Direction Concentration Learning: Enhancing Congruency in Machine Learning

Yan Luo, Yongkang Wong, *Member, IEEE*,
Mohan Kankanhalli, *Fellow, IEEE*, and Qi Zhao, *Member, IEEE*

Abstract—One of the well-known challenges in computer vision tasks is the visual diversity of images, which could result in an agreement or disagreement between the learned knowledge and the visual content exhibited by the current observation. In this work, we first define such an agreement in a concepts learning process as congruency. Formally, given a particular task and sufficiently large dataset, the congruency issue occurs in the learning process whereby the task-specific semantics in the training data are highly varying. We propose a Direction Concentration Learning (DCL) method to improve congruency in the learning process, where enhancing congruency influences the convergence path to be less circuitous. The experimental results show that the proposed DCL method generalizes to state-of-the-art models and optimizers, as well as improves the performances of saliency prediction task, continual learning task, and classification task. Moreover, it helps mitigate the catastrophic forgetting problem in the continual learning task. The code is publicly available at https://github.com/luoyan407/congruency.

Index Terms—Optimization, machine learning, computer vision, accumulated gradient, congruency

1 Introduction

Deep learning has been receiving considerable attention due to its success in various computer vision tasks [4], [13], [16], [27] and challenges [6], [32]. To prevent model overfitting and enhance the generalization ability, a training process often sequentially updates the model with gradients w.r.t. a mini-batch of training samples, as opposed to using a larger batch [12]. Due to the complexity and diversity in the nature of image data and task-specific semantics, the discrepancy between current and previous observed mini-batches could result in a circuitous convergence path, which possibly hinders the convergence to a local minimum.

To better understand the circuitousness/straightforwardness in a learning process, we introduce *congruency* to quantify the agreement between new information used for an update and the knowledge learned from previous iterations. The word "congruency" is borrowed from a psychology study [51] that inspects the influence of an object which is inconsistent with the scene in the visual attention perception task. In this work, we define congruency ν as the cosine similarity between the gradient g to be used for update and a referential gradient $\hat{g}$ that indicates a general descent direction resulting from previous updates, i.e.,

$$\nu = \cos\alpha(g, \hat{g}), \tag{1}$$

(The detailed formulation is presented in Section 3). Fig. 1 presents an illustration of congruency in the saliency prediction task. Due to similar scene (i.e., dining) and similar fixations on faces and foods, the update of sample S_2 (i.e., Δw_{S_2}) is congruent with Δw_{S_1}. In contrast, the scene and fixations in sample S_3 are different from sample S_1 and S_2. This leads to a large angle ($> 90°$) between Δw_{S_3} and Δw_{S_2} (or Δw_{S_1}).

Congruency reflects the diversity of task-specific semantics in training samples (i.e., images and the corresponding ground-truths). In the visual attention task, attention is explained by various hypotheses [2], [3], [50] and can be affected by many factors, such as bottom-up feature, top-down feature guidance, scene structure, and meaning [55]. As a result, objects in the same category may exhibit disagreements with each other in various images in terms of attracting attention. Therefore, there is a high variability in the mapping between visual appearance and the corresponding fixations. Another task that has a considerable amount of diversity is continual learning, which is able to learn continually from a stream of data that is related to new concepts (i.e., unseen labels) [33]. The diversity of the data among multiple classification subtasks may be so much discrepant such that learning from new data violates previously established knowledge (i.e., catastrophic forgetting) in the learning process. Moreover, congruency can also be found in the classification task. Compared to saliency prediction and continual learning, the source of diversity in classification task is relatively simple, namely, diverse visual appearances w.r.t. various labels in the real-world images. In summary, saliency prediction, continual learning, and classification are challenging scenarios susceptible to the effects of congruency.

In machine learning, congruency can be considered as a factor that influences the convergence of optimization

- *Y. Luo and Q. Zhao are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455. E-mail: luoxx648@umn.edu, qzhao@cs.umn.edu.*
- *Y. Wong and M. Kankanhalli are with the School of Computing, National University of Singapore, Singapore 117417. E-mail: yongkang.wong@nus.edu.sg, mohan@comp.nus.edu.sg.*

Manuscript received 16 Aug. 2019; revised 16 Dec. 2019; accepted 21 Dec. 2019. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Qi Zhao.)
Recommended for acceptance by H. Ling.
Digital Object Identifier no. 10.1109/TPAMI.2019.2963387

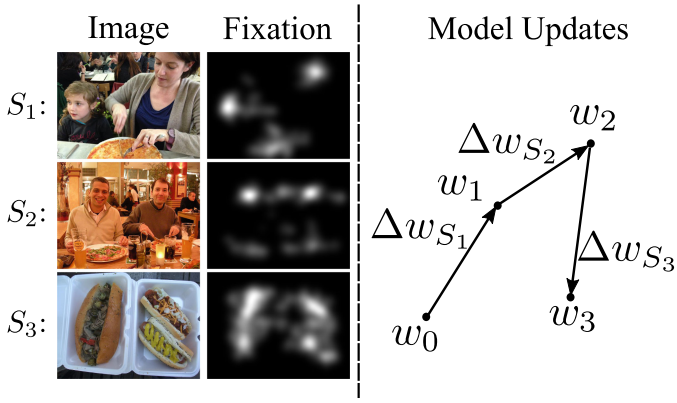


Fig. 1. An illustration of congruency in the saliency prediction task. Assuming training samples are provided in a sequential manner, an incongruency occurs since the food item is related to different saliency values across these samples. Here, S_j stands for sample $j = \{1, 2, 3\}$, w_i is the weight at time step i, Δw_{S_j} is the weight update generated with S_j for w_i, and the arrows indicate updates for the model. Specifically, $\Delta w_{S_j} = -\eta g_{S_j}$ where η is the learning rate and g_{S_j} is the gradient w.r.t. S_j. The update of S_2 (i.e., Δw_{S_2}) is congruent with Δw_{S_1}, whereas Δw_{S_3} is incongruent with Δw_{S_1} and Δw_{S_2}.

methods, such as stochastic gradient descent (SGD) [42], RMSProp [14], or Adam [23]. Without specific rectification, the diversity among training samples is implicitly and passively involved in a learning process and affects the descent direction in convergence. To understand the effects of congruency on convergence, we explicitly formulate a direction concentration learning (DCL) method by sensing and restricting the angle of deviation between an update gradient and a referential gradient that indicates the descent direction according to the previous updates. Inspired by Nesterov's accelerated gradient [37], we consider the accumulated gradient as the referential gradient in the proposed DCL method.

We comprehensively evaluate the proposed DCL method with various models and optimizers in saliency prediction, continual learning, and classification tasks. The experimental results show that the constraints restricting the angle deviation between the gradient for an update and the accumulated referential gradient can help the learning process to converge efficiently, comparing to the approaches without such constraints. Furthermore, we present the congruency patterns to show how the task-specific semantics affect congruency in a learning process. Last but not least, our analysis shows that enhancing congruency in continual learning can improve backward transfer.

The main contributions in this work are as follows:

- We define *congruency* to quantify the agreement between new information and the learned knowledge in a learning process, which is useful to understand the model convergence in terms of tractability.
- We propose a direction concentration learning (DCL) method to enhance congruency so that the disagreement between new information and the learned knowledge can be alleviated. It also generally adapts to various optimizers (e.g., SGD, RMSProp and Adam) and various tasks (e.g., saliency prediction, continual learning and classification).
- The experimental results from continual learning task demonstrate that enhancing congruency can improve backward transfer. Note that large negative backward transfer is known as catastrophic forgetting [33].
- A general method analyzing congruency is presented and it can be used within both conventional models and models with the proposed DCL method. Comprehensive analyses w.r.t saliency prediction and classification show that our DCL method generally enhances the congruencies of the corresponding learning processes.

The rest of the paper is organized as follows. We begin by highlighting related works in Section 2. Then, we formulate the problem of congruency and discuss its factors in Section 3. The proposed DCL method is introduced in Section 4. Moreover, the experiments and analyses are provided in Sections 5 and 6, respectively. Section 7 concludes the paper.

2 Related Works

2.1 State-of-the-Art Models for Classification

Convolutional networks (ConvNets) [13], [16], [27], [56] have exhibited their powers in the classification task. AlexNet [27] is a typical ConvNet and consists of a series of convolutional, pooling, activation, and fully-connected layers, it achieves the best performance on ILSVRC 2012 [6]. Since then, there are more and more attempts to delve into the architecture of ConvNets. He *et al.* proposed residual blocks to solve the vanishing gradient problem and the resulting model, i.e., ResNet [13], achieves best performance on ILSVRC 2015. Along with a similar line of ResNet, ResNeXt [56] is proposed to extend residual blocks to multi-branch architecture and DenseNet [16] is devised to establish the connections between each layer and later layers in a feed-forward fashion. Both models achieve desirable performance. Recently, Tan and Le [47] study how network depth, width, and resolution influence the classification performance and propose EfficientNet that achieves state-of-the-art performance on ImageNet. In this work, we use ResNet, ResNeXt, DenseNet, and EfficientNet in the image classification experiments.

Yang *et al.* [60] introduce a regularized feature selection framework for multi-task classification. Specifically, the trace norm of a low rank matrix is used in the objective function to share common knowledge across multiple classification tasks. Congruency generally works with gradient based optimization methods, whereas trace norm works with a specific optimization method. Moreover, congruency measures the agreement (or disagreement) between new information learned from a sample and the established knowledge, whereas trace norm is based on the weights of multiple classifiers and only measures the correlation between established knowledge w.r.t. different classification tasks.

2.2 Computational Modelling of Visual Attention

Saliency prediction is an attentional mechanism that focuses limited perceptual and cognitive resources on the most pertinent subset of the available sensory data. Itti *et al.* [19] implement the first computational model to predict saliency maps by integrating bottom-up features. Recently, Huang *et al.* [17] propose a data-driven DNN model, named SALICON, to

model visual attention. Cornia *et al.* [5] propose a convolutional LSTM to iteratively refine the predictions and Kummerer *et al.* [28] design a readout network that is fed with the output features of VGG [46] to improve saliency prediction. Yang *et al.* [59] introduce an end-to-end Dilated Inception Network (DINet) to capture multi-scale contextual features for saliency prediction and achieves state-of-the-art performance. In this work, we adopt the SALICON model and DINet in the saliency prediction experiments.

There are several insightful works [11], [51], [52] exploring the effects of congruency/incongruency in visual attention. In particular, according to the perception experiments, Gordon finds that the object which is inconsistent with the scene, e.g., a live chicken standing on a kitchen table, has significant influence on attentive allocation [11]. Underwood and Foulsham [51] find an unexpected interaction between saliency and negative congruency in the search task, that is, the congruency of the conspicuous object does not influence the delay in its fixation, but it is fixated earlier when the other object in the scene is incongruent. Furthermore, Underwood *et al.* [52] investigate whether the effects of semantic inconsistency appear in free viewing. In their studies, inconsistent objects were fixated for significantly longer duration than consistent objects. These works inspire us to explore the congruency between the current and previous updates. In saliency prediction, negative congruency may result from the disagreement among the training samples in terms of visual appearance and ground-truth.

2.3 Catastrophic Forgetting

Catastrophic forgetting problem has been extensively studied in [9], [10], [35], [39]. McCloskey and Cohen [35] study the problem that new learning may interfere catastrophically with old learning when models are trained sequentially. New learning may alter weights that are involved in representing old learning, and this may lead to catastrophic interference. Along the same line, Ratcliff [39] further investigates the causes of catastrophic forgetting, and two problems are observed: 1) sequential learning is prone to rapidly forget well-learned information as new information is learned; 2) discrimination between observed samples and unobserved samples either decreases or is non-monotonic as a function of learning. To address the catastrophic forgetting problem, there are several works [24], [33], [40] proposed to solve the problem by using episodic memory. Kirkpatrick *et al.* [24] propose an algorithm named elastic weight consolidation (EWC), which can adjust learning to minimize changes in parameters important for previously seen task. Moreover, Lopez and Ranzato [33] introduce the gradient episodic memory (GEM) method to alleviate catastrophic forgetting problem. However, there could exist incongruency in the training process of GEM.

3 Congruency in Machine Learning

3.1 Problem Statement

We first review the general goal in machine learning. Without loss of generality, given a training set $D = \{(I_i, y_i)\}_{i=1}^{n}$, where a pair (I_i, y_i) represents a training sample composed of an image $I_i \in \mathbb{R}^{N_I}$ (N_I is the dimension of images) and the corresponding ground-truth $y_i \in \mathcal{Y}$, the goal is to learn a model $f : \mathbb{R}^{N_I} \underset{\rightarrow}{} Y$. Specifically, a Deep Neural Network (DNN) model has a trunk net to generate discriminative features $x_i \in \mathcal{X}$ and a classifier $f_w : \mathcal{X} \xrightarrow{w} \mathcal{Y}$ to fulfill the task, where w is the weights of classifier. Note that we consider that DNN is a classifier as whole and the input is raw RGB images.

To accomplish the learning process, the conventional approach is to first specify and initialize a model. Next, the empirical risk minimization (ERM) principle [53] is employed to find a desirable w w.r.t. f by minimizing a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$ penalizing prediction errors, i.e., $\text{minimize}_w \ \frac{1}{|D|} \sum_{(x_i, y_i) \in D} \ell(f_w(x_i), y_i)$. At time step k, the gradient computed by the loss is used to update the model, i.e., $w_{k+1} := w_k + \Delta w_k$, where Δw_k is an update as well as a function of gradient $g(w_k; x_k, y_k) = \nabla_{w_k} \ell(f_{w_k}(x_k), y_k)$. Optimizers, such as SGD [42], RMSProp [14], or Adam [23], determine $\Delta w_k(g(w_k; x_k, y_k))$. Without loss of generality, we assume the optimizer is SGD in the following for convenience.

There exist two challenges w.r.t. congruency for practical use. First, due to the dynamic nature of the learning process, how to find a stable referential direction which can quantify the agreement between current and previous updates. Second, how to guarantee the referential direction is beneficial to search for a local minimum.

As the gradient at a training step implies the direction towards a local minimum by the currently observed mini-batch, the accumulation of all previous gradients provides an overall direction towards a local minimum. Hence, it provides a good referential direction to measure the agreement between a specific update and its previous updates. We denote the accumulated gradient as

$$\hat{g}_{k|w_m} = \sum_{i=m}^{k} g_i, \tag{2}$$

where w_m is the weights learned at time step m and $\hat{g}_{k|w_m}$ indicates that the accumulation starts from w_m at time step k. If there is no explicit w_m indicated, $\hat{g}_k = \hat{g}_{k|w_1}$. Fig. 2 shows an example of accumulated gradient, where the gradient of S_3 deviates from the accumulated gradient of S_1 and S_2. This also elicits our solution to measure congruency in a training process.

3.2 Definition

Congruency ν is a metric to measure the agreement of updates in a training process. In general, it is directly related to an angle between the gradient for an update and the accumulated gradient, i.e., $\alpha(\hat{g}_{k-1|w_m}, g_k) \in [0, \pi]$. Smaller angle indicates higher congruency. Practically, we use cosine similarity to approximate the angle for computational simplicity. Mathematically, at time step k, ν_k can be defined as follows:

$$\nu_{k|w_m} = \cos\alpha(g_k, \hat{g}_{k-1|w_m}) = \frac{\hat{g}_{k-1|w_m}^{\top} g_k}{\|\hat{g}_{k-1|w_m}\| \|g_k\|}, \ m \leq k, \tag{3}$$

where w_m is the weight learned at time step m and taken as a reference point in weight space. $\alpha(\hat{g}_k, g_k)$ is the angle between $\hat{g}_k$ and g_k. Based on $\nu_{k-1|w_m}$, the congruency of a

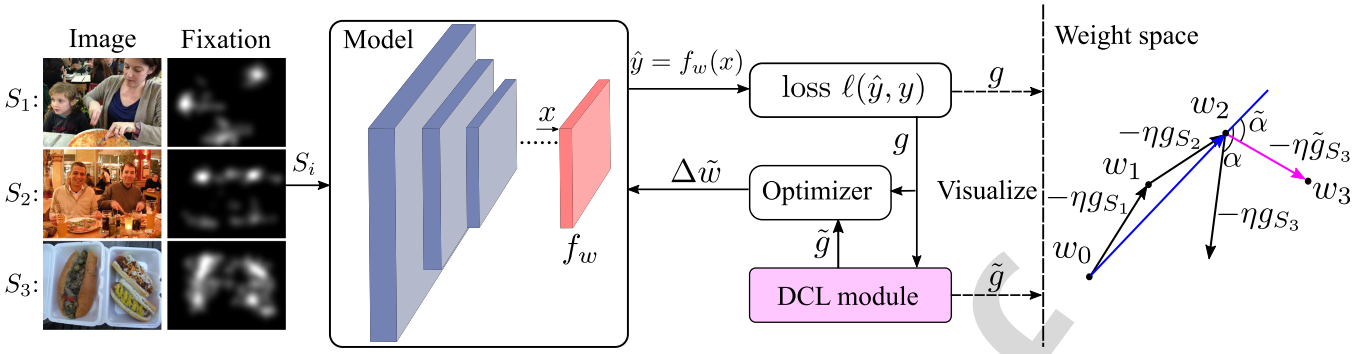


Fig. 2. An illustration of model training with the proposed DCL module. Here, 3 samples are observed in a sequential manner. The gradient generated by S_3 is expected to be different with the gradients generated by S_1 and S_2. Hence, to tackle the expected violation between the update $-\eta g_{S_3}$ and the accumulated update $-\eta\sum_{i=1}^{2} g_{S_i}$, the proposed DCL method finds a corrected update $-\eta\tilde{g}_{S_3}$ (the pink arrow) by solving a quadratic programming problem (5). In this way, the angle between $-\eta\tilde{g}_{S_3}$ and $-\eta\sum_{i=1}^{2} g_{S_i}$ (the blue arrow), i.e., $\tilde{\alpha}$, is guaranteed to be equal to or less than α. Note that the gradient descent processes with or without the proposed DCL module is identical in the test phase.

training process that starts from w_j to learn out w_n can be defined as

$$\nu_{w_j \to w_n | w_m} = \frac{1}{n-j+1}\sum_{i=j}^{n} \nu_{i|w_m},\ m \le j < n. \tag{4}$$

Since the concept of congruency is built upon cosine similarity, $\nu_{k|w_m}$ will range from $[-1, 1]$. Another advantage of using cosine similarity is the tractability. The gradient computed from the loss is considered as a vector in weight space. Hence, cosine similarity can take any pair of gradients, such as the accumulated gradient and the gradient computed by a training sample, or two gradients computed by two respective training samples.

3.3 Task-Specific Factors

Congruency is semantics-aware. As congruency is based on the gradients which are computed with the images and the semantic ground-truth, such as class label in the classification task or human fixation in the saliency prediction task. Therefore, congruency reflects the task-specific semantics. We discuss congruency task-by-task in the following subsection.

Saliency Prediction. Visual attention is attracted to visually salient stimuli and is affected by many factors, such as scale, spatial bias, context and scene composition, oculomotor constraints, and so on. These factors result in high variabilities over fixations across various persons. The variabilities of visual semantics imply that same class objects in two images may have different salience levels, i.e., one object is predicted as salient object while the other same class object is not. In this sense, negative congruency in learning for saliency prediction may result from both feature-level and label-label disagreement across the images.

Continual Learning. In the continual learning setting [24], [33], [40], a classification model is learned with past observed classes and samples. New samples w.r.t. the unobserved classes may be distinct from previously seen samples in terms of both visual appearance and label. This leads to negative congruency in learning.

Classification. For classification, the class labels are usually deterministic to human. The factors that cause negative congruency in learning lie in visual appearances. Due to the variability of real-world images, visual appearance of samples from the same class may be very different from each other in different images.

4 Methodology

In this section, we first overview the proposed DCL method. Then, we introduce its formulation and properties in detail. Finally, we discuss the lower bound of congruency with gradient descent methods. For simplicity, we assume it is at time step k and omit underscored k in the following formulations unless we explicitly indicate it.

4.1 Overview

Fig. 2 demonstrates the basic idea of the proposed DCL method. Given training sample (I, y), where I is an image and y is the ground-truth, the corresponding feature x are first generated by the sample before it is passed to the classifier for computing the predictions $\hat{y} = f_w(x)$. Conventionally, the derivatives g of the loss $\ell(\hat{y}, y)$ are computed to determine the update Δw by an optimizer to back-propagate the error layer by layer. In the proposed DCL method, g is taken to estimate a corrected gradient $\tilde{g}$ that is congruent with previous updates. For example, as shown in Fig. 2, the gradient of S_3 is expected to have a large deviation angle α to the accumulated anti-gradient $-\sum_{i=1}^{2} g_{si}$ because S_1 and S_2 share similar visual appearance, but S_3 is different from them. The proposed DCL method aims to estimate a corrected $\tilde{g}$ which has a smaller deviation angle $\tilde{\alpha}$ to $-\sum_{i=1}^{2} g_{S_i}$.

4.2 Direction Concentration Learning

The core idea of the proposed DCL method is to concentrate the current update to a certain search direction. The accumulated gradient $\hat{g}$ is the direction voted by previous updates which provides information towards the possible local minimum. Ideally, according to the definition of congruency, i.e., Eqs. (3) and (4), cosine similarity should be considered in optimization. However, minimizing cosine similarity with constraints is complicated. Therefore, similar to GEM [33], we adopt an alternative that optimizes the inner product, instead of the cosine similarity. According to Eq. (3), $< g_1, g_2 > \ge 0$ indicates that the angle between the two vectors is less than or equal to $90°$.

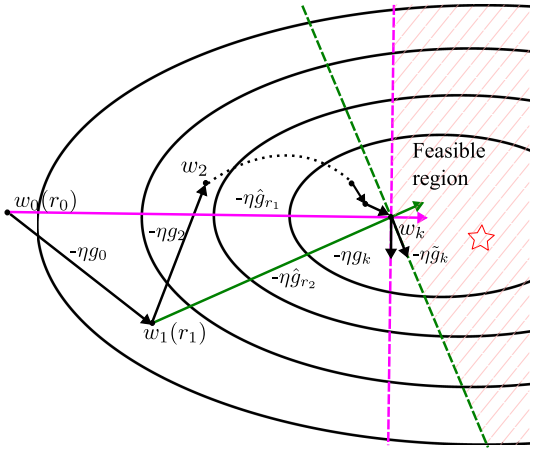


Fig. 3. An illustration of DCL constraints with two reference points $r_0 = w_0, r_1 = w_1$. $\hat{g}_{r_0}$ is the pink arrow while $\hat{g}_{r_1}$ is the green one. The colored dashed line indicates the border of feasible region with regards to $-\hat{g}_{r_i}, i \in \{0, 1\}$, since Constraint (6) forces $-\eta\tilde{g}_k$ to have an angle which is smaller than or equal to 90° w.r.t. $\hat{g}_{r_0}$ and $\hat{g}_{r_1}$.

As shown in Fig. 2, the proposed DCL method uses the accumulated gradient as a referential direction to compute a corrected gradient $\tilde{g}$, i.e.,

$$\begin{aligned} \underset{\tilde{g}}{\text{minimize}} \quad & \frac{1}{2}\|\tilde{g} - g\|_2^2 \\ \text{s.t.} \quad & \langle -\hat{g}_{r_i}, -\tilde{g}\rangle \geq 0, \quad 1 \leq i \leq N_r, \end{aligned} \tag{5}$$

where r_i is a reference point in weight space, $\hat{g}_{r_i}$ is the accumulated gradient that starts the accumulation from r_i, and N_r is the number of reference points. The accumulated gradient $\hat{g}_{r_i}$ indicates that the accumulation starts from the reference r_i to the current weights w. The proposed DCL method can take N_r points as the references $\{r_i | 1 \leq i \leq N_r\}$. Assume that the weights at time step t is taken as the reference r_i, i.e., $r_i = w_t$, we denote $sub(\cdot)$ as a function to find the index of a point in weight space. For example, with $t = sub(r_i) = sub(w_t)$, we can compute the accumulated gradient $\hat{g}_{r_i} = \sum_{j=sub(r_i)} g_j$. On the other hand, the function $\frac{1}{2}\|\tilde{g} - g\|_2^2$ is widely used in gradient-based methods [15], [25], [33], [44], [48] and forces $\tilde{g}$ to be close to g in euclidean space as much as possible. The constraints $\langle -\hat{g}_{r_i}, -\tilde{g}\rangle \geq 0$ are to guarantee that the gradient that is used for an update should not substantially deviate from the accumulated gradient.

In practice, instead of directly computing $\hat{g}_{r_i}$ by its definition (2)), we compute it by subtracting the current point w with the reference point r_i, i.e., $\hat{g}_{r_i} = w - r_i = -\eta\sum_{j=i} g_j$. Hence, the constraints can be deformed in a matrix form

$$A(-\tilde{g}) = -1 \times \begin{bmatrix} (w - r_1)^\top \\ (w - r_2)^\top \\ \vdots \\ (w - r_{N_r})^\top \end{bmatrix} \tilde{g} \geq 0. \tag{6}$$

Fig. 3 demonstrates the effect of constraints in optimization. The dashed line in the same color indicates the border of feasible region with regards to $-\hat{g}_{r_i}, i \in \{1, 2\}$ as Constraint (6) forces $\tilde{g}$ to have an angle smaller than 90°. Due to two references in this example, the intersection between two feasible regions, i.e., the shaded region, is the intersected feasible region for optimization. Note that an accumulated gradient determines half-plane (hyperplane) as feasible region, instead of the full plane (hyperplane) in conventional gradient descent case.

The optimization (5) becomes a classic quadratic programming problem and we can easily solve it by off-the-shelf solvers like quadprog[1] or CVXOPT.[2] However, since the size of $\tilde{g}$ can be sufficiently large, straightforward solution may be computationally expensive in terms of both time and storage. As introduced by Dorn [7], we apply a primal-dual method for quadratic programs to solve it efficiently.

Given a general quadratic problem, it can be formulated as follows:

$$\underset{z}{\text{minimize}} \quad \frac{1}{2}z^\top Cz + q^\top z \quad \text{s.t.} \quad Bz \geq b, \tag{7}$$

whereas the corresponding dual problem to Problem (7) is

$$\begin{aligned} \underset{u,v}{\text{minimize}} \quad & \frac{1}{2}u^\top Cu + b^\top v \\ \text{s.t.} \quad & B^\top v - Cu = q, \quad v \geq 0. \end{aligned} \tag{8}$$

Dorn provides the proof of the connection between Problems (7) and (8).

Theorem 4.1 (Duality). *if $z = z^*$ is a solution to Problem (7) then a solution $(u, v) = (u^*, v^*)$ exists to Problem (8). Conversely, if a solution $(u, v) = (u^*, v^*)$ to Problem (8) exists then a solution which satisfies $Cz = Cu^*$ to Problem (7) also exists.*

Due to the equality constraint $B^\top v - Cu = q$, assume C is full rank, we can plug $u = C^{-1}(B^\top v - q)$ back to the objective function to further simplify Problem (8), i.e.,

$$\begin{aligned} \underset{v}{\text{minimize}} \quad & \frac{1}{2}v^\top B(C^{-1})^\top B^\top v + (b - p^\top B^\top)v \\ \text{s.t.} \quad & v \geq 0. \end{aligned} \tag{9}$$

Now it turns out to be a quadratic problem w.r.t. v only.

The DCL quadratic problem can be solved by the aforementioned primal-dual method. Specifically, $\|\tilde{g} - g\|_2^2 = (\tilde{g} - g)^\top(\tilde{g} - g) = \tilde{g}^\top\tilde{g} - 2g^\top\tilde{g} + g^\top g$. By omitting the constant term $g^\top g$, it turns to a quadratic problem form $\tilde{g}^\top\tilde{g} - 2g^\top\tilde{g}$. Since we know the primal problem (7) can be converted to its dual problem (8), the related coefficient matrices/vectors are easily determined by

$$C = I, \quad B = -A, \quad b = \mathbf{0}, \quad p = -g,$$

where I is a unit matrix. With these coefficients at hand, we have the corresponding dual problem

$$\underset{v}{\text{minimize}} \quad \frac{1}{2}v^\top AA^\top v - g^\top A^\top v \quad \text{s.t.} \quad v \geq 0. \tag{10}$$

By solving (10), we have v^*. On the other hand, $C\tilde{g} = Cu^*, C = I$ and we can have the solution $\tilde{g}^*$ by

$$\tilde{g}^* = Cu^* = B^\top v - q = -A^\top v + g. \tag{11}$$

1. https://github.com/rmcgibbo/quadprog
2. https://cvxopt.org/

Note that $\tilde{g}, u \in \mathbb{R}^p$, $v \in \mathbb{R}^{N_r}$, $A \in \mathbb{R}^{N_r \times p}$, and $b \in \mathbb{R}^{N_r}$ where p is the size of w. If taking the fully-connected layer of ResNet as w, $p = 2048$. In contrast with p, N_r is usually smaller, i.e., 1,2, or 3. As N_r becomes larger, it increases the possibility that the constraints are inconsistent. Thus, $N_r \ll p$. This implies that solving Problem (10) in $\mathbb{R}^{N_r}$ is more efficient than solving Problem (5) in $\mathbb{R}^p$.

4.3 Theoretical Lower Bound

Here, we discuss about the congruency lower bound with gradient descent methods. First, we recall the theoretical characteristics w.r.t. gradient descent methods.

Proposition 4.2 (Quadratic upper bound [36]). *If the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L for any $x, y \in \mathbb{R}^n$, i.e.,*

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|, \tag{12}$$

then

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2. \tag{13}$$

On the other hand, there is a proved bound w.r.t. the loss.

Corollary 4.3 (The bound on the loss at one iteration [8], [49]). *Let x_k be the kth iteration result of gradient descent and $\eta_k \geq 0$ the kth step size. If ∇f is L-Lipschitz continuous, then*

$$f(x_{k+1}) \leq f(x_k) - \eta_k \left(1 - \frac{L\eta_k}{2}\right)\|\nabla f(x_k)\|^2. \tag{14}$$

By adding up a collection of inequalities, we can move further along this line to have the following corollary.

Corollary 4.4. *Let x_k be the kth iteration result of gradient descent and $\eta_k \geq 0$ the kth step size. If ∇f is L-Lipschitz continuous, then*

$$f(x_k) \leq f(x_0) - \sum_{i=0}^{k-1} \eta_i \left(1 - \frac{L\eta_i}{2}\right)\|\nabla f(x_i)\|^2. \tag{15}$$

Theorem 4.5 (Congruency lower bound). *Assume the gradient descent method uses a fixed step size η and the gradient of the loss function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L, the congruency $\nu_{k|x_0}$ referring to the initial point x_0 at the kth iteration has the following lower bound*

$$\nu_{k|x_0} \geq \max\Bigg\{(1 - L\eta)\sum_{i=0}^{k-1}\frac{\|\nabla f(x_i)\|}{\|\nabla f(x_k)\|} - L\eta\frac{\sum_{i=0}^{k-1}\|\nabla f(x_i)\|\|\sum_{j=0}^{i-1}\nabla f(x_j)\|}{\|\nabla f(x_k)\|\|\sum_{i=0}^{k-1}\nabla f(x_i)\|}, -1\Bigg\}. \tag{16}$$

Proof. Given x_k and x_0, according to Proposition 4.2 we have

$$\nabla f(x_k)^\top (x_k - x_0) \leq f(x_k) - f(x_0) + \frac{L}{2}\|x_k - x_0\|^2.$$

Since $x_k = x_0 - \eta\sum_{i=0}^{k-1}\nabla f(x_i)$ and $\nu_{k|x_0} = (-\nabla f(x_k))^\top (-\sum_{i=0}^{k-1}\nabla f(x_i))/(\|\nabla f(x_k)\| \ \|\sum_{i=0}^{k-1}\nabla f(x_i)\|)$, we can have

$$\begin{aligned}\nabla f(x_k)^\top (x_k - x_0) &= -\eta(-\nabla f(x_k))^\top\Big(-\sum_{i=0}^{k-1}\nabla f(x_i)\Big)\\ &= -\eta\|\nabla f(x_k)\|\|\sum_{i=0}^{k-1}\nabla f(x_i)\|\nu_{k|x_0}.\end{aligned}$$

Plugging this in the inequality, it yields

$$\nu_{k|x_0} \geq \frac{1}{\eta}\frac{f(x_0) - f(x_k) - \frac{L\eta^2}{2}\|\sum_{i=0}^{k-1}\nabla f(x_i)\|^2}{\|\nabla f(x_k)\|\|\sum_{i=0}^{k-1}\nabla f(x_i)\|}.$$

According to Corollary 4.4, the inequality can be rewritten as

$$\nu_{k|x_0} \geq \frac{(1 - \frac{L\eta}{2})\sum_{i=0}^{k-1}\|\nabla f(x_i)\|^2 - \frac{L\eta}{2}\|\sum_{i=0}^{k-1}\nabla f(x_i)\|^2}{\|\nabla f(x_k)\|\|\sum_{i=0}^{k-1}\nabla f(x_i)\|}. \tag{17}$$

By using polynomial expansion and the Cauchy-Schwarz inequality, we can expand the term $\|\sum_{i=0}^{k-1}\nabla f(x_i)\|^2$ as follows:

$$\begin{aligned}&\|\sum_{i=0}^{k-1}\nabla f(x_i)\|^2 = \|\nabla f(x_{k-1}) + \sum_{i=0}^{k-2}\nabla f(x_i)\|^2\\ \leq&\|\nabla f(x_{k-1})\|^2 + 2\|\nabla f(x_{k-1})\|\|\sum_{i=0}^{k-2}\nabla f(x_i)\| + \|\sum_{i=0}^{k-2}\nabla f(x_i)\|^2.\end{aligned}$$

Recursively, $\|\sum_{i=0}^{k-2}\nabla f(x_i)\|^2$, $\|\sum_{i=0}^{k-3}\nabla f(x_i)\|^2$, ..., till $\|\sum_{i=0}^{1}\nabla f(x_i)\|^2$ can be expanded, e.g.,

$$\begin{aligned}&\|\sum_{i=0}^{1}\nabla f(x_i)\|^2 = \|\nabla f(x_1) + \nabla f(x_0)\|^2\\ \leq&\|\nabla f(x_1)\|^2 + 2\|\nabla f(x_1)\|\|\nabla f(x_0)\| + \|\nabla f(x_0)\|^2.\end{aligned}$$

The above inequalities yield

$$\|\sum_{i=0}^{k-1}\nabla f(x_i)\|^2 \leq \sum_{i=0}^{k-1}\|\nabla f(x_i)\|^2 + 2\sum_{i=0}^{k-1}\|\nabla f(x_i)\|\|\sum_{j=0}^{i-1}\nabla f(x_j)\|.$$

Plugging it into Inequality (17), we have

$$\begin{aligned}\nu_{k|x_0} \geq&(1 - L\eta)\frac{\sum_{i=0}^{k-1}\|\nabla f(x_i)\|^2}{\|\nabla f(x_k)\|\|\sum_{i=0}^{k-1}\nabla f(x_i)\|}\\ &- L\eta\frac{\sum_{i=0}^{k-1}\|\nabla f(x_i)\|\|\sum_{j=0}^{i-1}\nabla f(x_j)\|}{\|\nabla f(x_k)\|\|\sum_{i=0}^{k-1}\nabla f(x_i)\|}.\end{aligned}$$

Due to $\frac{\sum_{i=0}^{k-1}\|\nabla f(x_i)\|}{\|\sum_{i=0}^{k-1}\nabla f(x_i)\|} \geq 1$, the congruency lower bound can be further simplified as

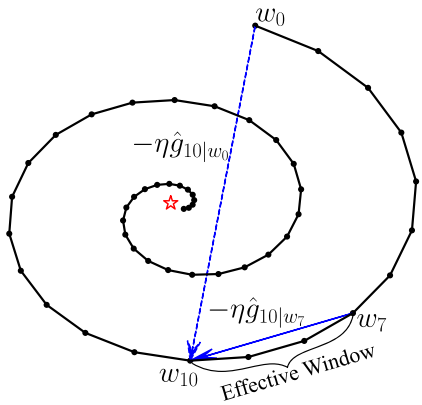


Fig. 4. An illustration to demonstrate the concept of the effective window. Given the spiral convergence path, $-\eta\hat{g}_{10|w_0}$ restricts the search direction and the minimum (i.e., the red star) and w_{11} are unreachable according to the search direction. In contrast, w_{11} can be reached along the search direction of $-\eta\hat{g}_{10|w_7}$. To adaptively yield appropriate accumulated gradients that converge to the minimum, we define an effective window to periodically update the reference.

$$\nu_{k|x_0} \geq (1 - L\eta) \sum_{i=0}^{k-1} \frac{\|\nabla f(x_i)\|}{\|\nabla f(x_k)\|} - L\eta \frac{\sum_{i=0}^{k-1} \|\nabla f(x_i)\| \|\sum_{j=0}^{i-1} \nabla f(x_j)\|}{\|\nabla f(x_k)\| \|\sum_{i=0}^{k-1} \nabla f(x_i)\|}.$$

Combining with the fact $\nu_{k|x_0} \geq -1$, we complete the proof. □

Remark 4.6. Theorem 4.5 implies that when we apply gradient descent method to search a local minimum, the congruency lower bound at a certain iteration in the learning process is determined by the gradients at current iteration and previous iterations.

Remark 4.7. Theorem 4.5 implies that the lower bound of congruency with a small step size, i.e., $\eta < \frac{1}{L}$, is tighter than the one of congruency with a large step size, i.e., $\eta \geq \frac{1}{L}$. This is consistent with the fact the large step size could lead to a zigzag convergence path. The negative lower bound of congruency when $\eta \geq \frac{1}{L}$ indicates the huge turnaround would possibly occur in the learning process.

4.4 Adaptivity to Learning

As the reference is used to compute the accumulated gradient for narrowing down the search direction, a desirable referential direction should orient to a local minimum. Conversely, an inappropriate referential direction could mislead the training and slow down the convergence. Therefore, it is important to update the references to adapt to the target optimization problem.

In this work, we update the references with a short temporal window so as to yield a locally stable and reliable referential direction. For instance, Fig. 4 shows an unfavorable case that takes w_0 as the reference, where the convergence path is spiral. Due to the circuitous manifold, w_0 results in a misleading direction $-\eta\hat{g}_{10|w_0}$. In contrast, if taking w_7 as a reference, it can yield the appropriate search direction to reach w_{11}. Therefore, we introduce an "effective window" to allow the proposed DCL method to find an appropriate search direction. The effective window forces the proposed DCL method to only accumulate the gradients within the window. In Fig. 4, the proposed DCL method with a small window size would converge, whereas the one with a large window size would diverge. We denote the window size as β_w and the reference offset as β_o. When the time step t satisfies

$$t \bmod \beta_w = \beta_o, \tag{18}$$

where mod is the modulo operator, it would trigger the reset mechanism, i.e., starting over to set references $r_i \leftarrow w_t$, $1 \leq i \leq N_r$. β_o indicates the first reference weight point. Once the reset process starts, the proposed DCL method would use g, instead of $\tilde{g}$, for update until all the N_r references are reset.

4.5 Effect of DCL

To intuitively understand the effect of the proposed DCL method, we present visual comparisons of the convergence paths with three popular optimizers, i.e., SGD [42], RMSProp [14], and Adam [23], on a publicly available problem.[3]

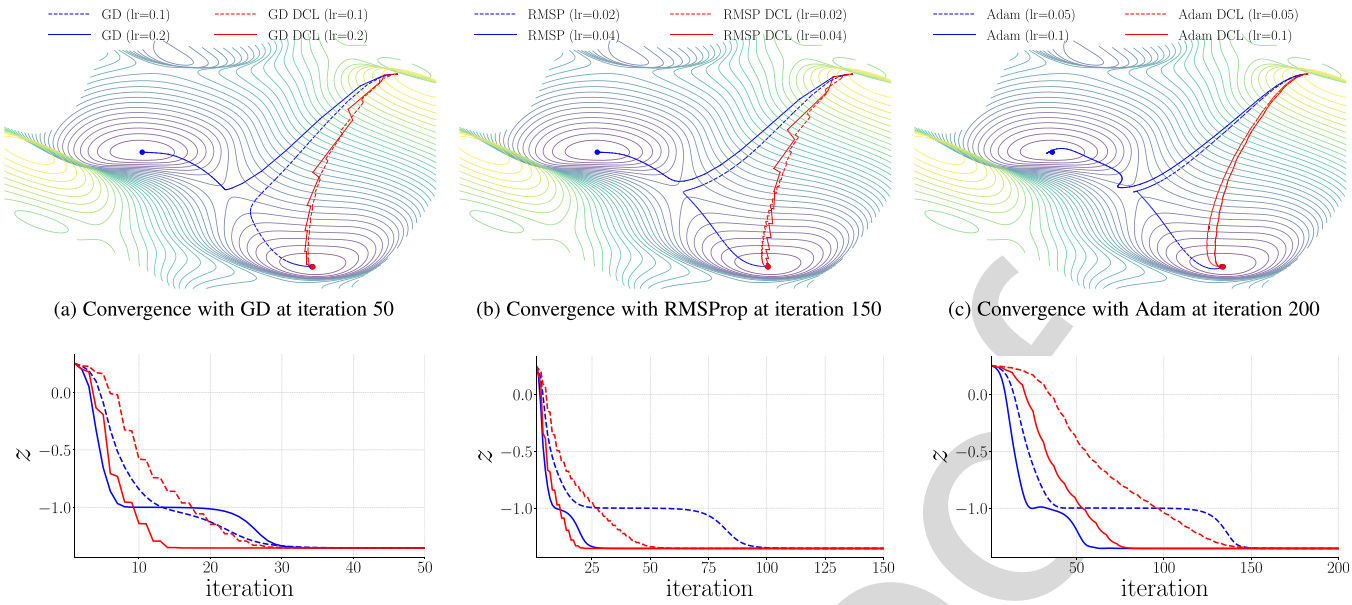
In particular, given the problem $z = f(x, y)$, we apply the three optimizers to compute a local minimum (x^*, y^*). Unlike image classification, the problem does not need randomized data sequence as input so there is no stochastic process. For a fair comparison, except the learning rate, we keep the settings and hyperparameters the same between ALGO and ALGO DCL, where ALGO={GD, RMSProp, Adam} and GD stands for gradient descent. The convergence paths w.r.t. the optimization algorithms are shown in Figs. 5a, 5b, and 5c, while the corresponding z versus iteration curves are plotted in Figs. 5d, 5e, and 5f.

We can see that all the baseline curves are circuitous, i.e., a sharp turn at the ridge region between two local minima. Moreover, different learning rates lead to different local minima. It implies that the training process in this case is influenceable and fickle in terms of the direction of the convergence. The proposed DCL method noticeably improves the convergence direction by choosing a relatively straightforward path over the three optimization algorithms. Note that as the objective function (5) implies, if we do not take any the accumulated gradients (i.e., no constraints), or take the gradient for the coming update as the accumulated gradient (i.e., $\hat{g}_{r_i} = g$), the proposed DCL method would become the baseline (i.e., $\tilde{g} = g$).

4.6 DCL in Continual Learning

In previous subsections, we introduce the proposed DCL method in mini-batch learning. By its very nature, it can also work in continual learning manner. GEM [33] is a recent method proposed for continual learning. The objective function of GEM is the same as the proposed DCL method, whereas the constraints of GEM and the proposed DCL method are devised for respective purposes. To apply the proposed DCL method in continual learning, we can merge the constraints of the proposed method with the ones of GEM. Hence, we have a new A as follows:

3. https://github.com/Jaewan-Yun/optimizer-visualization



(a) Convergence with GD at iteration 50 (b) Convergence with RMSProp at iteration 150 (c) Convergence with Adam at iteration 200

(d) Curves of z v.s. iteration with GD (e) Curves of z v.s. iteration with RMSProp (f) Curves of z v.s. iteration with Adam

Fig. 5. An example demonstrating the effect of the proposed DCL method on three optimizers, i.e., gradient descent (GD), RMSProp, and Adam. Given a problem $z = f(x, y)$, we use these optimization algorithms to compute the local minima, i.e., (x^*, y^*) that yield the minimal z^*. In the experiment, except the learning rate, the setting and hyperparameters are the same for ALGO and ALGO DCL, where ALGO={GD, RMSProp, Adam}. The proposed DCL method encourages the convergence paths to be as straight as possible.

$$A = \begin{bmatrix} (w - r_1)^\top \\ \vdots \\ (w - r_{N_r})^\top \\ -g(x_{S_1}, y_{S_1})^\top \\ \vdots \\ -g(x_{S_{N_m}}, y_{S_{N_m}})^\top \end{bmatrix}, \quad S_i \in \mathcal{M}, \tag{19}$$

where $\mathcal{M}$ is the memory and N_m is the size of the memory. With the proposed DCL constraints, the corrected $\tilde{g}$ is forced to be consistent with both the accumulated gradients and the directions of gradients generated by the samples in memory.

4.7 Comparison With Memory-Based Constraints

Now, we discuss the difference between the proposed DCL constraints and the memory-based constraints used in GEM [33].

There are two main differences between the DCL constraints and the GEM constraints. First, as shown in Fig. 6, the descent direction in the proposed DCL method is regulated by the accumulated gradient, whereas the gradient for an update in GEM is regulated to avoid the violation with the gradients of the memory samples (i.e., images and the corresponding ground-truths). Since the weights are iteratively updated and the memory samples are preserved, the gradients of the memory samples could be changed at each iteration so the direction of the adjusted gradient could be dynamically varying. Second, the proposed DCL method only needs to memorize the references, whereas GEM memorizes the images and the corresponding ground-truths. The proposed DCL constraints are efficiently computed by a subtraction in Eq. (6), other than by computing the corresponding gradients like GEM.

Although the proposed DCL constraints are different from GEM constraints in terms of definition, they are able to work with each other in continual learning. We will dive into the details in the following experiment section. Moreover, GEM computes the gradients on all the parameters of a DNN. This works in the situations that input image resolution is relatively small, e.g., 784 for MNIST [29] or 3,072 for CIFAR-10/100 [26]. The networks used to classify these images have small number of weights like MLP and ResNet-18. However, the number of parameters in a DNN could be huge. For example, ResNeXt-29 (16×64) [56] has 68 million parameters. Although GEM applies primal-dual method to reduce the computation in optimization, the overall computation is still considerably high. In this work, we instead compute the gradients on the highest-level layer to generalize the proposed DCL method to any general DNN.

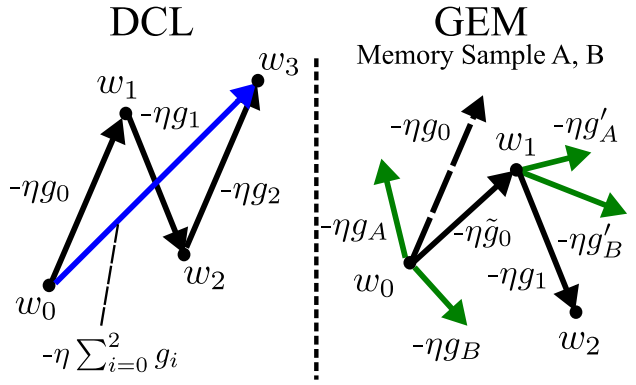


Fig. 6. An illustration demonstrating the difference between DCL (left) and GEM [33] (right). The search direction in DCL is determined by the accumulated gradient while the adjusted gradient (solid line) of GEM is optimized by avoiding the violation between the gradient (dashed line) and memory samples' gradients (green line). Since the weights are iteratively updated and the memory samples are preserved, the direction of the adjusted gradient of the memory samples could be dynamically varying.

TABLE 1
Saliency Prediction Performance of the Models Which are Trained on SALICON 2017 Training Set and Evaluated on SALICON 2017 Validation Set

	NSS	sAUC	AUC	CC
ResNet-50 RMSP	1.7933 ± 0.0083	0.8311 ± 0.0017	0.8393 ± 0.0039	0.8472 ± 0.0048
ResNet-50 RMSP GEM	1.7522 ± 0.0150	0.8267 ± 0.0017	0.8341 ± 0.0016	0.8291 ± 0.0033
ResNet-50 RMSP DCL-∞-1	**1.8226** ± 0.0014	**0.8376** ± 0.0017	**0.8445** ± 0.0016	**0.8569** ± 0.0032
ResNet-50 Adam	1.7978 ± 0.0019	0.8328 ± 0.0007	0.8405 ± 0.0011	0.8495 ± 0.0004
ResNet-50 Adam GEM	1.7962 ± 0.0034	0.8344 ± 0.0021	0.8399 ± 0.0009	0.8494 ± 0.0034
ResNet-50 Adam DCL-∞-1	**1.8019** ± 0.0024	**0.8360** ± 0.0023	**0.8430** ± 0.0023	**0.8548** ± 0.0038
DINet Adam [59]	1.8786 ± 0.0063	0.8426 ± 0.0008	0.8489 ± 0.0008	0.8799 ± 0.0010
DINet Adam GEM	1.8746 ± 0.0067	0.8423 ± 0.0014	0.8492 ± 0.0012	0.8791 ± 0.0030
DINet Adam DCL-500-1	**1.8857** ± 0.0006	**0.8430** ± 0.0002	**0.8493** ± 0.0002	**0.8804** ± 0.0009

Higher score is better in all the metrics. Each experiment is repeated for 3 times and the mean and std of the scores are reported. We follow [59] to only use Adam as the optimizer for DINet.

TABLE 2
Saliency Prediction Performance of the Models Which are Trained on OSIE and Tested on MIT1003

	NSS	sAUC	AUC	CC
ResNet-50 RMSP	2.4047 ± 0.0055	0.7612 ± 0.0019	0.8455 ± 0.0028	0.7595 ± 0.0002
ResNet-50 RMSP GEM	2.3960 ± 0.0057	0.7566 ± 0.0045	0.8412 ± 0.0055	0.7500 ± 0.0037
ResNet-50 RMSP DCL-∞-1	**2.4252** ± 0.0053	**0.7620** ± 0.0018	**0.8469** ± 0.0027	**0.7658** ± 0.0016
ResNet-50 Adam	2.4064 ± 0.0015	0.7597 ± 0.0012	0.8429 ± 0.0021	0.7618 ± 0.0005
ResNet-50 Adam GEM	2.3685 ± 0.0065	0.7594 ± 0.0007	0.8427 ± 0.0017	0.7524 ± 0.0011
ResNet-50 Adam DCL-∞-1	**2.4108** ± 0.0063	**0.7613** ± 0.0007	**0.8442** ± 0.0008	**0.7617** ± 0.0007
DINet Adam	2.4406 ± 0.0058	0.7570 ± 0.0005	0.8442 ± 0.0016	0.7534 ± 0.0005
DINet Adam GEM	2.4456 ± 0.0037	0.7571 ± 0.0005	0.8432 ± 0.0003	0.7540 ± 0.0006
DINet Adam DCL-120-1	**2.4566** ± 0.0007	**0.7611** ± 0.0011	**0.8476** ± 0.0008	**0.7597** ± 0.0008

Each experiment is repeated for 3 times and the mean and std of the scores are reported.

5 Experiments

5.1 Experimental Setup

To comprehensively evaluate the proposed DCL method, we conduct experiments on three tasks, i.e., saliency prediction, continual learning, and classification.

5.1.1 Datasets

For saliency prediction task, we use SALICON [20] (the 2017 version), MIT1003 [22], and OSIE [58]. For continual learning task, we follow the same experimental settings in GEM [33] to use MNIST Permutations (MNIST-P), MNIST Rotations (MNIST-R), and incremental CIFAR-100 (iCIFAR-100). For classification, we use CIFAR [26], Tiny ImageNet, and ImageNet [6].

5.1.2 Models

For saliency prediction, we adopt an improved SALICON saliency model [17] and DINet [59] as the baselines. Both the baseline models takes ResNet-50 [13] as the backbone architecture.

For continual learning, we adopt the same models used in GEM, i.e., Multiple Layer Perceptron (MLP) and ResNet-18, as well as EfficientNet-B1 [47] as the backbone architecture for evaluation. EWC [24] and GEM are used for comparison.

For classification, we use the state-of-the-art model without any architecture modifications for a fair evaluation. ResNeXt [56] (i.e., ResNeXt-29), DenseNet [16] (i.e., DenseNet-100-12), and EfficientNet-B1 [47] are used in the evaluation of CIFAR-10 and CIFAR-100. ResNet (i.e., ResNet-101), DenseNet (i.e., DenseNet-169-32), and EfficientNet-B1 [47] are used in the experiments on Tiny ImageNet. ResNet (i.e., ResNet-34 and ResNet-50) is used in the experiments on ImageNet.

5.1.3 Notation

For convenience, we notate *model name* + *optimizer name* + *DCL-β_w-N_r* for key experimental details in Tables 1, 2, 6 and 7. $\beta_w = \infty$ indicates it never resets the references when the initialization of references is finished.

5.1.4 Evaluation Metrics

For saliency prediction, we report the performance using the commonly use metrics, namely area under curve (AUC) [1], [21], shuffled AUC (sAUC) [1], [45], normalized scanpath saliency (NSS) [18], [43], and correlation coefficient (CC) [38]. Human fixations are used to form the positive set while the points from the saliency map are sampled to form the negative set. With the two sets, an ROC curve of true positive rate versus false positive rate would be plotted by thresholding over the saliency map. If the points are sampled in a uniform distribution, it is AUC. If the points are sampled from the human fixation points, it is sAUC. NSS would average the response values at human eye positions

TABLE 3
Performances on MNIST-R in Continual Learning Setting Using SGD [42] as the Optimizer

	Accuracy	BWT	FWT
EWC	54.61	-0.2087	0.5574
GEM	83.35	-0.0047	**0.6521**
MLP DCL-30-1 MEM	**84.08**	0.0094	0.6423
MLP DCL-40-1 MEM	84.02	0.0127	0.6351
MLP DCL-50-1 MEM	82.77	**0.0238**	0.6111

The reported accuracy is in percentage. MEM *indicates that the constraints of GEM [33] are concatenated to use as Eq. (19) describes.*

TABLE 4
Performances on MNIST-P in Continual Learning Setting Using SGD as the Optimizer

	Accuracy	BWT	FWT
EWC	59.31	-0.1960	-0.0075
GEM	82.44	0.0224	-0.0095
MLP DCL-3-1 MEM	82.30	0.0248	**-0.0038**
MLP DCL-4-1 MEM	**82.58**	0.0402	-0.0092
MLP DCL-5-1 MEM	82.10	**0.0464**	-0.0095

in an predicted saliency map which has been normalized to be zero-mean and with unit standard deviation. CC measures the strength of a linear correlation between a ground-truth map and a predicted saliency map. For continual learning, we use the same metrics used in GEM [33], i.e., accuracy, backward transfer (BWT), and forward transfer (FWT). For classification, we evaluate the proposed DCL method with top 1 error rate metric on the CIFAR experiments while both top 1 and top 5 error rate are reported in the experiments of Tiny ImageNet and ImageNet.

5.1.5 Experimental and Training Details

In the experiments of saliency prediction, we use Adam [23] and RMSProp (RMSP) [14] optimizers. In the setting with Adam, we use $\eta = 0.0002$, weight decay 1e-5 while $\eta = 0.0005$, weight decay 1e-5 are used within the setting of RMSP. The momentum is set to 0.9 for both Adam and RMSP. η would be adjusted along with the epochs, i.e., $\eta_{k+1} \leftarrow \eta_0 \times 0.5^{k-1}$, where k is the current epoch. The batch size is 8 by default. To fairly evaluate the performances of the models, we use cross-dataset validation technique, i.e., the models are trained on the SALICON 2017 training set and evaluated on the SALICON 2017 validation set, and trained on OSIE and evaluated on MIT1003.

We follow the experimental settings in [33] for continual learning. Specifically, MNIST-P and MNIST-R have 20 tasks and each task has 1,000 examples from 10 different classes. On iCIFAR-100, there are 20 tasks and each task has 2,500 examples from 5 different classes. For each task, the first 256 training samples will be selected and stored as the memory on MNIST-P, MNIST-R, and iCIFAR-100. In this work, GEM constraints are concatenated with the DCL constraints by Eq. (19). As the different concepts are learned across the episodes, i.e., the tasks, we only consider that the accumulation of gradients would take place in each episode.

In the classification task, we evaluate the models with SGD optimizer [42]. The hyperparameters are kept by default, i.e., weight decay 5e-4, initial $\eta = 0.1$, the number of total epochs 300. η would be changed to 0.01 and 0.001 at epoch 150 and 225, respectively. For the Tiny ImageNet experiments, we will train the models in 30 epochs with weight decay 1e-4, initial $\eta = 0.001$. η would be changed to 1e-4 and 1e-5 at epoch 11 and 21, respectively. The momentum is 0.9 by default. The batch size is 128 in the CIFAR experiments and 64 in the Tiny ImageNet experiments. In the ImageNet experiments, we use batch size of 512 to train ResNet-50.

In addition, we present the performance of GEM for reference as well. Note that more samples in memory may lead to inconsistent constraints. We set memory size to 1 and reset the memory at each epoch beginning, which is analogous to the case that GEM for continual learning would reset the memory at each beginning of the episode. The implementations of this work are built upon PyTorch[4] and quadprog package is employed to solve quadratic programming problems.

5.2 Performance Evaluation

5.2.1 Saliency Prediction

Table 1 reports the mean and standard deviation (std) of the scores in NSS, sAUC, AUC, and CC over 3 runs on the SALICON 2017 validation set. We can see that the proposed DCL method overall improves the saliency prediction performance with both ResNet-50 and DINet over all the metrics. Moreover, small values of stds w.r.t. the proposed DCL method show that the randomness caused by the stochastic process does not contribute much to the improvement. Table 2 shows that the proposed DCL method trained on OSIE consistently improves the saliency prediction performance on MIT1003.

Note that Adam and RMSP optimizer are different algorithms to compute effective step sizes based on the gradients. The consistency of the improvement with both optimizers shows that the proposed DCL method generally works with these optimizers.

5.2.2 Continual Learning

As introduced in Section 4, we apply the proposed DCL method to enhance the congruency of the learning process for continual learning. Specifically, following Eq. (19), we concatenate the DCL constraints with the GEM constraints [33]. As reported in Table 3, the proposed DCL method improves the classification accuracy by 0.7 percent on MNIST-R. Similarly, the proposed DCL method improves the classification accuracy on MNIST-P as well (see Table 4). The marginal improvement may results from the difference between MNIST-R and MNIST-P. Permuting the pixels of the digits is harder to recognize than rotating the digits by a fixed angle, and makes the accumulated gradient less informative in terms of leading to the solution. We observe that shorter effective window size is helpful to improve the accuracy in the continual learning task. This is because the training process of continual learning is one-off and a fast variation could be caused by the limited images with brand new labels in each episode. The experiments on

4. https://github.com/pytorch/pytorch

TABLE 5
Performances on iCIFAR-100 in Continual Learning Setting Using SGD as the Optimizer

	Accuracy	BWT	FWT
EWC	48.33	-0.1050	0.0216
iCARL	51.56	-0.0848	0.0000
ResNet GEM	66.67	0.0001	0.0108
ResNet DCL-4-1 MEM	**67.92**	0.0063	0.0102
ResNet DCL-8-1 MEM	67.27	0.0104	0.0190
ResNet DCL-12-1 MEM	66.58	0.0089	0.0139
ResNet DCL-20-1 MEM	66.56	0.0030	0.0102
ResNet DCL-24-1 MEM	64.97	0.0082	**0.0238**
ResNet DCL-32-1 MEM	66.10	**0.0305**	0.0176
ResNet DCL-50-1 MEM	64.86	0.0244	0.0125
EffNet GEM	80.80	0.0318	-0.0050
EffNet DCL-4-1 MEM	**81.55**	0.0383	-0.0048
EffNet DCL-8-1 MEM	80.84	0.0367	**0.0068**
EffNet DCL-12-1 MEM	79.45	0.0322	0.0011
EffNet DCL-20-1 MEM	79.33	0.0316	-0.0095
EffNet DCL-24-1 MEM	79.05	0.0375	-0.0006
EffNet DCL-32-1 MEM	79.97	0.0452	-0.0145
EffNet DCL-50-1 MEM	77.87	**0.0602**	-0.0101

EffNet *stands for EfficientNet [47].*

iCIFAR-100 in Table 5 confirm this pattern. The proposed DCL method with ResNet and $\beta_w = 4$ improves the accuracy by 1.25 percent on iCIFAR-100.

There are another two metrics for continual learning, i.e., forward transfer (FWT) and backward transfer (BWT). FWT is that learning a task is helpful in learning for the future tasks. Particularly, positive FWT is correlated to n-shot learning. Since the proposed DCL method utilizes the directional information of the past updates, it has less influence/correlation to FWT. Hence, we will focus on BWT. BWT is the influence that learning a task has on the performance on the previous tasks. Positive BWT is correlated to congruency in the learning process, while large negative BWT is referred as catastrophic forgetting. Tables 3 and 4 show that the proposed DCL method is useful in improving BWT on MNIST-R and MNIST-P. The BWT of GEM is negative (-0.0047) and the proposed DCL method improves it to 0.0238 on MNIST-R. Similarly, the BWT of GEM is 0.0224 and the proposed DCL method improves it to 0.0464 on MNIST-P. Similarly, in Table 5, the proposed DCL method with ResNet improves BWT of GEM from 0.0001 to 0.0305, while the proposed DCL method with EfficientNet [47] improves BWT to 0.0602.

TABLE 6
Top 1 Error Rate (in %) on CIFAR With Various Models

	CIFAR-10	CIFAR-100
ResNeXt-29 SGD	3.53	17.30
ResNeXt-29 SGD GEM	7.70	32.70
ResNeXt-29 SGD DCL-∞-1	**3.33**	**17.02**
DenseNet-100-12 SGD	4.54	22.88
DenseNet-100-12 SGD GEM	6.92	33.72
DenseNet-100-12 SGD DCL-90-1	**4.32**	**22.16**
EfficientNet-B1 SGD [47]	1.91	11.81
EfficientNet-B1 SGD GEM	3.06	19.48
EfficientNet-B1 SGD DCL-5-1	**1.79**	**11.65**

TABLE 7
Top 1 and Top 5 Error Rate (in %) on the Validation Set of Tiny ImageNet With Various Models

	Top 1 error	Top 5 error
ResNet-101 SGD	17.34	4.82
ResNet-101 SGD GEM	21.78	7.21
ResNet-101 SGD DCL-60-1	**16.89**	**4.50**
DenseNet-169-32 SGD	20.24	6.11
DenseNet-169-32 SGD GEM	26.81	9.43
DenseNet-169-32 SGD DCL-50-1	**19.55**	**6.09**
EfficientNet-B1 SGD	15.73	3.90
EfficientNet-B1 SGD GEM	28.74	11.31
EfficientNet-B1 SGD DCL-8-1	**15.61**	**3.75**

5.2.3 Classification

Table 6 reports the top 1 error rates on CIFAR-10 and CIFAR-100 with ResNeXt, DenseNet, and EfficientNet. In all cases, the proposed DCL method outperforms the baseline, i.e., ResNeXt-29 SGD, DenseNet-100-12 SGD, and EfficientNet-B1 SGD. Specifically, the proposed DCL method with ResNeXt decreases the error rate by 0.2 percent on CIFAR-10 and by 0.28 percent on CIFAR-100, while the proposed DCL method with EfficientNet decreases the error rate by 0.12 percent on CIFAR-10 and by 0.16 percent on CIFAR-100. Similar improvements can be found in the experiments with DenseNet and this shows that the proposed DCL method is generally able to work with various models. Moreover, it can be seen in Table 6 that GEM has a higher error rate than the baseline in the experiments with ResNeXt, DenseNet, and EfficientNet. Because of the dynamical update process in learning, the gradient of the samples in memory does not guarantee that the direction leads to the solution. The direction can be even worse, e.g., it is possible to go in an opposite way to the solution.

A consistent improvement w.r.t. the proposed DCL method can be found in the experiments on Tiny ImageNet (see Table 7). The proposed DCL method decreases top 1 error rate by 0.45 percent with ResNet, by 0.69 percent with DenseNet, and by 0.12 percent with EfficientNet. Also, the performance degradation caused by GEM [33] can be observed that top 1 error rate generated by GEM with ResNeXt is increased by almost 4.44 percent, comparing to the baseline ResNet.

Table 8 reports the mean and std of 1-crop validation error of ResNet-50 on ImageNet. Comparing to Tiny ImageNet and CIFAR, ImageNet has more categories and more high resolution images. Given such difficulties, the

TABLE 8
Top 1 and Top 5 1-Crop Validation Error (in %) on ImageNet With SGD Optimizer

	Top 1 error	Top 5 error
ResNet-50 [13]	24.70	7.80
ResNet-50 (reproduced)	24.33 ± 0.08	7.30 ± 0.07
ResNet-50 DCL	24.09 ± 0.03	7.23 ± 0.02

$\beta_w = 5$ and $N_r = 1$ are used for ResNet-50 DCL. Within the same experimental settings, ResNet-50 GEM does not converge in this experiment. The mean and std of errors are computed over three runs.

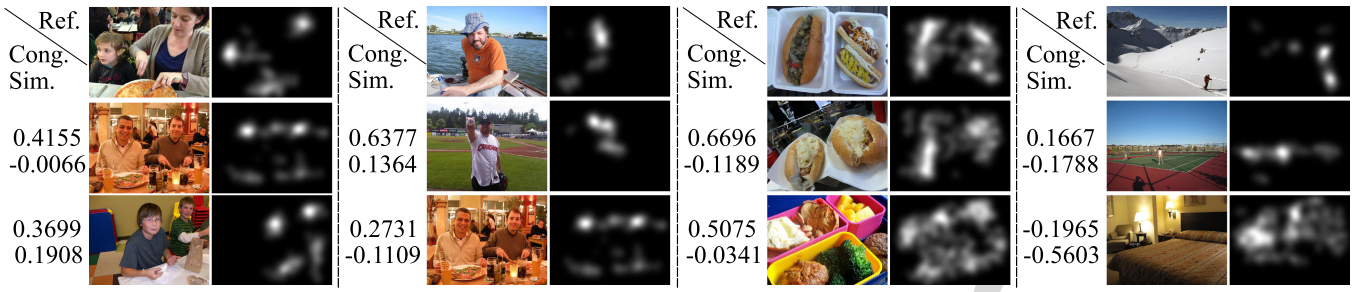


Fig. 7. The congruencies (*Cong.*) generated by the given references (*Ref.*) and samples with the baseline ResNet-50 RMSP in Table 2. The cosine similarities (*Sim.*) between referred images and sample images are provided for comparison purposes. Source images and the corresponding ground-truths, i.e., fixation maps, are displayed along with the congruencies. The first and second block are the results of subset that contains persons in various scenes. The third block is examples of food subset. The rightmost block shows subset with mixed image categories, i.e., contain objects of various categories in various scenes.

proposed DCL method reduces the mean of top 1 errors by 0.24 percent over three runs. In summary, the improvement gained by the proposed DCL method is benefited from the better solution searched by optimizing DCL quadratic programming problem (5).

6 Analysis

In this section, we first validate the defined congruency by comparing through qualitative examples. Then, an ablation study w.r.t. β and N_r is presented. Moreover, we provide a congruency analysis in the training processes for the three tasks. In the end, the comparison between training from scratch and fine-tuning, as well as the computational cost are provided.

6.1 Validity of Congruency Metric

In this subsection, we conduct a sanity check on the validity of the defined congruency. To do this, we consider a simple case where we directly take the gradients (i.e., g_{S_1} and g_{S_2}) of two samples (i.e., S_1 and S_2) to compute the corresponding congruency, i.e., $\nu = \frac{g_{S_1}^\top g_{S_2}}{\|g_{S_1}\|\|g_{S_2}\|}$. For comparison purposes, the cosine similarity, Sim, between raw image S_1 and S_2 is also computed by $Sim = \frac{S_1^\top S_2}{\|S_1\|\|S_2\|}$. Note that congruency is semantics-aware, whereas cosine similarity between the two raw images is semantics-blind. This is because the gradients are computed by images and its semantic ground truth, e.g., the class label in the classification task or human fixation in the saliency prediction task.

For the analysis in the saliency prediction task, we sample 3 subsets, where 20 training samples w.r.t. person, 20 training samples w.r.t. food, and 20 training samples w.r.t. various scenes and categories were sampled from SALICON. For the analysis in the classification task, 3 subsets were sampled from Tiny ImageNet, which comprised of 100 images of tabby cat and Egyptian cat to form a intra-similar-class subset, 100 images of tabby cat and German shepherd dog to form a inter-class subset, and 50 images from various classes to form a mixed subset. In this way, we can analyze the correlation between the samples in terms of congruency. With these subsets, we use the baselines, i.e., ResNet-50 for saliency prediction and ResNet-101 for classification, to yield the samples gradients without updating the model.

Fig. 7 demonstrates the congruencies w.r.t. the references and various samples (image + fixation map). In contrast to the deterministic nature in the classification task, saliency is context-related and semantics-based. It implies that the same objects within two different scenarios may have different saliency labels. Hence, we select the examples of same/similar objects for this experiment. In Fig. 7, the first and second block on left are based on the person subset within various scenarios. The first block consists of the images of person and dining table. Taking the first row sample as reference, the sample in the second row has higher congruency (0.4155) when compared to bottom row sample (0.3699). Although all the fixation maps of all the samples are different, pizza in the second image is more similar to the reference image whereas food in the bottom sample is inconspicuous. In the second block, both the portrait of the fisher (reference) and the portrait of the baseball player (second sample) are similar in terms of the layout, comparing to the persons in dining room (third sample). Their fixation maps are similar as well.

The congruency of the reference and second sample (0.6377) are higher than the one of the reference and third sample (0.2731). In the third block, the image of the reference is three hot dogs and its fixation maps is similar to the fixation maps of the second sample. The two hog dog samples have similar visual appearance and layout of fixations to yield a higher congruency (0.6696). In contrast, third sample is different from the reference in terms of visual appearance and layout of fixations, which yields a lower congruency (0.5075). The rightmost block shows an interesting fact that two outdoor samples yield a positive congruency 0.1667, whereas the outdoor reference and the indoor sample yield a negative congruency -0.1965. One possible reason is that the fixation pattern are different between the reference and the bottom indoor sample. In addition, the visual appearance like illumination may be the another factor causing such the discrepancy.

For classification, Fig. 8 shows the congruencies w.r.t. the references and given samples in each subset. In all cases, we first observe that images with same genuine class as references yield high congruency, i.e., larger than 0.94 for all cases. These show that the gradients of the same labels are similar in the direction of the updates. Another observation is that the congruency of pairs with different labels are significantly smaller than the matched label counterpart. In Fig. 8, the congruencies of the reference (Tabby cat) and Egyptian cat images are below 0.03, while the congruencies of the reference and German shepherd dog images are below 0.016 in

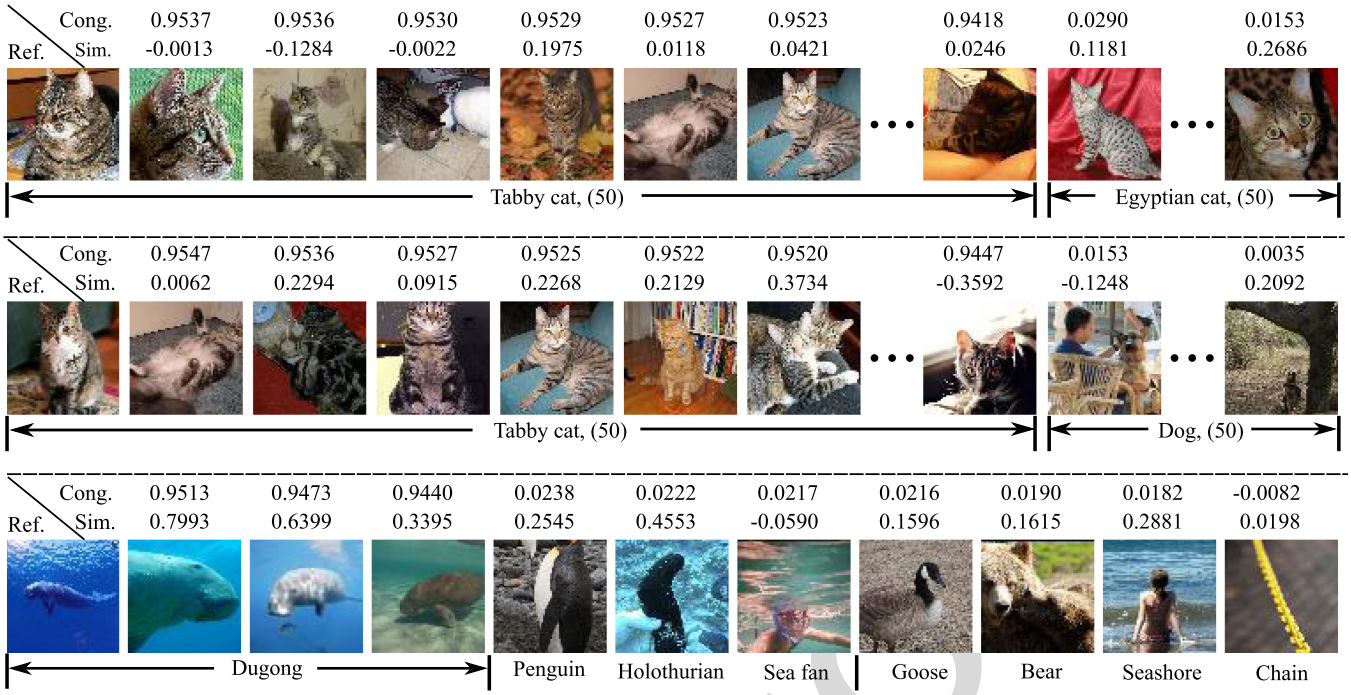


Fig. 8. The congruencies (*Cong.*) generated by the given references (*Ref.*) and samples with the baseline ResNet-101 SGD in Table 7. The images with its labels are displayed along with the congruencies. The cosine similarities (*Sim.*) between referred images and sample images are provided for comparison purposes. The first block is the results of the intra-similar-class subset consisting of images of tabby cat and Egyptian cat. The middle block is the results of the inter-class subset consisting of images of tabby cat and German shepherd dog. The value in bracket indicates number of images. The bottom block is the results of images of various labels.

the middle block. These demonstrate that the gradients of inter-class samples are nearly perpendicular to each other. The reference of class 'Dugong' has positive congruencies w.r.t. all the images that fall in the category of animal, except for the image of chain, which falls into a non-animal category. Last but not least, given the images with different labels, similar visual appearance would lead to relatively higher congruency. For example, the congruencies between tabby cat and Egyptian cat are overall higher than the ones between tabby cat and German shepherd dog. In summary, the labels are an important factor to influence the direction of the gradient in the classification task. Second, the visual appearance is another factor for congruency.

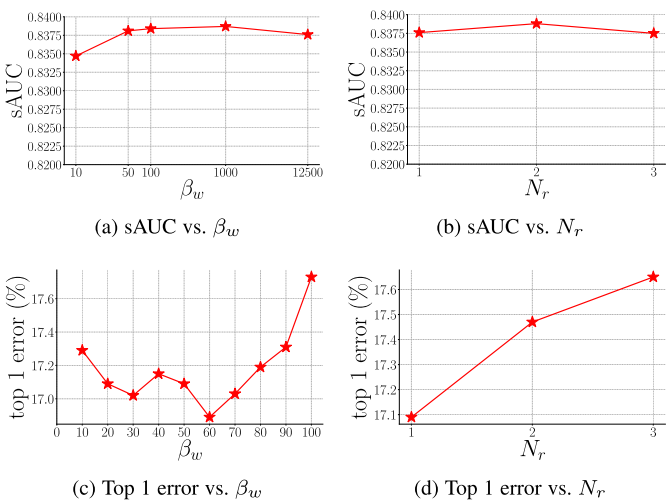
In contrast with congruency, cosine similarity between two raw images make less sense in the context of a specific task. For example, two similar dining scenes in the first column in Fig. 7 yield a negative cosine similarity -0.0066 in the saliency prediction task. Similarly, the first two cat images in the first row in Fig. 8, which are cast to the same category, yield a negative cosine similarity -0.0013. The negative cosine similarity between two images with the same or similar ground truth are counterintuitive. It results from the fact that cosine similarity between two images only focuses on the difference between two sets of pixels and ignores the semantics associated to the pixels.

6.2 Ablation Study

In this subsection, we study the effects of effective window size β_w and reference number N_r on saliency prediction task (with SALICON) and classification task (with Tiny ImageNet).

In the saliency prediction experiment, Fig. 9a shows the curve of sAUC versus β_w based on DCL-β_w-1, while Fig. 9b shows the curve of sAUC versus N_r based on DCL-∞-N_r. Note that for the reference number study, the training process on SALICON consists of 12,500 iterations so $\beta_w \geq 12500$ is equivalent to $\beta_w = \infty$, which means that it never resets the references in the whole learning process. It can be observed that different β_w and N_r yield relatively similar performance in sAUC. This aligns with the nature of saliency prediction, where it maps features to the salient label and the non-salient label. The features w.r.t. the salient label are highly related to each other so β_w and N_r would pervasively help the learning process make use of congruency.

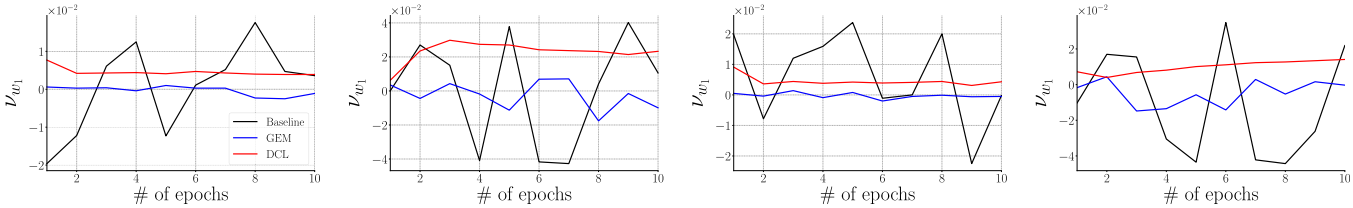
In the classification experiment, Fig. 9c shows the curve of top 1 error versus β_w based on DCL-β_w-1. We can see that



(a) sAUC vs. β_w (b) sAUC vs. N_r

(c) Top 1 error vs. β_w (d) Top 1 error vs. N_r

Fig. 9. Ablation study w.r.t. effective window size β_w and references number N_r. (a) and (b) are the experimental results on the SALICON validation set, while (c) and (d) are with the Tiny ImageNet validation set. $\beta_w = \infty$ in (b) and $\beta_w = 50$ in (d).



(a) Stochastical permutations on OSIE. (b) Stochastical permutations on SALICON. (c) Fixed permutations on OSIE. (d) Fixed permutations on SALICON.

Fig. 10. Congruencies along the epochs in saliency prediction learning, as defined in Eq. (4). The samples sequences for training models are determined by independent stochastic processes in Fig. 10a and 10b, while the permuted samples sequences are pre-determined and fixed for all models in Fig. 10c and 10d. The baseline, GEM, and DCL are ResNeXt-29 SGD, ResNeXt-29 SGD GEM, and ResNeXt-29 SGD DCL-∞-1 (see Table 6), respectively.

only a small β_w range, i.e., between 20 and 70, yields relatively lower errors than the other β_w values. On the other hand, Fig. 9d shows that only using one reference is helpful in the learning process for classification. Different from the pattern shown in Figs. 9a and 9b, where the curves are relatively flat, the pattern in Figs. 9c and 9d implies that the gradients in the learning process for classification are dramatically changed in angle to satisfy the 200-way prediction. Hence, the learning process for classification does not prefer large β_w and N_r.

In summary, the nature of the task should be taken into account to determine the values of β_w and N_r. Both parameters can lead to significantly different performances if the task-specific semantics in the data are highly varying. Specifically, as N_r increases, the feasible region for searching a local minimum possibly becomes narrow as shown in Fig. 3. If the local minimum is not in the narrowed feasible region, large N_r could lead to a slower convergence or even a divergence.

6.3 Congruency Analysis

In this section, we focus on analyzing the patterns of congruency on saliency prediction, continual learning, and classification. For saliency prediction and classification, to study how the gradients of GEM and the proposed DCL method vary in the training process, we compute the congruency of each epoch in the training process by Eq. (4). Specifically, it turns to be $\nu_{w_{es}\rightarrow w_{ee}|w_1}$, where w_{es} and w_{ee} is the weights at the first and last iteration of each epoch, respectively. Here, w_0 is randomly initialized and w_1 represents the starting point of the training. For convenience, we simplify the notation of the average congruency $\nu_{w_{es}\rightarrow w_{ee}|w_1}$ for each epoch as ν_{w_1}. Correspondingly, we define the average magnitude d_{w_1} of the accumulated gradients over the iterations in an epoch, i.e.,

$$d_{w_1} = \frac{1}{sub(w_{ee}) - sub(w_{es}) + 1} \sum_{i=sub(w_{es})}^{sub(w_{ee})} \|w_i - w_1\|_2, \qquad (20)$$

where d_{w_1} indicates the measurement of magnitudes of the accumulated gradients takes w_1 as the reference. Note that Eq. (20) does work not only with an absolute reference (e.g., w_1), but can work with a relative reference (e.g., w_{i-1}) as well. Specifically, we can substitute w_{i-1} for w_1 in Eq. (20) to compute $d_{w_{i-1}}$. Eq. (20) can allow us to peek into the convergence process in the high dimensional weight space, where it is difficult to visualize the convergence. By taking an absolute reference (e.g., w_1) as the reference, it is able to provide an overview about how the learning process converges to the local minimum from the fixed reference, while a relative reference (e.g., w_{i-1}) is helpful to reveal the iterative pattern.

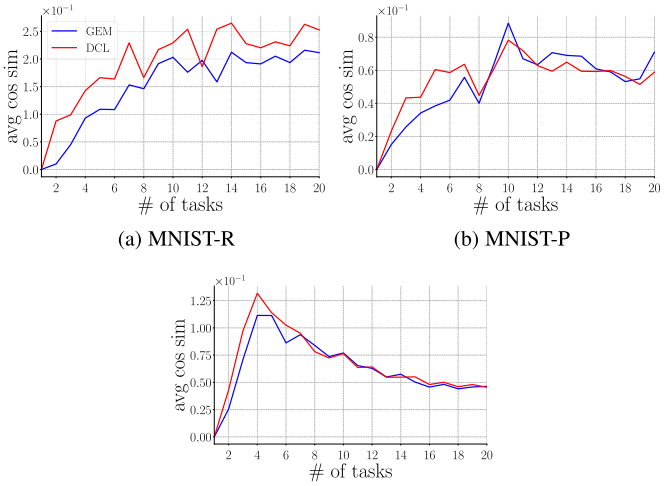
For the experiments of continual learning, since GEM uses the samples in memory to regulate the optimization direction, we follow this setting to check the effect of the proposed DCL method on the cosine similarities between the corrected gradient and the gradients generated by the samples in memory for analysis. More concretely, the average cosine similarity is defined as $\frac{1}{\|N_{it}\|}\frac{1}{\|\mathcal{M}\|}\sum_{i=1}^{N_{it}}\sum_{s\in\mathcal{M}}\cos(g_s, g_{GEMi})$, where N_{it} is the number of iterations in an epoch and g_{GEMi} is the gradient of GEM at ith iteration.

6.3.1 Saliency Prediction

We analyze the models from Table 2, i.e., ResNet-50 Adam (baseline), ResNet-50 Adam GEM (GEM), and ResNet-50 Adam DCL-∞-1 (DCL). As the training samples sequence is affected by the stochastic process and it may be a factor influencing the proposed DCL method, we present two settings, i.e., within the independent stochastic process and within the same stochastic process amid the training of the three models, to gauge the influence of the stochastic process on the proposed DCL method. Specifically, Figs. 10a and 10b are the curves with the independent stochastic process on OSIE and SALICON, respectively, whereas the same permuted samples sequences are used in the trainings of the three models in Figs. 10c and 10d. We can see that they are similar in pattern and it implies that the permutation of the training samples has less influence on the proposed DCL method. Moreover, the proposed DCL method consistently gives rise to a more congruent learning process than the baseline and GEM.

6.3.2 Continual Learning

Figs. 11a, 11b, and 11c shows the congruency along the tasks which are the episodes to learn the new classes. It can be seen that the proposed DCL method significantly enhances the cosine similarities between the gradients for updates and the gradients generated by the samples in memory on MNIST-R. There are improvements made by the proposed DCL method on early tasks on MNIST-P. Moreover, an overall consistent improvement of the proposed DCL method can be observed on iCIFAR-100. Overall, the



(a) MNIST-R (b) MNIST-P

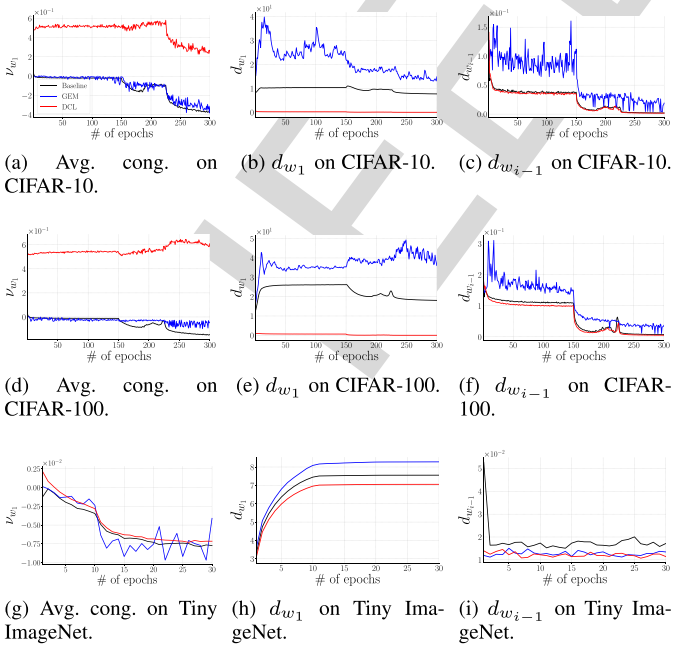
(c) iCIFAR-100

Fig. 11. The average congruencies over epochs in training on the three datasets for continual learning.

corrected updates for the model are computed by proposed DCL method to be more congruent with its previous updates. This consistently results in the improvement of BWT in Tables 3, 4, and 5.

6.3.3 *Classification*

We analyze the models from Table 6, i.e., ResNeXt-29 SGD (baseline), ResNeXt-29 SGD GEM (GEM), and ResNeXt-29 SGD DCL-∞-1 (DCL), in term of the resulting congruency of each epoch in the learning process on CIFAR. Similarly, ResNet-101 SGD, ResNet-101 SGD GEM, and ResNet-101 SGD DCL-50-1 in Table 7 are used for analysis on Tiny ImageNet. The curves of the average congruencies are shown in Figs. 12a, 12d, and 12g, while Figs. 12b, 12e, and 12h show the average magnitudes.

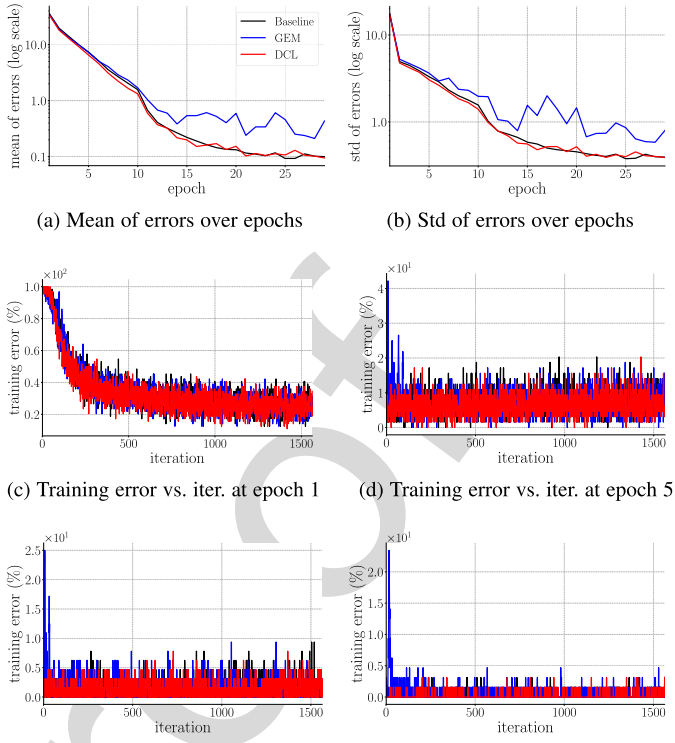


(a) Avg. cong. on CIFAR-10. (b) d_{w_1} on CIFAR-10. (c) $d_{w_{i-1}}$ on CIFAR-10.

(d) Avg. cong. on CIFAR-100. (e) d_{w_1} on CIFAR-100. (f) $d_{w_{i-1}}$ on CIFAR-100.

(g) Avg. cong. on Tiny ImageNet. (h) d_{w_1} on Tiny ImageNet. (i) $d_{w_{i-1}}$ on Tiny ImageNet.

Fig. 12. Analyses of the congruencies and magnitudes along the epochs in classification task, as defined in Eq. (4) and (20).



(a) Mean of errors over epochs (b) Std of errors over epochs

(c) Training error vs. iter. at epoch 1 (d) Training error vs. iter. at epoch 5

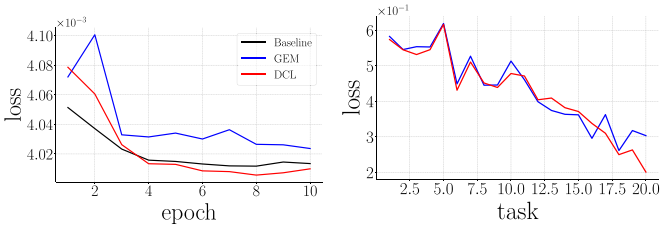
(e) Training error vs. iter. at epoch 10 (f) Training error vs. iter. at epoch 15

Fig. 13. Training error vs. iteration on Tiny ImageNet with ResNet-101. (a) and (b) plot the mean and standard deviation of training errors at each epoch, respectively. Specifically, we show four representative curves of training error vs. iteration at epoch 1, 5, 10, and 15 in (c) – (f), respectively.

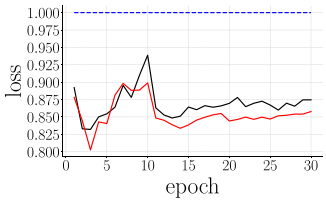
As shown in Figs. 12a, 12d, and 12g, the congruency of the proposed DCL method is significantly higher than the baseline and GEM along all epochs on CIFAR-10 and CIFAR-100. Higher congruency indicates the convergence path would be flatter and smoother. For example, if all the congruencies of each epoch are 0, the convergence path would be a straight line.

On the other hand, the average magnitudes of the proposed DCL method are relatively flat and smooth in Figs. 12b, 12e, and 12h, comparing to the baseline and GEM. Connecting the magnitudes with the congruencies in Figs. 12a, 12d, and 12g, we can infer two points. First, the proposed DCL method finds a nearer local minimum to its initialized weights on CIFAR-10 and CIFAR-100. Because the magnitudes of the proposed DCL method is the smallest among the three methods. Second, the convergence path of the proposed DCL method is the least oscillatory because its congruencies are overall higher than the other two methods and its magnitudes are the lowest among the three methods.

We take a further look at the training error versus iteration curves to better understand the convergences in Fig. 13. To give an overview along all epochs, we compute the mean and standard deviation of the training errors at each epoch and plot them at a logarithm scale in Figs. 13a and 13b, respectively. The results show that the proposed DCL method yields lower training errors from epoch 1 to epoch 16. From epoch 15 onwards, the proposed DCL method is little different from the baseline in terms of the mean because they are both around 0.1. Therefore, we plot the



(a) Saliency prediction on SALICON. (b) Continual learning on iCIFAR-100.



(c) Classification on Tiny ImageNet.

Fig. 14. Validation loss vs. epoch/task. In (c), the dashed blue curve indicates that the classification losses of GEM on Tiny ImageNet are all above 1.0 so they are not shown in the figure for clarity.

representative curve at epoch 1, 5, 10, and 15 in Figs. 13c, 13d, 13e, and 13f.

6.4 Empirical Convergence

Fig. 14 shows the validation losses w.r.t. the three tasks, i.e., saliency prediction (a), continual learning (b), and classification (c). In general, the proposed DCL method achieves lower loss than the baseline and GEM, which is aligned with the fact that the proposed DCL method outperforms the baseline and GEM. Note that classification losses of GEM are above 1.0 so they are not shown in Fig. 14c.

6.5 Training From Scratch Versus Fine-Tuning

We analyze the proposed approach with two types of training scheme on the validation set of Tiny ImageNet. The first training scheme train the models from scratch using the training set of the target dataset, whereas the second training scheme fine-tunes the pre-trained ImageNet models on Tiny ImageNet. For ease of comparison, the experimental results of training the models from scratch on Tiny ImageNet as the fine-tuning results are shown in Table 9. Similar to the results of fine-tuning, the proposed DCL method achieves lower top 1 error (i.e., 67.56 percent) and top 5 error (i.e., 40.74 percent) than the baseline and GEM.

TABLE 9
Top 1 and Top 5 Error Rate (in %) on the Validation Set of Tiny ImageNet

	Top 1 error		Top 5 error	
	TFS	FT	TFS	FT
ResNet-101 SGD	68.21	17.34	42.72	4.82
ResNet-101 SGD GEM	77.20	21.78	53.18	7.21
ResNet-101 SGD DCL	**67.56**	**16.89**	**40.74**	**4.50**

We compare of (1) training the models from scratch (TFS) on Tiny ImageNet, and (2) fine-tuning (FT) the pre-trained ImageNet models on Tiny ImageNet. ResNet-101 SGD DCL is with $\beta_w = 60$ and $N_r = 1$. The validation errors of FT are from Table 7.

TABLE 10
Computational Cost of Training Models on Tiny ImageNet

	# params	proc time
ResNet-101	42.50M	47 ms
ResNet-101 GEM	42.91M	78 ms
ResNet-101 DCL	42.91M	49 ms

The processing time (proc time) per image is calculated by (batch time −data time)/batch size.

6.6 Computational Cost

We report computational cost on Tiny ImageNet, SALICON and ImageNet in Tables 10 and 11, respectively. Specifically, the number of parameters of the models and the corresponding processing time per image are presented. The processing time per image is computed by (batch time −data time)/batch size, where batch time is the time to complete the process of a batch of images, and data time is the time to load a batch of images. Note that the processes of gradient descent with or without the proposed DCL method are the same in the testing phase. We train the models on 3 NVIDIA 1080 Ti graphics cards for the experiments on Tiny ImageNet and SALICON, and on 8 NVIDIA V100 graphics cards for the experiment on ImageNet.

ResNet-101 DCL on Tiny ImageNet is with $\beta_w = 60$ and $N_r = 1$. ResNet-50 DCL is with $\beta_w = \infty$ and $N_r = 1$ on SALICON, and $\beta_w = 1$ and $N_r = 1$ on ImageNet. ResNet-50 GEM is with $N_r = 1$ on all the datasets. The difference of the numbers of parameters between the baseline and the proposed DCL method (or GEM) lies in the final layer, i.e., 1×1 convolutional layer for saliency prediction and the fully connected layer for classification. The proposed DCL method has more parameters to store the weights of the final layer for the references.

In the experiment on Tiny ImageNet, the proposed DCL method with ResNet-101 takes 2 more milliseconds than the baseline to solve the constrained quadratic problem (5). Similarly, with ResNet-50, it takes 1 and 2 more milliseconds than the baseline on SALICON and ImageNet, respectively. This shows that quadratic problems with high dimensional input can be efficiently solved by the tool quadprog. Hence, the proposed DCL method is practically accessible. On the other hand, GEM [33] is less efficient than the other two methods across the three datasets. This is because GEM has to compute the gradients according to the memory, i.e., the input features of the final layer, at each iteration. Instead, the proposed DCL method uses a subtraction operation (i.e., Eq. (6)) to compute the accumulated gradient. Thus, it is faster than GEM.

TABLE 11
Computational Cost of Training Models on SALICON and ImageNet

	SALICON		ImageNet	
	# params	proc time	# params	proc time
ResNet-50	23.51M	64 ms	23.50M	6 ms
ResNet-50 GEM	23.51M	102 ms	25.55M	10 ms
ResNet-50 DCL	23.51M	65 ms	25.55M	8 ms

TABLE 12
The Effect of β_w and N_r on Computational Cost (i.e., Proc Time) With ResNet on Tiny ImageNet

β_w ($N_r = 1$)	proc time	N_r ($\beta_w = 50$)	proc time
10	49 ms	1	49 ms
20	49 ms	5	51 ms
30	49 ms	10	53 ms
40	49 ms	15	54 ms
50	49 ms	20	54 ms

Note that β_w would not affect computational cost because β_w indicates the effective window and resetting the references is implemented as a subtraction operation according to Eq. (6).

Moreover, we discuss the effects of β_w and N_r on the computational cost here. The computational cost w.r.t. β_w and N_r with ResNet on Tiny ImageNet is reported in Table 12. As β_w indicates the effective window, it is implemented by a subtraction operation according to Eq. (6) and updating the reference point is a copying operation in RAM which is fast. Therefore, β_w would not affect computational cost. On the other hand, the time difference between various N_r is small because we only apply the proposed DCL method to the downstream layer, i.e., the final layer, where the parameters are much fewer than the ones used by the whole network. For example, there are only 2,304 parameters in the final convolutional layer for saliency prediction. Any quadratic programming solver like quadprop can efficiently handle the corresponding dual problem (8) in a small scale.

6.7 Discussion of Generalization

Incongruency is ubiquitous in the learning process. It results from the diversity of the input data, e.g., real-world images, and rich task-specific semantics. The proposed DCL method can effectively alleviate the incongruency problem in saliency prediction, continual learning, and classification. Specifically, saliency prediction can be seen as a typical regression problem while continual learning and classification can be seen as a typical learning problem that aims to predict a discrete label. In this sense, the input-output mapping and the learning settings of the three tasks are fundamental to other vision tasks.

From the point of view of task-dependent incongruency, here we consider general vision tasks to be cast into three groups according to the form of input and output. The first group consists of visual tasks that take images as input for classification or regression, e.g., object detection [41] and visual sentiment analysis [61]. In object detection, visual appearance of a region of interest could be diverse in terms of its label and location, while an arbitrary sentiment class can have a number of visual representations in visual sentiment analysis. Since tasks in this group has similar incongruency as that in image classification, i.e., the diversity of raw image features w.r.t. a certain label, the proposed DCL method is expected to boost this type of vision tasks. The second group consists of visual tasks that have complex outputs of regression or classification, e.g., visual relationship detection [30], [34] and human object interaction [31], [57] whose output can involve multiple possible relationships among two or more objects that belong to various visual concepts. The incongruency of tasks in this group lies in the diversity of raw image features w.r.t. a higher dimensional variable, e.g., a relationship which involves multiple objects and corresponding predicates. Last but not least, the third group consists of visual tasks that take a series of images, e.g., action recognition [54]. Usually, it takes a clip of videos as input and incorporates temporal information. The incongruency of tasks in this group lies in the diversity of temporal raw image features w.r.t. a certain label, and the feature space with clips is often more complicated than that in static images. Therefore, the incongruency of tasks in the second and third groups could be more remarkable than that of tasks in the first group. Note that the proposed DCL method is gradient-based and not restricted to specific forms of input or output. Therefore, it could naturally generalize or be used as a starting point to alleviate incongruency for tasks with different forms of input and output in the three groups.

7 Conclusion

In this work, we define congruency as the agreement between new information and the learned knowledge in a learning process. We propose a Direction Concentration Learning (DCL) method to take into account the congruency in a learning process to search for a local minimum. We study the congruency in the three tasks, i.e., saliency prediction, continual learning, and classification. The proposed DCL method generally improves the performances of the three tasks. More importantly, our analysis shows that the proposed DCL method improves catastrophic forgetting.

Acknowledgments

This work was supported in part by the US National Science Foundation under Grants 1908711, 1849107, in part by the University of Minnesota Department of Computer Science and Engineering Start-up Fund (QZ), and in part by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative. The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg).

References

[1] A. Borji, D. N. Sihite, and L. Itti, "Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 55–69, Jan. 2013.

[2] N. Bruce and J. Tsotsos, "Attention based on information maximization," *J. Vis.*, vol. 7, no. 9, pp. 950–950, 2007.

[3] G. A. Carpenter, S. Grossberg, and G. W. Lesher, "The what-and-where filter: A spatial mapping neural network for object recognition and image understanding," *Comput. Vis. Image Understanding*, vol. 69, no. 1, pp. 1–22, 1998.

[4] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1617–1632, Aug. 2017.

[5] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, Oct. 2018.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[7] W. S. Dorn, "Duality in quadratic programming," *Quart. Appl. Math.*, vol. 18, no. 2, pp. 155–162, 1960.

[8] C. Fernandez-Granda, "Lecture notes: Optimization algorithms," Feb. 2016. [Online]. Available: https://math.nyu.edu/cfgranda/pages/OBDA_spring16/material/optimization_algorithms.pdf

[9] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends Cogn. Sci.*, vol. 3, no. 4, pp. 128–135, 1999.

[10] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," 2013, *arXiv:1312.6211*.

[11] R. D. Gordon, "Attentional allocation during the perception of scenes," *J. Exp. Psychol.: Hum. Perception Perform.*, vol. 30, no. 4, 2004, Art. no. 760.

[12] P. Goyal *et al.*, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv: 1706.02677*.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[14] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning: Lecture 6a—Overview of mini-batch gradient descent," 2012. [Online]. Available: https://www.cs.toronto.edu/tijmen/csc321/slides/lecture_slides_lec6.pdf

[15] S. C. H. Hoi, J. Wang, and P. Zhao, "LIBOL: A library for online learning algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 495–499, 2014.

[16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[17] X. Huang, C. Shen, X. Boix, and Q. Zhao, "SALICON: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 262–270.

[18] L. Itti, N. Dhavale, and F. Pighin, "Realistic avatar eye and head animation using a neurobiological model of visual attention," in *Proc. SPIE 48th Annu. Int. Symp. Opt. Sci. Technol.*, 2003, pp. 64–78.

[19] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[20] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "SALICON: Saliency in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1072–1080.

[21] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," MIT Comput. Sci. Artif. Intell. Lab., Cambridge, MA, USA, Tech. Rep. MIT-CSAIL-TR-2012–001, 2012.

[22] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2106–2113.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[24] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Academy Sci. United States America*, vol. 114, no. 13, pp. 3521–3526, 2017.

[25] E. Knight and O. Lerner, "Natural gradient deep Q-learning," 2018, *arXiv:1803.07482*.

[26] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 4, 2009.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] M. Kummerer, T. S. A. Wallis, L. A. Gatys, and M. Bethge, "Understanding low- and high-level contributions to fixation prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4789–4798.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[30] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Dual-glance model for deciphering social relationships," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2669–2678.

[31] Y.-L. Li *et al.*"Transferable interactiveness knowledge for human-object interaction detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3585–3594.

[32] T.-Y. Lin *et al.*"Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[33] D. Lopez-Paz and M. A. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6470–6479.

[34] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 852–869.

[35] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivation*, vol. 24, pp. 109–165, 1989.

[36] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, 1st ed. Berlin, Germany: Springer, 2014.

[37] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate o (1/k^2)," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[38] N. Ouerhani, R. Von Wartburg, H. Hugli, and R. Müri, "Empirical validation of the saliency-based model of visual attention," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 3, no. 1, pp. 13–24, 2004.

[39] R. Ratcliff, "Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions," *Psychol. Rev.*, vol. 97, no. 2, 1990, Art. no. 285.

[40] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5533–5542.

[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[42] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[43] A. L. Rothenstein and J. K. Tsotsos, "Attention links sensing to recognition," *Image Vis. Comput.*, vol. 26, no. 1, pp. 114–126, 2008.

[44] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 901–909.

[45] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *J. Vis.*, vol. 9, no. 12, pp. 15–15, 2009.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[47] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[48] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 260–274, Feb. 2009.

[49] R. Tibshirani, "Lecture notes: Optimization," Sep. 2013. [Online]. Available: http://www.stat.cmu.edu/ryantibs/convexopt-F13/scribes/lec6.pdf

[50] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cogn. Psychol.*, vol. 12, no. 1, pp. 97–136, 1980.

[51] G. Underwood and T. Foulsham, "Visual saliency and semantic incongruency influence eye movements when inspecting pictures," *Quart. J. Exp. Psychol.*, vol. 59, no. 11, pp. 1931–1949, 2006.

[52] G. Underwood, L. Humphreys, and E. Cross, "Congruency, saliency and gist in the inspection of objects in natural scenes," in *Eye Movements*. Amsterdam, The Netherlands: Elsevier, 2007, pp. 563–579.

[53] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[54] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3551–3558.

[55] J. M. Wolfe and T. S. Horowitz, "Five factors that guide attention in visual search," *Nature Hum. Behav.*, vol. 1, no. 3, 2017, Art. no. 0058.

[56] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.

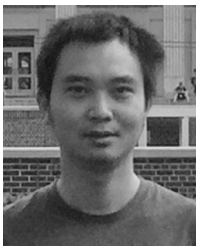
[57] B. Xu, Y. Wong, J. Li, Q. Zhao, and M. S. Kankanhalli, "Learning to detect human-object interactions with knowledge," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2019–2028.

[58] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao, "Predicting human gaze beyond pixels," *J. Vis.*, vol. 14, no. 1, pp. 28–28, 2014.

[59] S. Yang, G. Lin, Q. Jiang, and W. Lin, "A dilated inception network for visual saliency prediction," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2019.2947352.

[60] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.

[61] Q. You, H. Jin, and J. Luo, "Visual sentiment analysis by attending on local image regions," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 231–237.



Yan Luo received the BSc degree in computer science from the Xi'an University of Science and Technology, Xi'an, China. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, University of Minnesota at Twin Cities, Minneapolis, Minnesota. In 2013, he joined the Sensor-enhanced Social Media (SeSaMe) Centre, Interactive and Digital Media Institute, National University of Singapore, as a research assistant. In 2015, he joined the Visual Information Processing Laboratory, National University of Singapore as a PhD student. He worked in the industry for several years on distributed system. His research interests include computer vision, computational visual cognition, and deep learning.

Yongkang Wong received the BEng degree from the University of Adelaide, Adelaide, Australia, and the PhD degree from the University of Queensland, Saint Lucia, Australia. He is currently a senior research fellow with the School of Computing, National University of Singapore. He is also the assistant director of the NUS Centre for Research in Privacy Technologies (N-CRiPT). He has worked as a graduate researcher with NICTA's Queensland Laboratory, Brisbane, OLD, Australia, from 2008 to 2012. His current research interests include the areas of image/video processing, machine learning, action recognition, and human centric analysis. He is a member of the IEEE since 2009.

Mohan Kankanhalli received the BTech degree from IIT Kharagpur, Kharagpur, India, and the MS and PhD degrees from the Rensselaer Polytechnic Institute, Troy, New York. He is currently the Provost's chair professor with the Department of Computer Science, National University of Singapore. He is the director with the N-CRiPT and also the dean, School of Computing, NUS. His current research interests include multimedia computing, multimedia security and privacy, image/video processing, and social media analysis. He is on the editorial boards of several journals. He is a fellow of the IEEE.

Qi Zhao received the PhD degree in computer engineering from the University of California, Santa Cruz, California, in 2009. She is currently an assistant professor with the Department of Computer Science and Engineering, University of Minnesota, Twin Cities. Her main research interests include computer vision, machine learning, cognitive neuroscience, and mental disorders. She was a postdoctoral researcher with the Computation & Neural Systems, and Division of Biology, California Institute of Technology from 2009 to 2011. Prior to joining the University of Minnesota, she was an assistant professor with the Department of Electrical and Computer Engineering and the Department of Ophthalmology, National University of Singapore. She has published more than 50 journal and conference papers in top computer vision, machine learning, and cognitive neuroscience venues, and edited a book with Springer, titled *Computational and Cognitive Neuroscience of Vision*, which provides a systematic and comprehensive overview of vision from various perspectives, ranging from neuroscience to cognition, and from computational principles to engineering developments. She is a member of the IEEE since 2004.