MEASURING ASYMMETRIC GRADIENT DISCREPANCY IN PARALLEL CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

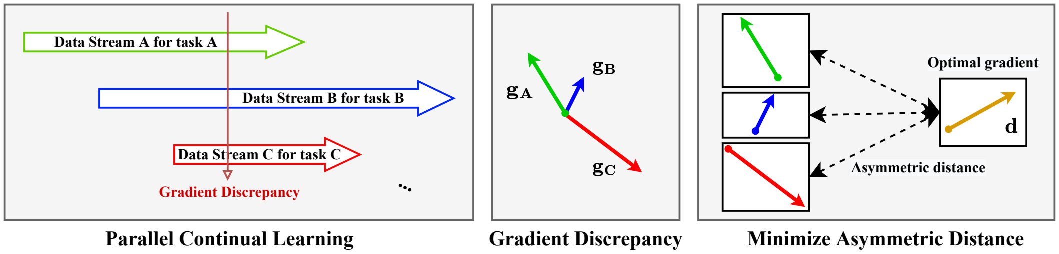
In Parallel Continual Learning (PCL), the parallel multiple tasks start and end training unpredictably, thus suffering from training conflict and catastrophic forgetting issues. The two issues are raised because the gradients from parallel tasks differ in directions and magnitudes. Thus, in this paper, we formulate the PCL into a minimum distance optimization problem among gradients and propose an explicit Asymmetric Gradient Distance (AGD) to evaluate the gradient discrepancy in PCL. AGD considers both gradient magnitude ratios and directions, and has a tolerance when updating with a small gradient of inverse direction, which reduces the imbalanced influence of gradients on parallel task training. Moreover, we propose a novel Maximum Discrepancy Optimization (MaxDO) strategy to minimize the maximum discrepancy among multiple gradients. Solving by MaxDO with AGD, parallel training reduces the influence of the training conflict and suppresses the catastrophic forgetting of finished tasks. Extensive experiments validate the effectiveness of our approach on three image recognition datasets.

1 INTRODUCTION

Continual Learning (CL) (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Lopez-Paz & Ranzato, 2017), aims to continuously learn new knowledge from a sequence of tasks with non-overlapping data streams over a lifelong time. In the era of Internet of Things (IoT), people are using many smart devices, where data and tasks would be accessed by the learning system at any time. It is necessary for a CL system to respond to parallel data streams from multiple devices. We study *Parallel Continual Learning* (PCL), as shown in Fig. 1, in which an unfixed number of tasks are trained in a parallel way at any time. Specifically, according to the access time of each task, PCL builds an adaptive number of parallel data pipes, thus enabling instant response to new-coming tasks without pending.

Due to the parallel data streams from different tasks, PCL suffers from not only the *catastrophic forgetting* but the *training conflict* among parallel tasks. Most existing methods in CL are proposed to tackle the catastrophic forgetting (French, 1999; Kirkpatrick et al., 2017) of any finished tasks, including regularization-based (Kirkpatrick et al., 2017; Chaudhry et al., 2018; Dhar et al., 2019; Zenke et al., 2017; Aljundi et al., 2018), rehearsal-based (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Guo et al., 2019; Atkinson et al., 2018; Shah et al., 2018; Pomponi et al., 2020), and architecture-based (Mallya et al., 2018; Yoon et al., 2017; Rusu et al., 2016; Rosenfeld & Tsotsos, 2018) methods. In PCL, the training processes of different tasks are diverse, *i.e.*, each task starts and ends training unpredictably (See Fig. 1). Thereby the gradient from different task differs in direction and magnitude (Yu et al., 2020) and may be neutralized. The gradient discrepancies lead to catastrophic forgetting and training conflict issues, which may fail the learning of some tasks. At any time in PCL, therefore, we present that the problem can be formulated to find an optimal gradient in a *minimum distance multi-objective optimization*, where each objective is to minimize the distance to a target gradient. In general, the distance metric is proportional to the effect of the optimal gradient on the corresponding task.

In most situations, the mentioned distance metric D between gradients is set to symmetric intuitively, such as the Euclidean distance and cosine distance. In other words, we usually have $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$ for any $\mathbf{x}$ and $\mathbf{y}$. However, the gradient influence is imbalanced among parallel tasks in the gradient descent. For example in Fig. 1, at the marked time, we have three gradients with diverse directions and magnitudes, and updating with any of them provides different influences to the other



Parallel Continual Learning

Gradient Discrepancy

Minimize Asymmetric Distance

Figure 1: Overview of the proposed method in PCL. Left: PCL trains parallel tasks according to their access time without pending. Middle: At any time, gradients from different tasks (corresponding colors) have unpredicted direction and magnitude (the length of vectors). Right: We formulate PCL into a min-distance problem and propose an asymmetric distance for effective optimization.

two. In the minimum distance problem, the optimal solution should have the minimum negative influence on all parallel tasks, but using symmetric metrics means the influences are optimized indistinguishably at the same time. Due to the fact that the gradients are with wide differences, the solution may have large biases, which would get the near-fitting task out of its local minimum but has less impact on a new-coming task.

To measure the gradient discrepancy, we hold the opinion that the distance metric in the min-distance problem should be *asymmetric*. First, though the metric is bound up with both the gradient magnitude and direction, the influences on model training from gradients should be asymmetric, where the model should have more tolerance to small gradients even if they indicate an inverse direction. Second, because gradients are with different magnitudes, the discrepancy between two large gradients is often set to larger than that between small gradients when using symmetric distance, such as Euclidean distance. Directly optimizing using magnitude-aware distance values may lead to the solution close to large gradients and thus hinder the catastrophic forgetting of old tasks. To mitigate the bias from the magnitude difference, it is better to employ the magnitude ratio instead of magnitude itself.

Motivated by this, in this paper, we propose an explicit measurement for the learning from gradient discrepancy in PCL, named Asymmetric Gradient Distance (AGD), which considers gradient magnitude ratios and directions, and sets a tolerance for smaller gradients. As shown in Fig. 1, the proposed AGD is used in solving the minimum distance problem with multiple gradients from parallel tasks. Then, we propose an effective optimization strategy for minimizing the gradient discrepancy to avoid self-interference. We name the strategy Maximum Discrepancy Optimization (MaxDO), which minimizes the maximum discrepancy from each gradient to the others. Moreover, to address the catastrophic forgetting issue, we follow the rehearsal strategy (Lopez-Paz & Ranzato, 2017) in traditional CL and build an extra memory data stream. The rehearsal data stream is used to provide a gradient of finished tasks in MaxDO. Solving by MaxDO with AGD, parallel training mitigates the impacts of the diverse training process and slows the catastrophic forgetting of finished tasks. Extensive results on three datasets show the superiority and effectiveness of our approach.

Our main contributions are three-fold:

(1) For the first time, we formulate the PCL into a minimum distance problem and compare symmetric and asymmetric distances. Considering the influence of gradient on task training, we show that symmetric metrics are not effective in solving the problem and suggest asymmetric metrics.

(2) We propose an asymmetric metric, named AGD, to evaluate the gradient discrepancy, which is proportional to the gradient magnitude ratios and directions. AGD takes the diverse training process into account and measures the imbalance of gradient influence on task training.

(3) We propose a MaxDO strategy for minimizing gradient discrepancy of different tasks, which maximumly reduces the asymmetric discrepancy from a gradient to the others. MaxDO avoids the self-interference among gradients and reduces the training conflict and catastrophic forgetting.

2 Related Work

Continual Learning (CL) represents receiving data from new domains continually. In traditional CL, the new domains show up one by one, say serial CL. CL methods can be classified into three kinds. (1) *Rehearsal* (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Guo et al., 2019; Atkinson et al., 2018; Shah et al., 2018; Pomponi et al., 2020), which saves or generates data of old tasks for retraining together with the current training. (2) *Regularization* (Kirkpatrick et al., 2017; Chaudhry et al., 2018; Dhar et al., 2019; Zenke et al., 2017; Aljundi et al., 2018; Farajtabar et al., 2020), which leverages extra regularization terms to consolidate previous knowledge when learning new tasks. (3) *Dynamic architecture* (Mallya et al., 2018; Yoon et al., 2017; Rusu et al., 2016; Rosenfeld & Tsotsos, 2018), which freezes task-specific parameters and grows new branches for new tasks automatically. However, most of the existing CL methods are designed for reducing catastrophic forgetting in the serial scenario. Contrastively, in PCL, we need to tackle not only catastrophic forgetting but training conflict among parallel tasks, which is somehow related to multi-task learning.

Multi-Task Learning (MTL) (Caruana, 1997) is used to address multiple tasks with a single model from one to many domains. Traditional MTL solutions can be mainly grouped into feature-based and parameter-based approaches (Zhang & Yang, 2021). The feature-based approaches focus on learning common feature representations for multiple tasks (Maurer et al., 2013; Wang & Ye, 2015). The parameter-based approaches use model parameters in a task to help learn model parameters in other tasks, such as task clustering (Thrun & O'Sullivan, 1996; Barzilai & Crammer, 2015) and decomposition (Jalali et al., 2010). In recent years, some MTL methods formulate the problem into finding an optimal gradient for updating and can be categorized into three types. (1) *Learning-based* methods (Chen et al., 2018), which learn a set of weights by backpropagation. (2) *Solving-based* methods (Sener & Koltun, 2018; Liu et al., 2021), solve the problem by finding an optimal gradient that is not dominated by the gradient from any task. (3) *Calculating-based* methods (Liu et al., 2019; Javaloy & Valera, 2021; Chen et al., 2020; Wang et al., 2020; Yu et al., 2020; Groenendijk et al., 2021; Lin et al., 2021) compute the gradient weights by combining gradients or losses of all tasks. Inspired by MTL, we also formulate the problem into finding an optimal gradient. Specifically, we consider the optimal gradient should have a small distance to all gradients.

Asymmetric Metric. In most situations, the distance is set to symmetric, e.g., the Euclidean distance. However, the symmetric metric is not always suitable for finding the optimal gradient (see the next section for details). Asymmetric metric (Collins & Zimmer, 2007; Mennucci, 2013), also known as quasi-metric (Collins & Zimmer, 2007) or pseudo metric (Fiol, 2001; Cagliari et al., 2015) is a generalization of a metric but the symmetry axiom is eliminated in the definition of metric spaces. A classical example of using asymmetric metric is the taxicab geometry topology including one-way streets, where a path from point A to B has different streets compared to a path from B to A. In this paper, we propose to measure the gradient discrepancy using an asymmetric metric and raise a novel optimization strategy to minimize the maximum discrepancy.

3 Our Approach

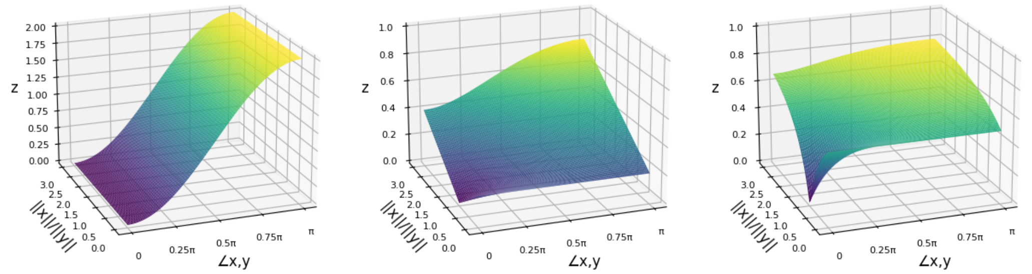
3.1 Parallel Continual Learning

On a timeline, given a sequence of T tasks with parallel data streams $\{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$ for continual training, and each data stream can be accessed and suspended at any time. For simplest, we assume each data stream is i.i.d., and tasks are accessed in order from 1 to T and there exists no real gap that no data stream flows on the timeline. Note that traditional CL is an edge situation of PCL that all tasks are nose-to-tail. A PCL model contains a *shared backbone* with parameter $\boldsymbol{\theta}$ to learn task-agnostic knowledge and adaptively incremental number of *task-specific classifiers* with parameters $\boldsymbol{\theta}_i$. When a new task is accessed, a corresponding task-specific classifier will be constructed.

In PCL, a task will be forgotten by learning any other tasks when its data stream ends. To avoid forgetting, we leverage the popular *rehearsal* strategy (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Guo et al., 2019; Atkinson et al., 2018; Shah et al., 2018; Pomponi et al., 2020) in our training. Rehearsal builds an extra data stream sampled from all seen tasks and retrains them to suppress the forgetting of finished tasks. For convenience, we denote the rehearsal data stream as $\mathcal{D}_0$. At time t, we use $\mathcal{T}_t$ to represent the activated data streams (including $\mathcal{D}_0$). Together with the rehearsal data stream, PCL training yields the following dynamic multi-objective empirical risk minimization:

$$\min\nolimits_{\boldsymbol{\theta}, \{\boldsymbol{\theta}_i | i \in \mathcal{T}_t\}} \quad \left\{ \ell_i \left(\mathcal{D}_i \right) | i \in \mathcal{T}_t \right\}. \tag{1}$$

Because the task-specific classifiers are updated by their own gradients $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \alpha_i \nabla_{\boldsymbol{\theta}_i} \ell_i$ $(\forall i \in \mathcal{T}_t)$ with step size α_i, we focus on the update of the shared backbone $\boldsymbol{\theta}$. At any PCL step, the goal of dynamic MOO is to optimize multiple objectives simultaneously while updating only once, and the only update of the shared parameters depends on the gradients of all in-training tasks. In PCL, the



(a) $z = 1 - \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$ (b) $z = \|\mathbf{x} - \mathbf{y}\|,\ \|\mathbf{y}\| = 0.2$ (c) $z = \frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|+\|\mathbf{x}-\mathbf{y}\|}$

Figure 2: The measures of two gradient discrepancy from $\mathbf{x}$ to $\mathbf{y}$. Note that the x- and y-axes are the angle (i.e., $\angle \mathbf{x}, \mathbf{y}$) between $\mathbf{x}$ and $\mathbf{y}$, and the magnitude ratio $\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}$, respectively. (a) Cosine distance; (b) Euclidean distance where $\|\mathbf{y}\| = 0.2$ as an example; (c) Asymmetric gradient distance.

update of the shared parameters at any time depends on the gradients of all in-training tasks. It will exit an uncertain number of tasks, and each task will provide a task-specific gradient on the shared parameter $\boldsymbol{\theta}$. Let $\mathbf{g}_i = \nabla_{\boldsymbol{\theta}} \ell_i$ and α be a step size for optimization. The problem of the backbone update can be formulated as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \mathbf{d}^*, \quad \text{where } \mathbf{d}^* = f(\{\mathbf{g}_i | \forall i \in \mathcal{T}_t\}). \tag{2}$$

The key question is how to compute the optimal gradient $\mathbf{d}^*$ via the function $f(\cdot)$. In this paper, we define the function $f(\cdot)$ as a min-distance multi-objective problem by minimizing the gradient distance from all in-training tasks:

$$\mathbf{d}^* = \arg\min_{\mathbf{d}} \quad \{D(\mathbf{d}, \mathbf{g}_i) \mid \forall i \in \mathcal{T}\}, \tag{3}$$

where we need to identify what distance metric D is used to measure gradient discrepancy. The motivation of Eq. (3) is that for the task i in PCL, its own gradient $\mathbf{g}_i$ is the most qualified update direction for itself. The solution $\mathbf{d}^*$ should be as close to every gradients as possible.

3.2 Measuring Asymmetric Gradient Discrepancy

To measure the gradient discrepancy, the Euclidean Distance (EuDist, $D(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| \in [0, \infty)$) and Cosine Distance (CosDist, $D(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \in [0, 2]$) are the two most popular choices. Both of them are symmetric, *i.e.*, $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$. A symmetric metric $D(\mathbf{x}, \mathbf{y})$ means the forward influence ($\mathbf{x}$ to $\mathbf{y}$) and backward influence ($\mathbf{y}$ to $\mathbf{x}$) are treated as symmetric. For example, given two in-training tasks A and B, the distance $D(\mathbf{g}_\text{A}, \mathbf{g}_\text{B})$ represents both the effect of $\mathbf{g}_\text{A}$ on task B and $\mathbf{g}_\text{B}$ on task A because of $D(\mathbf{g}_\text{A}, \mathbf{g}_\text{B}) = D(\mathbf{g}_\text{B}, \mathbf{g}_\text{A})$. Note that large distance from $\mathbf{g}_\text{A}$ to $\mathbf{g}_\text{B}$ means large negative influence on the training of task B with $\mathbf{g}_\text{A}$.

However, the model update is highly related to gradient magnitude and direction, which are asymmetric to model updating. The influence of the gradient $\mathbf{g}_\text{A}$ on task B may be quite different from that of the gradient $\mathbf{g}_\text{B}$ on task A. In previous studies (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Yu et al., 2020), the two tasks are treated as conflict when $\langle \mathbf{g}_\text{A}, \mathbf{g}_\text{B} \rangle < 0$. In PCL, due to the diverse training process, gradients from parallel tasks are diverse in magnitude and direction. When $\|\mathbf{g}_\text{A}\| \ll \|\mathbf{g}_\text{B}\|$, the gradient $\mathbf{g}_\text{A}$ will have little negative influence on task B even if $\langle \mathbf{g}_\text{A}, \mathbf{g}_\text{B} \rangle < 0$; when $\|\mathbf{g}_\text{A}\| \gg \|\mathbf{g}_\text{B}\|$ (e.g., a new task A is accessed when task B has been trained for some time near convergence), the update produces huge impact on task B even if $\langle \mathbf{g}_\text{A}, \mathbf{g}_\text{B} \rangle > 0$. Using the traditional symmetric distance can hardly represent the asymmetric update influence difference.

To effectively measure gradient discrepancy in PCL, we introduce the asymmetric metric.

Lemma 1 (Asymmetric Metric (Collins & Zimmer, 2007)) $D : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is an asymmetric metric (also known as quasi-metric (Wilson, 1931)) if* D *satisfies*

(1) $D(\mathbf{x}, \mathbf{y}) \geq 0$ *and* $\forall \mathbf{x} \in \mathbb{R}^d$, $D(\mathbf{x}, \mathbf{x}) = 0$;
(2) $D(\mathbf{x}, \mathbf{z}) \leq D(\mathbf{x}, \mathbf{y}) + D(\mathbf{y}, \mathbf{z}), \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^d$.

The asymmetric metric does not require the symmetric property, *i.e.*, $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$. Based on the definition, in this paper, we design an asymmetric metric to measure gradient discrepancy named Asymmetric Gradient Distance.

Definition 1 (Asymmetric Gradient Distance (AGD)) *Given two gradient $\mathbf{g}_A$ and $\mathbf{g}_B$, the asymmetric gradient distance is defined as*

$$\widehat{D}(\mathbf{g}_A, \mathbf{g}_B) = \begin{cases} 0 & , \quad \textit{if } \mathbf{g}_A = \mathbf{g}_B = \mathbf{0}, \\ \dfrac{\|\mathbf{g}_A - \mathbf{g}_B\|}{\|\mathbf{g}_B\| + \|\mathbf{g}_A - \mathbf{g}_B\|}, & \quad \textit{Otherwise}. \end{cases} \tag{4}$$

In Definition 1, we consider the edge situation when $\mathbf{g}_\mathrm{A} = \mathbf{g}_\mathrm{B} = \mathbf{0}$ to meet the definition of the asymmetric metric in Lemma 1. In AGD, gradient directions and magnitudes are considered. Instead of using gradient magnitude value difference, we use magnitude ratio difference to avoid the diverse training of different tasks in PCL. Therefore, we derive the corollary of the magnitude ratio:

Corollary 1 $\widehat{D}(\mathbf{g}_A, \mathbf{g}_B) \in [0, 1]$ *is an asymmetric metric and holds*

$$\lim_{\frac{\|\mathbf{g}_A\|}{\|\mathbf{g}_B\|} \to \infty} \widehat{D}(\mathbf{g}_A, \mathbf{g}_B) = 1, \quad \lim_{\frac{\|\mathbf{g}_A\|}{\|\mathbf{g}_B\|} \to 0} \widehat{D}(\mathbf{g}_A, \mathbf{g}_B) = \frac{1}{2}. \tag{5}$$

We illustrate why AGD is qualified to evaluate the gradient discrepancy according to the definition and corollary. In Definition 1, we use AGD to represent the influence of $\mathbf{g}_\mathrm{A}$ on task B rather than the inverse. This is the key difference from the symmetric metrics such as Euclidean distance. Specifically, $\mathbf{g}_\mathrm{A}$ may make task B worse if $\widehat{D}(\mathbf{g}_\mathrm{A}, \mathbf{g}_\mathrm{B})$ is large (close to 1). If $\widehat{D}(\mathbf{g}_\mathrm{A}, \mathbf{g}_\mathrm{B})$ is close to 0, $\mathbf{g}_\mathrm{A}$ and $\mathbf{g}_\mathrm{B}$ has less conflict. Moreover, Corollary 1 involves that when $\|\mathbf{g}_\mathrm{A}\| \ll \|\mathbf{g}_\mathrm{B}\|$, AGD has a tolerance $\frac{1}{2}$ even if $\langle \mathbf{g}_\mathrm{A}, \mathbf{g}_\mathrm{B} \rangle < 0$, which means the impact of $\mathbf{g}_\mathrm{A}$ on task B is mild. This is because updating with a zero gradient will neither improve nor damage the performance. Even though, we prefer positive influence rather than non-influence. Thus, we define that the distance $\widehat{D}(\mathbf{g}_\mathrm{A}, \mathbf{g}_\mathrm{B})$ in this situation is the mid-level in the value range. See different tolerances in Appendix C.

Moreover, we compare AGD (Fig. 2(c)) with Euclidean and cosine distance in Fig. 2. First, the cosine distance (Fig. 2(a)) is magnitude irrelevant, which ignores the magnitude difference in PCL. Second, the Euclidean distance (Fig. 2(b)) depends heavily on the magnitude value difference, but ignores that the gradient influence on the model update is asymmetric. For example, when $\|\mathbf{x}\| \to 0$, EuDist will get large if we have large $\|\mathbf{y}\|$. However, updating with a zero gradient will neither improve nor damage the performance. See the contours of Fig. 2 in Appendix D.

3.3 Maximum Discrepancy Optimization

At time t, let the optimal solution to Problem (3) be $\mathbf{d}^*$, where $\mathcal{T}_t$ is the index set of in-training tasks ($\mathcal{T}$ for simplicity). However, directly optimizing the problem is difficult due to the large decision space that has the same dimension as $\boldsymbol{\theta}$. Following (Lin et al., 2021; Sener & Koltun, 2018), we use linear scalarization to solve the transformed problem that allows only optimizing decision variable $\mathbf{w} \in \mathbb{R}^{|\mathcal{T}|}$. That is, let $\mathbf{d} = \sum_{i \in \mathcal{T}} \mathbf{w}_i \mathbf{g}_i$, where $\forall \mathbf{w}_i \geq 0$ and $\sum_{i \in \mathcal{T}} \mathbf{w}_i = 1$, we have

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \quad \left\{ \widehat{D}\left(\sum_j \mathbf{w}_j \mathbf{g}_j, \mathbf{g}_i \right) \middle| \forall i \in \mathcal{T} \right\}. \tag{6}$$

Each objective of the dual problem will be highly affected by the minimum discrepancy, *i.e.*, each gradient itself. For example, by minimizing objective $\widehat{D}(\sum_j \mathbf{w}_j \mathbf{g}_j, \mathbf{g}_i)$, weight $\mathbf{w}_i$ is more like to be activated than others. Thus, multiple objectives will be compromised by multiple self-interference but fail to reduce the maximum discrepancy in the dual problem optimization.

As shown in Fig. 3., we propose Maximum Discrepancy Optimization (MaxDO) to reduce the maximum gradient discrepancy. Specifically, instead of the weight vector $\mathbf{w} \in \mathbb{R}^{|\mathcal{T}|}$, we optimize a weight matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$, in which $\forall \mathbf{W}_{ij} \geq 0$. $\mathbf{W}$ can be combined by a diagonal vector $\dot{\mathbf{w}} = [\mathbf{W}_{1,1}, \cdots, \mathbf{W}_{|\mathcal{T}|,|\mathcal{T}|}]$ and an off-diagonal matrix $\widetilde{\mathbf{W}} = \mathbf{W} - \mathrm{Diag}(\dot{\mathbf{w}})$, where $\sum_{i \in \mathcal{T}} \dot{\mathbf{w}}_i = 1$ and $\sum_{j \in \mathcal{T}} \widetilde{\mathbf{W}}_{ij} = 1, \forall i$. Thus, $\sum_{i,j \in \mathcal{T}} \mathbf{W}_{ij} = |\mathcal{T}| + 1$ and the two weights are independent and can be optimized without disturbance: (1) $\widetilde{\mathbf{W}}$, computed by Stochastic Gradient Descent (SGD), is used to make up the maximum gradient discrepancy for each row. The objectives of any two rows in $\widetilde{\mathbf{W}}$ are different and independent. For row i, to formulate the maximum discrepancy of gradient $\mathbf{g}_i$, the objective is the combination of non-diagonal entries. The weighted other gradients should

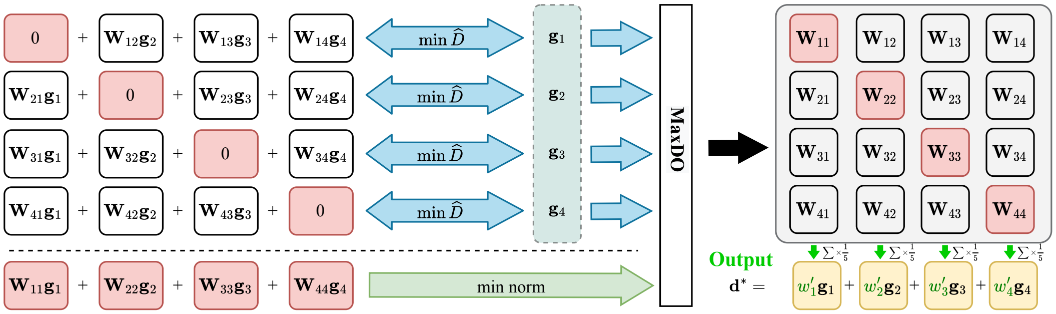

0 + $\mathbf{W}_{12}\mathbf{g}_2$ + $\mathbf{W}_{13}\mathbf{g}_3$ + $\mathbf{W}_{14}\mathbf{g}_4$
$\mathbf{W}_{21}\mathbf{g}_1$ + 0 + $\mathbf{W}_{23}\mathbf{g}_3$ + $\mathbf{W}_{24}\mathbf{g}_4$
$\mathbf{W}_{31}\mathbf{g}_1$ + $\mathbf{W}_{32}\mathbf{g}_2$ + 0 + $\mathbf{W}_{34}\mathbf{g}_4$
$\mathbf{W}_{41}\mathbf{g}_1$ + $\mathbf{W}_{42}\mathbf{g}_2$ + $\mathbf{W}_{43}\mathbf{g}_3$ + 0
min $\widehat{D}$
$\mathbf{g}_1$
$\mathbf{g}_2$
$\mathbf{g}_3$
$\mathbf{g}_4$
$\mathbf{W}_{11}\mathbf{g}_1$ + $\mathbf{W}_{22}\mathbf{g}_2$ + $\mathbf{W}_{33}\mathbf{g}_3$ + $\mathbf{W}_{44}\mathbf{g}_4$
min norm
MaxDO
$\mathbf{W}_{11}$ $\mathbf{W}_{12}$ $\mathbf{W}_{13}$ $\mathbf{W}_{14}$
$\mathbf{W}_{21}$ $\mathbf{W}_{22}$ $\mathbf{W}_{23}$ $\mathbf{W}_{24}$
$\mathbf{W}_{31}$ $\mathbf{W}_{32}$ $\mathbf{W}_{33}$ $\mathbf{W}_{34}$
$\mathbf{W}_{41}$ $\mathbf{W}_{42}$ $\mathbf{W}_{43}$ $\mathbf{W}_{44}$
$\Sigma\times\frac{1}{5}$
Output
$\mathbf{d}^* =$ $w'_1\mathbf{g}_1$ + $w'_2\mathbf{g}_2$ + $w'_3\mathbf{g}_3$ + $w'_4\mathbf{g}_4$

Figure 3: Schematic of Maximum Discrepancy Optimization. Given multiple gradients $\{\mathbf{g}_i | \forall i \in \mathcal{T}\}$ ($|\mathcal{T}| = 4$ for example) (1) A weight matrix $\mathbf{W}$ is initialized with $\frac{1}{|\mathcal{T}|}$ for each entry. (2) For each row, the off-diagonal entries are used to weighted gradients and optimized for minimum AGD to the target gradient. (3) The diagonal entries (🟥) are used to optimize with min-norm with MGDA. (4) The final weight matrix is reduced by each column for the final weights (w'). See Sec. 3.3 for details.

be with the smallest asymmetric distance to $\mathbf{g}_i$. (2) $\grave{\mathbf{w}}$ is obtained by the Multiple Gradient Descent Algorithm (MGDA) (Désidéri, 2012), which is to obtain a weighted gradient that does not damage any tasks with a *min-norm* optimization. The objective of MGDA is 0 and the resulting point satisfies the Karush–Kuhn–Tucker condition or the solution gives a Pareto descent direction that improves all tasks. See Appendix E for more details of MGDA. For each off-diagonal entry of the i-th column, their sum means the effect of the gradient $\mathbf{g}_i$ reducing the maximum discrepancy from other gradients. MGDA is used to reduce the possible negative effect in MaxDO. On the other hand, MaxDO reduces the training failure of new tasks in MGDA. To sum up, our MaxDO with AGD can be computed by

$$\mathbf{W}^* = \underset{\widetilde{\mathbf{W}}}{\arg\min} \underbrace{\left\{ \widehat{D} \left(\sum\nolimits_{j \neq i} \widetilde{\mathbf{W}}_{i,j} \mathbf{g}_j, \mathbf{g}_i \right) \middle| \forall i \in \mathcal{T} \right\}}_{\text{SGD with Maximum Discrepancy}} + \text{Diag} \underbrace{\left(\underset{\grave{\mathbf{w}}}{\arg\min} \left\| \sum\nolimits_j \grave{\mathbf{w}}_j \mathbf{g}_j \right\| \right)}_{\text{MGDA (Désidéri, 2012)}}. \tag{7}$$

In Eq. (7), we can obtain an approximate solution by combining the closed-form solution and the iterative solution. Fig. 3 reveals the diagram of solving MaxDO. We project the solution of SGD onto the feasible set ($\sum_{i \neq j} \mathbf{W}_{ij} = 1$) via softmax at each step in the multiple steps for fast convergence.. First, we initialize all entries of $\mathbf{W}$ by $\frac{1}{|\mathcal{T}|}$. Then, the off-diagonal matrix is used to minimize the maximum gradient discrepancy via SGD and the diagonal vector is optimized by min-norm. Finally, the final weights are reduced to a vector by dividing $|\mathcal{T}| + 1$ to guarantee that their sum is 1. Note that, MaxDO is implemented only when $|\mathcal{T}| > 1$, *i.e.*, multiple tasks are given at the current time. Otherwise, we have $\mathbf{d}^* = \mathbf{g}_1$ for the only current task 1. Thus, the final gradient $\mathbf{d}^*$ is computed by

$$\mathbf{d}^* = \begin{cases} \mathbf{g}_1, & |\mathcal{T}| = 1, \\ \sum_i \left(\frac{1}{|\mathcal{T}| + 1} \sum_j \mathbf{W}^*_{j,i} \right) \mathbf{g}_i, & |\mathcal{T}| > 1. \end{cases} \tag{8}$$

The detailed algorithm is shown in Algorithm 1. With the rehearsal data stream, our algorithm learns a PCL model through a timeline. At the time t on the timeline, given a mini-batch $\mathcal{B}$ from each data stream, we compute the corresponding gradients on shared and task-specific parameters. The task-specific parameters are updated directly and the gradients on the shared backbone are collected for computed the final updated gradient $\mathbf{d}$. By using our MaxDO, we update $\boldsymbol{\theta}$ with the optimal $\mathbf{d}^*$ and update the shared parameters.

Algorithm 1: MaxDO (🟪) in PCL

```
Input: Random-initialized $\boldsymbol{\theta}$, $\boldsymbol{\theta}_{1:T}$;
       Step sizes $\alpha$, $\alpha_{1:T}$
Output: $\boldsymbol{\theta}$, $\boldsymbol{\theta}_{1:T}$
for $t$ in timeline do
    $\mathcal{T}_t \leftarrow$ in-training task index;
    for $i \in \mathcal{T}_t$ do
        $\mathcal{B}_i \sim \mathcal{D}_i$;
        $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \alpha_i \nabla_{\boldsymbol{\theta}_i} \ell_i(\mathcal{B}_i)$;
        $\mathbf{g}_i = \nabla_{\boldsymbol{\theta}} \ell_i(\mathcal{B}_i)$;
    end
    $\mathbf{W}^* \leftarrow$ Optimization by Eq. (7);
    $\mathbf{d}^* \leftarrow$ Final graident from Eq. (8);
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha\mathbf{d}^*$;
end
```

4 Experiment

4.1 Dataset

In our experiments, 3 traditional image recognition datasets are transformed into parallel data streams: (1) *Parallel Split EMNIST (PS-EMNIST)*. We split EMNIST (Cohen et al., 2017) (62 classes) into

Table 1: Comparisons (avg ± std) with different buffer sizes on PS-EMNIST (62 classes).

Method (+ Rehearsal)	**Buffer size 124 (62*2)**		**Buffer size 186 (62*3)**		**Buffer size 310 (62*5)**	
	A_T (%)	F_T (%)	A_T (%)	F_T (%)	A_T (%)	F_T (%)
MGDA(*NeurIPS'18*)	78.975 ± 0.165	$-\mathbf{5.386} \pm 1.252$	82.026 ± 0.851	-7.215 ± 1.637	84.783 ± 0.190	-5.780 ± 0.476
GradNorm(*ICML'18*)	83.985 ± 0.324	-9.989 ± 0.578	85.127 ± 0.647	-8.835 ± 1.215	86.060 ± 0.094	-7.896 ± 0.208
DWA(*CVPR'18*)	85.416 ± 0.622	-8.209 ± 1.279	85.939 ± 0.632	-7.787 ± 1.255	86.732 ± 0.089	-6.922 ± 0.175
GradDrop(*NeurIPS'20*)	87.285 ± 0.527	-6.983 ± 1.022	87.699 ± 0.870	-6.580 ± 1.709	88.460 ± 0.221	-5.820 ± 0.469
PCGrad(*NeurIPS'20*)	86.880 ± 0.400	-7.437 ± 0.800	87.848 ± 0.317	-6.464 ± 0.632	88.524 ± 0.135	-5.773 ± 0.273
CVweight(*Arxiv'20*)	85.662 ± 0.396	-8.581 ± 0.809	86.285 ± 0.740	-7.971 ± 1.475	87.174 ± 0.099	-7.092 ± 0.261
RLW(*Arxiv'21*)	85.936 ± 0.695	-8.368 ± 1.380	87.019 ± 0.440	-7.284 ± 0.854	87.397 ± 0.264	-6.896 ± 0.569
MaxDO (AGD)	$\mathbf{87.901} \pm 0.244$	-6.468 ± 0.270	$\mathbf{88.566} \pm 0.585$	$-\mathbf{5.776} \pm 0.640$	$\mathbf{88.744} \pm 0.361$	$-\mathbf{5.573} \pm 0.382$

Table 2: Comparisons (avg ± std) with different buffer sizes on PS-CIFAR-100 (100 classes).

Method (+ Rehearsal)	**Buffer size 1000**		**Buffer size 2000**		**Buffer size 3000**	
	A_T (%)	F_T (%)	A_T (%)	F_T (%)	A_T (%)	F_T (%)
MGDA(*NeurIPS'18*)	63.578 ± 0.315	22.866 ± 0.639	67.613 ± 0.166	25.001 ± 0.768	67.704 ± 0.238	24.725 ± 1.075
GradNorm(*ICML'18*)	62.498 ± 0.699	22.506 ± 1.427	63.932 ± 0.679	23.845 ± 1.185	64.538 ± 0.627	24.359 ± 1.450
DWA(*CVPR'18*)	64.952 ± 0.374	23.152 ± 0.487	66.310 ± 0.445	24.697 ± 0.880	66.947 ± 0.156	25.384 ± 0.447
GradDrop(*NeurIPS'20*)	66.371 ± 0.260	23.054 ± 0.633	68.483 ± 0.499	24.962 ± 1.007	69.353 ± 0.707	$\mathbf{26.269} \pm 1.401$
PCGrad(*NeurIPS'20*)	66.724 ± 0.263	$\mathbf{23.601} \pm 0.618$	68.652 ± 0.619	$\mathbf{25.183} \pm 1.081$	68.885 ± 0.134	25.704 ± 0.849
CVweight(*Arxiv'20*)	47.521 ± 2.333	11.868 ± 4.257	48.155 ± 1.682	13.202 ± 3.005	48.424 ± 1.960	13.138 ± 3.573
RLW(*Arxiv'21*)	65.974 ± 0.508	23.080 ± 1.411	68.066 ± 0.276	24.915 ± 0.697	68.162 ± 0.812	24.765 ± 1.078
MaxDO (AGD)	$\mathbf{67.415} \pm 0.803$	22.359 ± 1.028	$\mathbf{69.372} \pm 0.170$	24.523 ± 0.360	$\mathbf{70.078} \pm 0.134$	24.907 ± 0.720

5 tasks and the size of the label set for each task, *i.e.*, the number of classes, is larger than 2 while smaller than 15. (2) *Parallel Split CIFAR-100 (PS-CIFAR-100)*. We split CIFAR-100 into 20 tasks and the size of the label set for each task is larger than 2 while smaller than 15. (3) *Parallel Split ImageNet-TINY (PS-ImageNet-TINY)*. We split Tiny ImageNet (Le & Yang, 2015) (200 classes), which has a training set of 100,000 images and a test set of 10,000 images, into 20 tasks, and the size of the label set for each task is larger than 5 while smaller than 20. See more details in Appendix A.

All three datasets have 3 different label sets (3 different class splits), each of which has 3 different timelines (when to access). For each timeline, we have 3 different runs with fixed seeds 1234, 1235, and 1236 for parameter initialization. In other words, we have 27 different settings for each dataset, and we report the average and standard deviation (avg ± std) for each compared method in our experiments. Note that, we omit all blank time that no data stream flows for simplicity.

4.2 Experiment Details

We implement our experiments using Tensorflow and conduct on a single NVidia RTX 3090Ti GPU card. We take a 2-layer MLP as the backbone network for PS-EMNIST and a Resnet-18 (He et al., 2016) for PS-CIFAR-100 and PS-ImageNet-Tiny. The learning rate is set to 0.003, 0.0004 and 0.0005 for PS-EMNIST, PS-CIFAR-100 and PS-ImageNet-Tiny. The SGD in MaxDO has a learning rate of 5. Each task is trained in a data stream, i.e., each data point passes only once. For each task, we set the batch size to 128 per step. For any new task in PCL, we build a new classifier, which is a fully-connected layer with a softmax function.

To evaluate PCL, we compute the average accuracy and forgetting following previous continual learning studies (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Aljundi et al., 2019b;a; Risheng et al., 2021). Let e_t be the end time of task t and final time $\bar{e} = \max(e_1, e_2, \cdots, e_T)$, the two metrics are computed as follows:

$$A_T = \frac{1}{T} \sum\nolimits_{t=1}^{T} a_{\bar{e}}^{t}, \quad F_T = \frac{1}{T} \sum\nolimits_{t=1}^{T} a_{\bar{e}}^{t} - a_{e_t}^{t}, \tag{9}$$

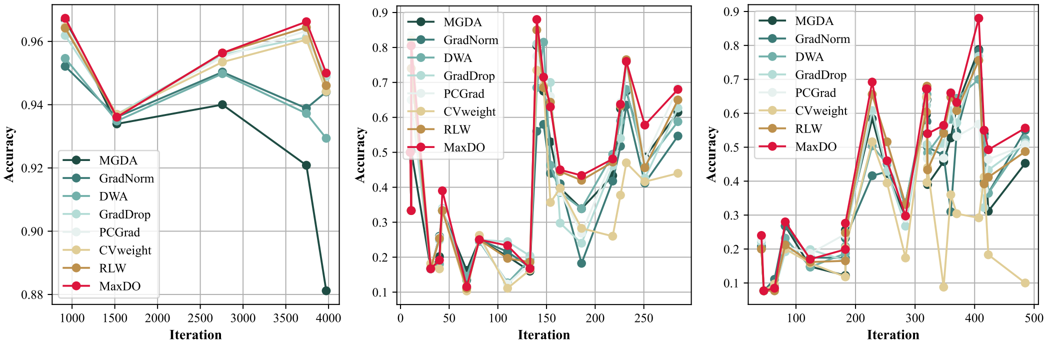
where a_k^j is the mean testing accuracy of task j at time k. The A_T denotes the final average accuracy on all the tasks, and the F_T (also known as backward transfer) means the final performance drop compared to each task that was first trained.

4.3 Main Results

We compare our method with MTL methods including MGDA (Désidéri, 2012), GradNorm (Chen et al., 2018), DWA (Liu et al., 2019), GradDrop (Chen et al., 2020), PCGrad (Yu et al., 2020), CVWeight (Groenendijk et al., 2021) and RLW (Lin et al., 2021) in the PCL setting. We treat any time on the timeline as an MTL subunit to train PCL. All results of previous MTL methods are produced by ourselves with the claimed design in their papers. We show the main comparisons with the proposed methods in Tables 1, 2 and 3 on the three datasets. We have several major observations. First, the rehearsal strategy is useful for reducing catastrophic forgetting in PCL for all compared

Table 3: Comparisons (avg ± std) with different buffer sizes on PS-ImageNet-TINY (200 classes).

Method (+ Rehearsal)	Buffer size 2000		Buffer size 3000		Buffer size 4000	
	A_T (%)	F_T (%)	A_T (%)	F_T (%)	A_T (%)	F_T (%)
MGDA(*NeurIPS'18*)	48.179 ± 0.969	10.936 ± 1.917	51.794 ± 0.427	13.506 ± 0.649	52.644 ± 1.133	13.586 ± 2.191
GradNorm(*ICML'18*)	47.311 ± 1.841	9.975 ± 3.651	49.501 ± 1.882	11.740 ± 3.975	49.331 ± 1.606	11.507 ± 3.530
DWA(*CVPR'18*)	47.429 ± 0.865	10.640 ± 1.764	48.387 ± 0.718	11.505 ± 1.438	47.520 ± 2.129	10.498 ± 4.267
GradDrop(*NeurIPS'20*)	49.955 ± 1.413	11.747 ± 2.961	54.141 ± 0.747	14.633 ± 1.001	53.827 ± 1.146	14.623 ± 1.177
PCGrad(*NeurIPS'20*)	49.052 ± 0.961	10.264 ± 1.406	51.701 ± 0.554	12.508 ± 0.698	50.837 ± 1.097	11.605 ± 1.478
CVweight(*Arxiv'20*)	34.032 ± 0.607	8.221 ± 1.521	36.992 ± 1.900	11.055 ± 3.272	37.007 ± 2.304	9.954 ± 3.830
RLW(*Arxiv'21*)	49.355 ± 0.904	10.857 ± 1.933	49.629 ± 1.454	10.973 ± 3.017	51.947 ± 1.202	12.907 ± 2.478
MaxDO (AGD)	$\mathbf{52.165} \pm 0.694$	$\mathbf{13.287} \pm 0.544$	$\mathbf{54.485} \pm 0.608$	$\mathbf{15.192} \pm 0.444$	$\mathbf{55.192} \pm 0.301$	$\mathbf{15.571} \pm 0.217$



(a) PS-EMNIST (5 tasks) (b) PS-CIFAR100 (20 tasks) (c) PS-ImageNet-TINY (20 tasks)

Figure 4: Task accuracy comparisons along parallel continual learning. Each point means a right finished task and its performance. Note that the order is up to its end time rather than the task ids.

methods. On one hand, as an extra data stream aparts from in-training data streams, rehearsal provides data from the finished tasks training together with other tasks to suppress forgetting. On the other hand, the memory buffer size of rehearsal affects the remembering of old knowledge, and larger size means better knowledge keeping, which is similar to traditional CL. For example in PS-CIFAR-100, we have 67.415%, 69.372% and 70.078% for buffer size 1,000, 2,000, and 3,000, respectively. Second, due to each task in PCL taking the data stream as input, only one pass of each data point is insufficient to make the model converge. With the rehearsal strategy, the memory may provide continual learning of finished tasks, and even better performance can be obtained, which results in positive forgetting value F_T. Third, the compared methods are designed for balanced training and ignore the diverse training process in PCL. Thus, some gradients may be counteracted because of the large gradient discrepancy when updating the model. In contrast, our MaxDO with AGD obtains the best final accuracy A^T on three datasets and different memory buffer sizes, which shows our superiority in balancing plasticity and stability. For example, we have 55.192% for PS-ImageNet-TINY (buffer size 4,000) while the compared best value is only 53.827%. On one hand, the proposed AGD is used to measure the asymmetric distance between gradients to boost the effective update of each task. On the other hand, the maximum discrepancies between multiple tasks are reduced. Note that, the forgetting measure of the proposed methods may not outperform the compared methods because we got both better new tasks (see the following section) and final accuracy performance, their difference value (forgetting) may be small.

4.4 Accuracy Trends

As shown in Fig. 4, we show the accuracy trends of the compared methods on the three evaluated datasets with buffer sizes 310, 3,000 and 4,000 for PS-EMNIST, PS-CIFAR-100 and PS-ImageNet-TINY, respectively. Each point in the figures means a right finished task and its performance then. Note that the task order is up to the end time of tasks rather than the task ids. We have the following observations. Firstly, in the first several tasks, fewer seen tasks mean that fewer discrepancies need to be considered and the compared methods have similar performance. Secondly, when more new tasks are accessed, MaxDO gains better performance for new tasks on three datasets compared to other methods, especially on PS-CIFAR100 and PS-ImageNet-TINY, which both contain 20 tasks. The observations show the proposed MaxDO is useful in PCL for solving diverse training processes. After learning more tasks, MaxDO balances the asymmetric discrepancies among gradients to improve the new task training and old task keeping at the same time. Because MaxDO gets better first accuracy than other methods, the forgetting value may not achieve the best yet is still comparable to others.

Table 4: Metric comparison (↑) and ablation study (↓).

Method (+ Rehearsal)	A_T (%)	F_T (%)
MaxDO (EuDist)	69.344 ± 0.024	24.268 ± 0.748
MaxDO (CosDist)	69.227 ± 0.370	24.552 ± 0.837
MaxDO (Normalized Eudist)	69.540 ± 0.340	24.629 ± 0.158
MGDA (Désidéri, 2012)	67.704 ± 0.238	24.725 ± 1.075
MaxDO (w/o Max-Discrepancy)	68.866 ± 0.443	24.093 ± 0.025
MaxDO (w/o MGDA)	69.953 ± 0.234	$\mathbf{24.933} \pm 0.621$
MaxDO (AGD)	$\mathbf{70.078} \pm 0.134$	24.907 ± 0.720

Table 5: Training time (second/iter).

Method	2 tasks	3 tasks	4 tasks	5 tasks	Total
MGDA	5.58	5.92	6.06	6.64	239
GradNorm	5.30	5.81	5.87	6.50	281
DWA	5.56	5.89	5.94	6.56	245
GradDrop	5.33	6.00	6.34	6.38	275
PCGrad	5.71	5.85	6.11	6.46	229
CVWeight	5.56	5.98	6.00	6.56	227
RLW	5.64	5.89	5.95	6.26	232
MaxDO	5.70	6.12	6.67	6.91	300

4.5 Comparison with Symmetric Metrics

As shown in Table 4, we compare AGD with three common symmetric metrics including EuDist, CosDist, and Normalized EuDist. EuDist, CosDist are defined in Sec. 3.2. The vanilla EuDist depends highly on the gradient magnitude difference, thus we also compare with its normalized version $D(\mathbf{x}, \mathbf{y}) = \frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{x}\|+\|\mathbf{y}\|} \in [0, 1]$, namely normalized EuDist. The results show that the three metrics can also obtain good performance with MaxDO. However, because of the over-emphasizing of gradient magnitude difference in EuDist, it fails to reduce the catastrophic forgetting effectively. Considering only gradient angle difference, MaxDO with CosDist obtains better performance than EuDist. But CosDist ignores the magnitude difference, which is also important in the min-distance problem, resulting in insufficient performance. Compared to EuDist, normalized EuDist obtains better performance but still set symmetric influence to gradient update. In contrast to the three metrics, MaxDO with AGD considers the asymmetric influence on gradient update, and tolerance is set to reduce the influence from small gradients to new-access tasks, which yields the best performance.

4.6 Ablation Study and Procedure Time

We evaluate the impact of the two main components of MaxDO in Table 4. First, we block the maximum discrepancy in MaxDO (MaxDO (w/o Max-Discrepancy)), which means that we solve the min-distance problem with Eq. (6) directly. Because of the self-interference, the solution combines the minimum discrepancy but fails to effectively reduce the discrepancy from other gradients (68.866% for A_T). We then block the MGDA that obtains a weighted gradient not damage any tasks. MGDA is quite useful in traditional MTL tasks but is not suitable in PCL (67.704% for A_T). Because of the diverse training process of parallel tasks, gradients are with large magnitude differences and MGDA prefers to set large factors to small gradients. We solve the problem by both MGDA and the maximum discrepancy, and the whole MaxDO method with AGD outperforms the two ablated methods (70.078% for A_T), where the characters of the two components are combined.

In Table 5, we show the training time comparison on PS-CIFAR-100. We first compare the training time for 2 to 5 parallel tasks in one iteration. We find that the generation of task numbers will grow the training time, and MaxDO needs more time than other methods because multiple minimum distance optimizations are performed. Thus, in the whole timeline, MaxDO gets slightly longer training time than other methods. It is interesting to explore how to speed up the MaxDO training in the future.

5 Conclusion

In this paper, we studied to address the training conflict and catastrophic forgetting issues in Parallel Continual Learning (PCL). We presented that the two issues are rooted in the gradient discrepancies and formulated the problem into a minimum distance optimization among gradients. However, the distance metric is often set to be symmetric, which is problematic in gradient descent. To evaluate the gradient discrepancy in PCL, we proposed an explicit Asymmetric Gradient Distance (AGD), which considers both gradient magnitude ratios and directions and has a tolerance when updating with a small gradient of inverse direction. Moreover, we also proposed a novel Maximum Discrepancy Optimization (MaxDO) strategy to minimize the maximum discrepancy among multiple gradients and avoid self-interference. Solving by MaxDO with AGD, the parallel training in PCL reduces the influence of the training conflict and slows the catastrophic forgetting of finished tasks. We verified the proposed benchmark on three image recognition datasets. The experimental results significantly showed the advantage of our MaxDO and the effectiveness of the proposed AGD. We list the latent limitation of our method: (1) The MaxDO cannot guarantee a theoretical Pareto optimum in the training process like MGDA, which means a better trade-off can be obtained in the future. (2) The MaxDO method needs more time for training.

REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *NeurIPS*, 2019a.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019b.

Craig Atkinson, Brendan McCane, Lech Szymanski, and Anthony Robins. Pseudo-rehearsal: Achieving deep reinforcement learning without catastrophic forgetting. *arXiv preprint arXiv:1812.02464*, 2018.

Aviad Barzilai and Koby Crammer. Convex multi-task learning by clustering. In *AISTATS*, 2015.

James M Buchanan. The relevance of pareto optimality. *Journal of conflict resolution*, 1962.

Francesca Cagliari, Barbara Di Fabio, and Claudia Landi. The natural pseudo-distance as a quotient pseudo-metric, and applications. In *Forum Mathematicum*. De Gruyter, 2015.

Rich Caruana. Multitask learning. *Machine learning*, 1997.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018.

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *NeurIPS*, 2020.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *IJCNN*, 2017.

Julia Collins and Johannes Zimmer. An asymmetric arzelà–ascoli theorem. *Topology and its Applications*, 2007.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 2012.

Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, 2019.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *AISTATS*, 2020.

MA Fiol. On pseudo-distance-regularity. *Linear Algebra and its Applications*, 2001.

Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 2000.

Robert M French. Catastrophic forgetting in connectionist networks. *TiCS*, 1999.

Rick Groenendijk, Sezer Karaoglu, Theo Gevers, and Thomas Mensink. Multi-loss weighting with coefficient of variations. In *WACV*, 2021.

Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. Learning with long-term remembering: Following the lead of mixed stochastic gradient. *arXiv preprint arXiv:1909.11763*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep Ravikumar. A dirty model for multi-task learning. In *NeurIPS*, 2010.

Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. *arXiv preprint arXiv:2103.02631*, 2021.

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. *PNAS*, 2017.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 2017.

Baijiong Lin, Feiyang Ye, and Yu Zhang. A closer look at loss weighting in multi-task learning. *arXiv preprint arXiv:2111.10603*, 2021.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, 2021.

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, 2019.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, 2018.

Andreas Maurer, Massi Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. In *ICML*, 2013.

Andrea CG Mennucci. On asymmetric distances. *Analysis and Geometry in Metric Spaces*, 2013.

Jary Pomponi, Simone Scardapane, and Aurelio Uncini. Pseudo-rehearsal for continual learning with normalizing flows. *arXiv preprint arXiv:2007.02443*, 2020.

Liu Risheng, Liu Yaohua, Zeng Shangzhi, and Jin Zhang. Gradient-based editing of memory examples for online task-free continual learning. In *NeurIPS*, 2021.

Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *TPAMI*, 2018.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *NeurIPS*, 2018.

Haseeb Shah, Khurram Javed, and Faisal Shafait. Distillation techniques for pseudo-rehearsal based incremental learning. *arXiv preprint arXiv:1807.02799*, 2018.

Sebastian Thrun and Joseph O'Sullivan. Discovering structure in multiple learning tasks: The tc algorithm. In *ICML*, 1996.

Jie Wang and Jieping Ye. Safe screening for multi-task feature learning with multiple data matrices. In *ICML*, 2015.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*, 2020.

Wallace Alvin Wilson. On quasi-metric spaces. *American Journal of Mathematics*, 1931.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE TKDE*, 2021.

Measuring Asymmetric Gradient Discrepancy in Parallel Continual Learning (Appendix)

A Dataset Construction

For effective transformation, several requirements are needed: (1) Random label set for each task, in which the data stream length of each task can be different; (2) Random timeline for each label set, in which the debut of each task can be any time between the first access of the former and latter tasks. For simplicity, we omit all blank time that all data streams are unavailable.

- *Parallel Split EMNIST* (PS-EMNIST): We split EMNIST (62 classes) into 5 tasks and randomly generate 3 label sets for each task and 3 timelines for each label set (say 9 different situations). The size of the label set for each task, i.e., the number of classes, is set to larger than 2 while no more than 15.
- *Parallel Split CIFAR-100* (PS-CIFAR-100): We split CIFAR-100 into 20 tasks and randomly generate 3 label sets for each task and 3 timelines for each label set. The size of the label set for each task is set to larger than 2 while no more than 15.
- *Parallel Split ImageNet-TINY* (PS-ImageNet-TINY): We split it into 20 tasks w.r.t. random 3 label sets, and each label set has 3 randomly generated timelines. The size of the label set for each task is set to larger than 5 while no more than 20.

B Proof of Lemma 1 on AGD

As an asymmetric metric, the proposed Asymmetric Gradient Discrepancy (AGD) measure needs to satisfy the two features in Lemma 1.

Proof: Given three arbitrary gradients $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, we have

(1) If $\mathbf{x} = \mathbf{y}$, $D(\mathbf{x}, \mathbf{y}) = 0$.

(2) **Positivity**: If $\mathbf{x} \neq \mathbf{y}$, then $\|\mathbf{x} - \mathbf{y}\| \neq 0$, and we have $D(\mathbf{x}, \mathbf{y}) = \dfrac{\|\mathbf{x} - \mathbf{y}\|}{\|\mathbf{y}\| + \|\mathbf{x} - \mathbf{y}\|} > 0$.

(3) **The triangle inequality**:

$$\begin{aligned}
&\frac{\|\mathbf{x}-\mathbf{z}\|}{\|\mathbf{z}\|+\|\mathbf{x}-\mathbf{z}\|} = \frac{\|\mathbf{x}-\mathbf{y}+\mathbf{y}-\mathbf{z}\|}{\|\mathbf{z}\|+\|\mathbf{x}-\mathbf{y}+\mathbf{y}-\mathbf{z}\|} \\
\leq\ &\frac{\|\mathbf{x}-\mathbf{y}\|+\|\mathbf{y}-\mathbf{z}\|}{\|\mathbf{z}\|+\|\mathbf{x}-\mathbf{y}\|+\|\mathbf{y}-\mathbf{z}\|} \\
=\ &\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{z}\|+\|\mathbf{x}-\mathbf{y}\|+\|\mathbf{y}-\mathbf{z}\|} + \frac{\|\mathbf{y}-\mathbf{z}\|}{\|\mathbf{z}\|+\|\mathbf{x}-\mathbf{y}\|+\|\mathbf{y}-\mathbf{z}\|} \\
\leq\ &\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{z}\|+\|\mathbf{x}-\mathbf{y}\|+\|\mathbf{y}-\mathbf{z}\|} + \frac{\|\mathbf{y}-\mathbf{z}\|}{\|\mathbf{z}\|+\|\mathbf{y}-\mathbf{z}\|} \\
\leq\ &\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|+\|\mathbf{x}-\mathbf{y}\|} + \frac{\|\mathbf{y}-\mathbf{z}\|}{\|\mathbf{z}\|+\|\mathbf{y}-\mathbf{z}\|}.
\end{aligned} \tag{10}$$

(4) **Asymmetric**: $D(\mathbf{x}, \mathbf{y}) = \dfrac{\|\mathbf{x} - \mathbf{y}\|}{\|\mathbf{y}\| + \|\mathbf{x} - \mathbf{y}\|}$, and $D(\mathbf{y}, \mathbf{x}) = \dfrac{\|\mathbf{x} - \mathbf{y}\|}{\|\mathbf{x}\| + \|\mathbf{x} - \mathbf{y}\|}$. Thus, it is obvious that $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$ is not always satisfied when $\mathbf{x} \neq \mathbf{y}$ and depends on the magnitude $\|\mathbf{x}\|$ and $\|\mathbf{y}\|$.

Therefore, the proposed AGD is an asymmetric metric. ■

C Tolerance analysis and Proof of Corollary 1

C.1 Proof of Corollary 1

Let us review the definition of AGD:

$$\widehat{D}(\mathbf{x}, \mathbf{y}) = \frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\| + \|\mathbf{x}-\mathbf{y}\|}. \tag{11}$$

$\widehat{D}(\mathbf{x}, \mathbf{y})$ represents the gradient influence from $\mathbf{x}$ to $\mathbf{y}$. The nature of this asymmetric measure is the norm effect should *only* be from gradient difference $\|\mathbf{x}-\mathbf{y}\|$ to $\|\mathbf{y}\|$ rather than to both $\|\mathbf{x}\|$ and $\|\mathbf{y}\|$. That is, the discrepancy should only depend on the ratio $\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|}$, which can be further reduced to

$$\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|} = \frac{\sqrt{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\|\mathbf{x}\|\|\mathbf{y}\| \cos\angle\mathbf{x}, \mathbf{y}}}{\|\mathbf{y}\|} = \sqrt{\left(\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\right)^2 - 2\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\cos\angle\mathbf{x}, \mathbf{y} + 1}. \tag{12}$$

It is easy to know that

$$\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\| + \|\mathbf{x}-\mathbf{y}\|} = 1 - \frac{1}{1 + \frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|}}. \tag{13}$$

Because $\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|} \geq 0$, $\widehat{D}(\mathbf{x}, \mathbf{y}) \in [0, 1]$.

In the paper, we illustrate the proposed AGD is an asymmetric measure of gradient discrepancy because $\widehat{D}(\mathbf{x}, \mathbf{y})$ brings a tolerance when $\|\mathbf{x}\| \ll \|\mathbf{y}\|$ instead of the absolute difference between them. To analyze the values of gradient discrepancy measure D regarding $\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}$, we consider the following asymmetric limits with $\|y\| \neq 0$:

- $\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty} D$: When $\|\mathbf{x}\| \gg \|\mathbf{y}\|$, the conflict should be large from $\mathbf{x}$ to $\mathbf{y}$;
- $\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0} D$: When $\|\mathbf{x}\| \ll \|\mathbf{y}\|$, the conflict is acceptable to some extend and should approach a tolerance value that less than $\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty} D$.

We show the two limits for different discrepancy measures including Cosine Similarity, Euclidean Distance, Normalized Euclidean Distance, and AGD.

Cosine Similarity: Using the Cosine Similarity to measure the discrepancy has no relevance to the magnitude difference.

$$\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0} 1 - \frac{\mathbf{x}^\top\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty} 1 - \frac{\mathbf{x}^\top\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = 1 - \cos\angle\mathbf{x}, \mathbf{y}. \tag{14}$$

Euclidean Distance: When $\|\mathbf{y}\| \neq 0$, we have

$$\frac{1}{1 + \|\mathbf{x}-\mathbf{y}\|} = \frac{1}{1 + \|\mathbf{y}\| \cdot \frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|}}. \tag{15}$$

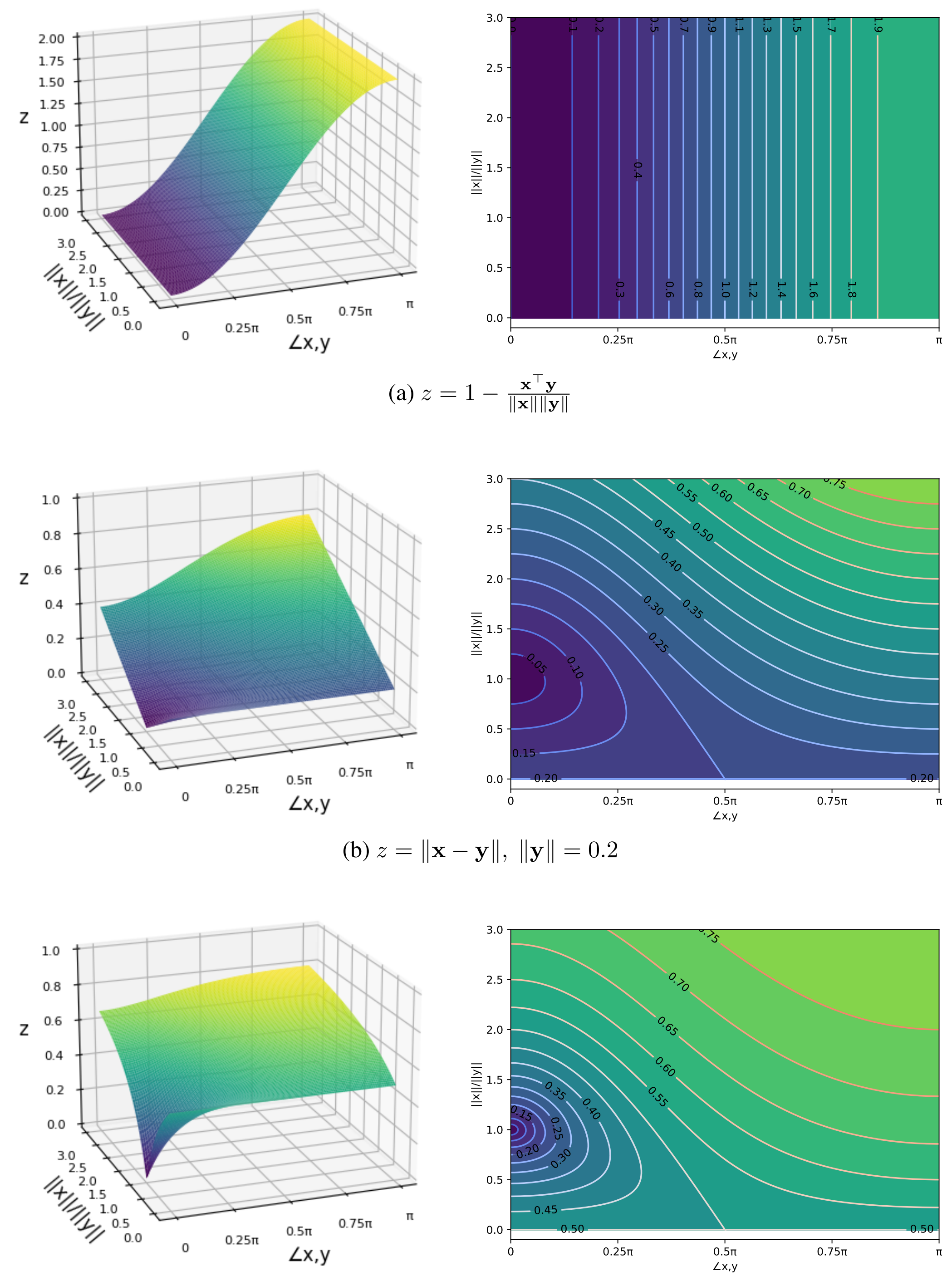
Thus, we have

$$\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0} 1 - \frac{1}{1 + \|\mathbf{x}-\mathbf{y}\|} = \frac{\|\mathbf{y}\|}{1 + \|\mathbf{y}\|}, \quad \lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty} 1 - \frac{1}{1 + \|\mathbf{x}-\mathbf{y}\|} = 1. \tag{16}$$

When $\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|} \to 0$, by using the Euclidean Distance highly depends on $\|\mathbf{y}\|$, which makes it unpredictable.

Normalized Euclidean Distance: When $\|\mathbf{y}\| \neq 0$, we have

$$\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0} \frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{x}\| + \|\mathbf{y}\|} = \lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0} \frac{\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|}}{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|} + 1} = 1, \tag{17}$$



(a) $z = 1 - \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$

(b) $z = \|\mathbf{x}-\mathbf{y}\|,\ \|\mathbf{y}\| = 0.2$

(c) $z = \frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|+\|\mathbf{x}-\mathbf{y}\|}$

Figure 5: Contours of different measures. Note that the x- and y-axes are the angle (i.e., $\angle\mathbf{x},\mathbf{y}$) between $\mathbf{x}$ and $\mathbf{y}$, and the magnitude ratio $\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}$, respectively. (a) Cosine distance; (b) Euclidean distance where $\|\mathbf{y}\| = 0.2$; (c) Asymmetric gradient distance.

$$\begin{aligned}
\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty} \frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{x}\|+\|\mathbf{y}\|} &= \lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty} \frac{\sqrt{\left(\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\right)^2 - 2\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\cos\angle\mathbf{x},\mathbf{y}+1}}{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}+1} \\
&= \lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty} \sqrt{\frac{2\cos\angle\mathbf{x},\mathbf{y}+2}{\left(\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}+1\right)^2} - \frac{2\cos\angle\mathbf{x},\mathbf{y}+2}{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}+1}+1} \\
&=1.
\end{aligned} \tag{18}$$

Table 6: Comparisons between different tolerances of AGD on PS-CIFAR-100.

γ	Tolerance	A_T (%)	F_T (%)
0.2	5/6	69.283 ± 0.307	24.126 ± 0.472
0.5	2/3	69.486 ± 0.204	24.520 ± 0.570
1 (ours)	1/2	70.078 ± 0.134	24.907 ± 0.720
2	1/3	69.626 ± 0.192	24.344 ± 0.610
3	1/4	69.505 ± 0.442	24.479 ± 0.408
4	1/5	69.332 ± 0.142	24.600 ± 0.404

The discrepancy using Normalized EuDist has the same value when $\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0}$ and $\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty}$, which means no tolerance.

AGD and **Proof of Corollary 1**: According to Eq. (13), we have

$$\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0} \widehat{D}(\mathbf{x},\mathbf{y}) = \lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0} 1 - \frac{1}{1+\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|}} = \frac{1}{2}, \tag{19}$$

$$\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to \infty} \widehat{D}(\mathbf{x},\mathbf{y}) = \lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to \infty} 1 - \frac{1}{1+\frac{\|\mathbf{x}-\mathbf{y}\|}{\|\mathbf{y}\|}} = 1. \tag{20}$$

The two equations denote that when $\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|} \to 0$, AGD has the tolerance value $\frac{1}{2} < \lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to\infty} = 1$, which means that $\|\mathbf{x}\| << \|\mathbf{y}\|$ is acceptable as the half of perfect equal.

C.2 Different tolerance analysis

In our paper, we propose an Asymmetric Gradient Distance (AGD) to evaluate the gradient discrepancy. AGD is designed to have a tolerance $\frac{1}{2}$ in Corollary 1. This is because updating with a zero gradient will neither improve nor damage the performance. Even though, we prefer positive influence rather than non-influence. Thus, we define that the distance $\widehat{D}(\mathbf{g}_{\mathrm{A}}, \mathbf{g}_{\mathrm{B}})$ in the situation $\|\mathbf{g}_{\mathrm{A}}\| \ll \|\mathbf{g}_{\mathrm{B}}\|$ is the mid-level in the value range.

In this subsection, we try to change the tolerance and observe the performance change. The tolerance can be controlled by adding a factor $\gamma > 0$. Omitting the edge situation, we have

$$\widehat{D}(g_{\mathrm{A}}, g_{\mathrm{B}}) = \frac{\|g_{\mathrm{A}} - g_{\mathrm{B}}\|}{\gamma\|g_{\mathrm{B}}\| + \|g_{\mathrm{A}} - g_{\mathrm{B}}\|}.$$

The experiments on different tolerances are shown in Table 6. The results show either larger or smaller tolerances compared to $\frac{1}{2}$ will get the performance drop.

D Contour of AGD

We show more function contour comparisons with existing measurement methods in Fig. 5, where the axes are the angle $\angle\mathbf{x},\mathbf{y}$, the ratio $\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}$ and the metric contour value z for better visualization. As we can see, the CosDist (Fig. 5(a)) has no relation to the ratio. The tolerance for $\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0}$ of EuDist depends on the norm of $\mathbf{y}$ (Fig. 5(b)). The proposed AGD has fixed tolerance for $\lim_{\frac{\|\mathbf{x}\|}{\|\mathbf{y}\|}\to 0}$ as shown in Fig. 5(c).

E Introduction of MGDA

At any time, PCL training yields the following dynamic multi-objective empirical risk minimization formulation:

$$\min_{\boldsymbol{\theta},\{\boldsymbol{\theta}_i|i\in\mathcal{T}\}} \quad \{\ell_i\left(\mathcal{D}_i\right)|i\in\mathcal{T}_t\},$$

where $\mathcal{T}$ is the task index set with activated data streams at time t.

Table 7: Comparisons between using AGD with and without rehearsal gradient on PS-CIFAR-100.

Method (+ Rehearsal)	A_T (%)	F_T (%)
MaxDO (w/o rehearsal gradient)	68.398 ± 0.776	23.676 ± 0.583
MaxDO (w/ rehearsal gradient)	70.078 ± 0.134	24.907 ± 0.720

An elegant solution to the MOO for Pareto optimality Buchanan (1962) is the Steepest Descent Method (SDM) Fliege & Svaiter (2000), which aims to obtain an optimal descent direction d^* that satisfies

$$d^*, \alpha^* = \arg\min \_d, \alpha \quad \alpha + \frac{1}{2}\left\|d\right\|^2, \text{s.t.} \quad g\_i^\top d \leq \alpha, \quad \forall i \in \mathcal{T},$$

where the constraints let each task have non-conflict with gradient d. Considering the Lagrange multipliers and Karush–Kuhn–Tucker (KKT) condition, the dual problem solved by the Multi-Gradient Descent Algorithm (MGDA) Désidéri (2012) is

$$w^* = \arg\min_{\mathbf{w}} \quad \left\| \sum_i \mathbf{w}_i \mathbf{g}_i \right\|^2, \text{s.t.} \quad \sum_i \mathbf{w}_i = 1 \text{ and } \mathbf{w}_i \geq 0, \forall i.$$

The objective of MGDA is 0 and the resulting point satisfies the KKT conditions, or the solution gives a Pareto descent direction that improves all tasks.

F MaxDO effects on rehearsal

In rehearsal-based PCL, the training conflict may worsen the forgetting of old tasks. That is, the new task produces large gradients and may mislead the replay of old tasks with small gradients. In our method, we consider reducing this gradient conflict and propose to measure the asymmetric gradient distance. Moreover, we propose to minimize the maximum discrepancy among multiple gradients.

To show the MaxDO's effectiveness of forgetting reduction on rehearsal gradient, we evaluate the result that only leverages MaxDO on new tasks instead memory data stream (*i.e.*, finished tasks). The result is shown in Fig. 7. In this case, the final gradient is calculated by $\mathbf{d} = \frac{1}{2}\mathbf{g}_{\text{reherasal}} + \frac{1}{2}\mathbf{g}_{\text{new}}$, where $\mathbf{g}_{\text{new}}$ is the solution gradient via MaxDO on only new tasks. The result shows that it is necessary to put the rehearsal gradient to the MaxDO. Otherwise, the model will get worse accuracy and weak forgetting.