of incremental learning. In addition, we count the number of parameters that need to be learned for each method. It can be seen that the number of parameters that OPC needs is significantly fewer than other mode connectivity finding methods, which indicates OPC is a parameter-saving algorithm.

E.2. Effectiveness Analysis

We try to analyze why Fourier-based OPC is better than other mode connectivity finding methods in terms of adapting to CIL. In the scenario of CIL, two adjacent continual minima tend to stay in regions with big differences. As shown in the first row of Figure 10, for task $t \geq 3$, the minimum of the previous task $\hat{\boldsymbol{w}}_{t-1}$ stays in the low-loss region of previous tasks, on the contrary, the minimum of the new task $\boldsymbol{w}_t$ stays in the high-loss ridge of previous tasks, this is a reflection of the catastrophic forgetting problem in the loss landscape. In this work, we try to find a low-loss path connecting adjacent continual minima. To minimize forgetting previous tasks to the maximum extent when moving from the previous minimum to the new minimum, *the path must walk along the same direction as the contour of a region*, since the change in loss is zero along the contour. In addition, *the path ultimately needs to pass through the new minimum, which forces the path to shift to the direction of the next contour and creates a bend in the path.*

These two requirements are very challenging for existing non-linear mode connectivity finding methods. For the polygonal chain and the Bezier curve, they can only construct a path with multiple bends in a parameter-inefficient way. As results are shown in Table 4, the Bezier curve with one bend achieves the best result for relatively simple 5-step learning. However, with the increasing of steps, forgetting is aggravating, the path needs to shift from one contour direction to another multiple times. Therefore, the path constructed by existing connectivity finding methods with only one or two bends can not meet this requirement, and results in Table 4 show that polygonal chain and the Bezier curve deteriorate the adaptation baseline when learning for 25 steps. The simplicial complexes construct a low-loss volume between minima and can improve the baseline for 5 and 10 steps, however, there will exist a similar problem as the Bezier curve when learning for 25 steps, it can not flexibly shift between different loss regions. The proposed OPC is based on the Fourier series and scaling model parameters in a layer-wise manner to construct a space curve in subspace $\mathbb{R}^{L+1}$, it exhibits more flexibility and parameter efficiency compared with existing methods. Extensive visualization results shown in Figures 8, 9, and 10 demonstrate the OPC optimized path tends to walk along the contours of the ridge for staying in the low-loss region as far as possible. Therefore, the Fourier-based OPC can perform better than other non-linear connectivity finding methods when adapting to CIL in a more parameter-saving way.

F. Dealing with the Steep Change of $\mathcal{L}$

As shown in Equation (4), the loss $\mathcal{L}$ used for calculation $\ell(\theta)$ is different before and after λ^*. We visualize the OPC-optimized paths during the process of optimization and calculate the memory loss along paths, results are shown in Figure 7. It can be seen that as the path is continuously pulled to the low-loss region in the loss landscape of previous tasks, there exist steep changes of $\mathcal{L}$ along the optimized paths. We find that this disconnectivity will affect the stability of optimization, especially for a long sequence of tasks. To deal with this problem, we take the following three measures:

- First, $\mathcal{L}$ is weighted according to its position, i.e., λ. The weighting rule is: $\mathcal{L} = \mathcal{L} * (\text{abs}(\lambda - 0.5) + \rho)$, where ρ is a preset balance coefficient, commonly set to 0.1. Therefore, points near the middle of the path will be assigned to smaller weights, and weaken the instability from discontinuity.
- Second, we expand the calculation range of $\mathcal{L}_{1:t-1}$ in Equation (4) to the whole interval [0, 1]. This will prioritize ensuring the continuity of $\mathcal{L}_{1:t-1}$ as catastrophic forgetting is the main problem of continual learning.
- Third, we empirically find that setting a nonzero weight decay on the parameters of the path (i.e., $\boldsymbol{\theta}$) will make optimization more stable. We commonly set the same value as the weight decay of the continual learning model.

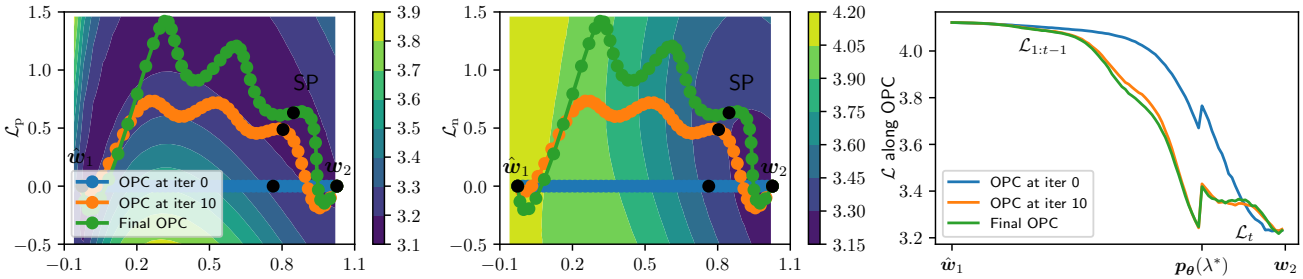

Figure 7. The loss landscapes and loss $\mathcal{L}$ curves when OPC optimizing path between $\hat{\boldsymbol{w}}_1$ and $\boldsymbol{w}_2$. **Left:** The loss landscape of previous tasks (task 1). **Center:** The loss landscape of the new task (task 2). **Right:** The loss $\mathcal{L}$ curves along OPC-optimized paths, the part between $[\hat{\boldsymbol{w}}_1, \boldsymbol{p}_{\boldsymbol{\theta}}(\lambda^*)]$ is the loss $\mathcal{L}_{1:t-1}$ on the memory of previous tasks, the part between $[\boldsymbol{p}_{\boldsymbol{\theta}}(\lambda^*), \boldsymbol{w}_2]$ is the loss $\mathcal{L}_t$ on the memory of the new task. There exist steep changes of $\mathcal{L}$ at the SP points along paths.

G. Visualization

To study the effects of order N in the Fourier series (Equation 10), we visualize the low-loss path found by OPC with different orders in Figure 8. Furthermore, to study the effects of the switching point, we visualize the low-loss path found by OPC with different values of λ^* in Figure 9. Finally, we visualize the detailed results of OPC finding the low-loss path between two adjacent continual minima in Figure 10. These results are obtained based on PODNet (Douillard et al., 2020) on CIFAR-100 for 5 steps of incremental learning.

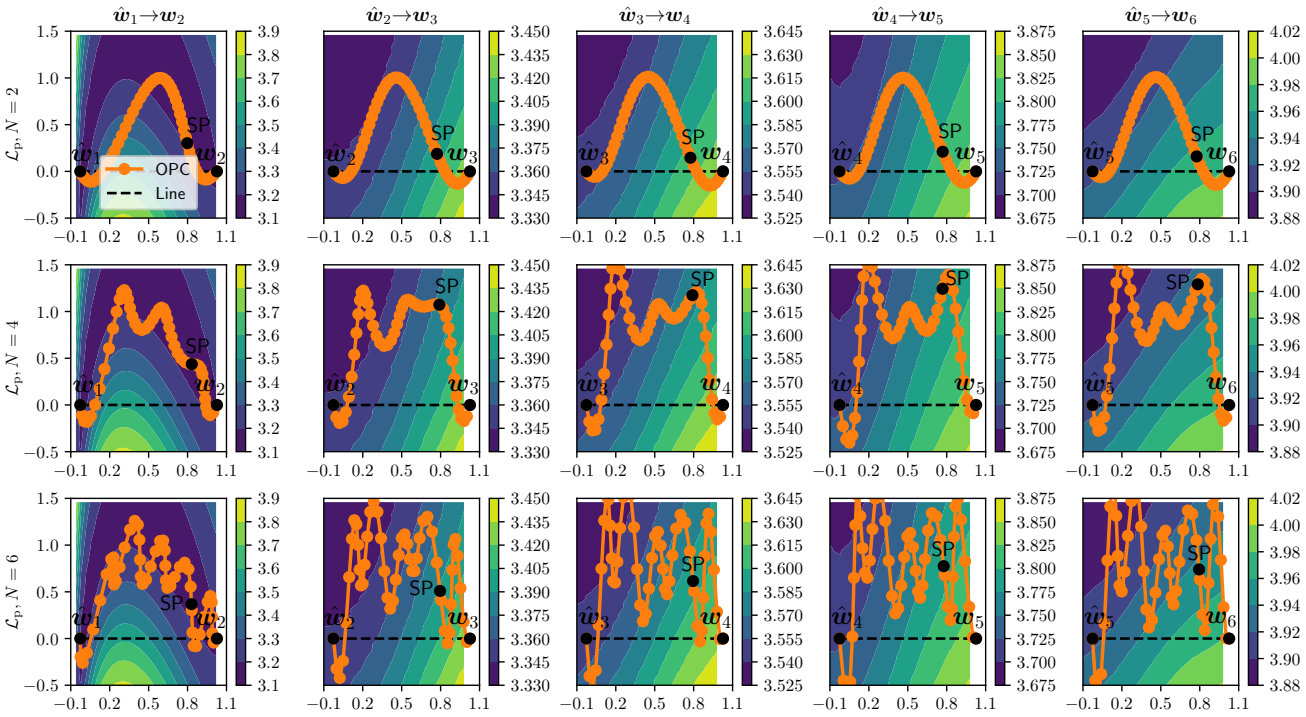

Figure 8. Results of OPC finding the low-loss path with different orders of Fourier series ($N \in \{2, 4, 6\}$), $\mathcal{L}_\mathrm{p}$ means the loss landscape of previous tasks. It can be seen that the path found with $N = 2$ is smooth, although part of the path stays in a lower-loss region compared with the linear connection, most of the path still stays on the high-loss ridge. When $N = 6$, the path found by OPC fluctuates severely, although more part of the path stays in the low-loss region compared with $N = 2$, it will make the switching point unstable and affect the function of ensembling. In contrast, the path found with $N = 4$ obtains a good balance between $N = 2$ and $N = 6$.

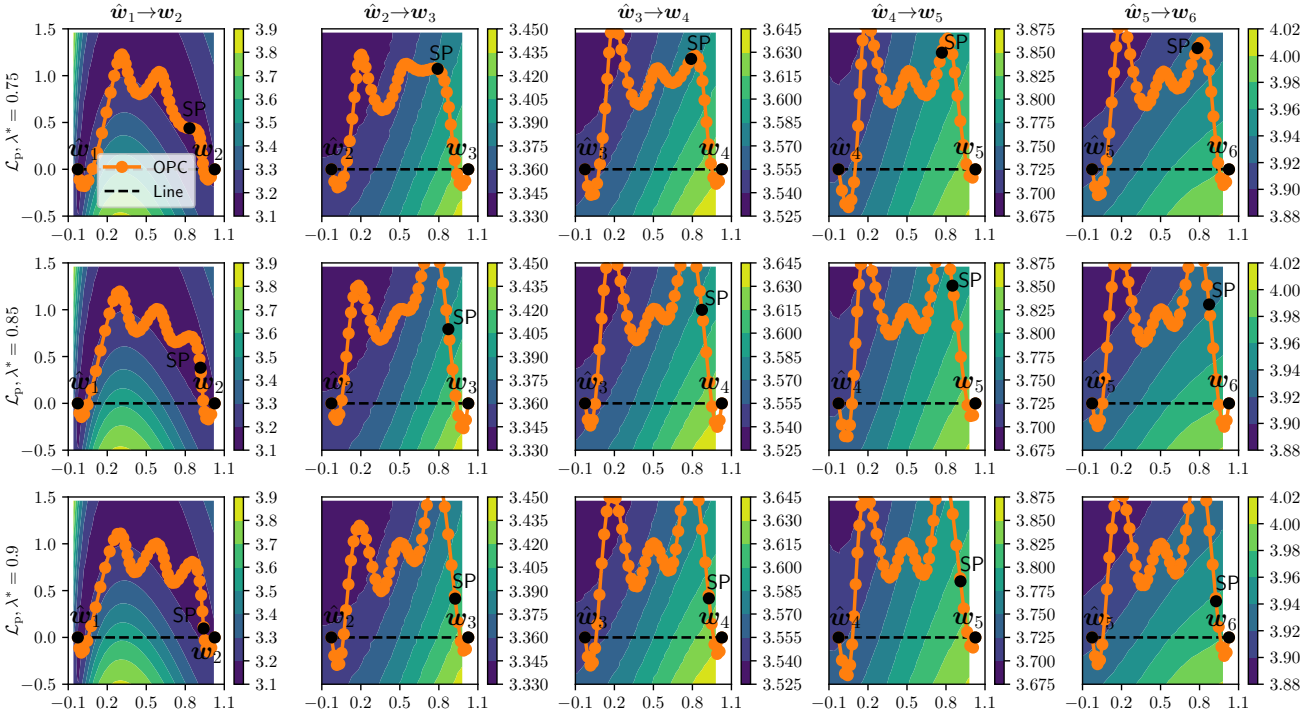

Figure 9. Results of OPC finding the low-loss path with different positions of the switching point ($\lambda^* \in \{0.75, 0.85, 0.9\}$), $\mathcal{L}_{\mathrm{p}}$ means the loss landscape of previous tasks. For a different SP, the final state of the path can be divided into two situations. For $t = 2$, two minima are blocked by the high-loss ridge but can be connected with a non-linear low-loss path. There is not much difference between the optimized paths with different SP points, they all successfully stay in the low-loss region. For $t \geq 3$, two adjacent continual minima stay in regions with big differences, the previous minimum stays in the low-loss region and the new stays on the high-loss ridge. In this case, with a larger value of λ^*, the optimized path will extend outward along the direction of the contour to stay in the low-loss region as far as possible, however, this will make the SP stay in a higher loss region. Therefore, we need to balance the position of SP and the demand that more part of the path is in the low-loss region.

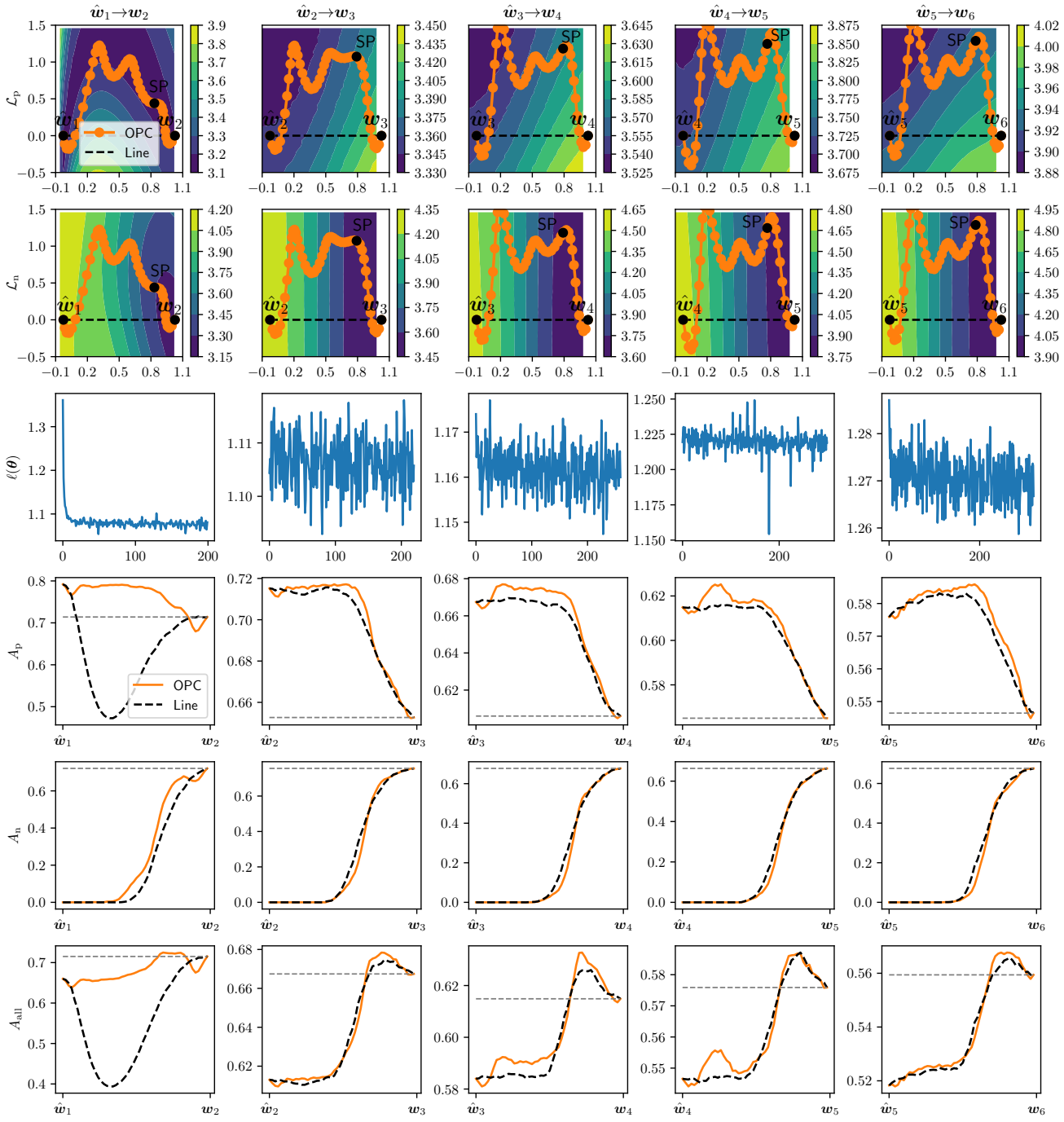

Figure 10. Detailed results of OPC finding the low-loss path between two adjacent continual minima. $\mathcal{L}_\mathrm{p}$, $\mathcal{L}_\mathrm{n}$, $\ell(\boldsymbol{\theta})$, A_p, A_n, and A_all are training loss on previous tasks, training loss on the new task, training loss of OPC on the memory $\mathcal{M}$, testing accuracy on previous tasks, testing accuracy on the new task, and testing accuracy on all learned tasks. It can be seen that the path found by OPC tends to stay in a lower-loss region in the training loss landscape of previous tasks compared with the linear connection. In addition, because we introduce the switching point in Equation (4), and assign different parts of the path to previous tasks and the new task, the testing accuracy of the new task along the path is not greatly affected. Therefore, there is an interval on the path that achieves higher accuracy on learned tasks. This confirms that OPC achieves higher CIL performance by mainly reducing forgetting of previous tasks. Furthermore, we can observe that the training loss of OPC along the path is rapidly decreased when minima are clearly blocked by the high-loss ridge ($\hat{\boldsymbol{w}}_1 \to \boldsymbol{w}_2$), which demonstrates OPC is an efficient algorithm for finding the connectivity between minima.