GrOD: Deep Learning with Gradients Orthogonal Decomposition for Knowledge Transfer, Distillation, and Adversarial Training

HAOYI XIONG*, Baidu, Inc., China
RUOSI WAN*, Peking University, China and Baidu, Inc., China
JIAN ZHAO†, Institute of North Electronic Equipment, China
ZEYU CHEN, Baidu, Inc., China
XINGJIAN LI, Baidu, Inc., China
ZHANXING ZHU, Peking University, China
JUN HUAN, Baidu, Inc., China

Regularization that incorporates the linear combination of empirical loss and explicit regularization terms as the loss function has been frequently used for many machine learning tasks. The explicit regularization term is designed in different types, depending on its applications. While regularized learning often boost the performance with higher accuracy and faster convergence, the regularization would sometimes hurt the empirical loss minimization and lead to poor performance. To deal with such issues in this work, we propose a novel strategy, namely *Gradients Orthogonal Decomposition* (**GrOD**), that improves the training procedure of regularized deep learning. Instead of linearly combining gradients of the two terms, **GrOD** re-estimates a new direction for iteration that does not hurt the empirical loss minimization while preserving the regularization affects, through orthogonal decomposition. We have performed extensive experiments to use **GrOD** improving the commonly-used algorithms of transfer learning [2], knowledge distillation [3], adversarial learning [4]. The experiment results based on large datasets, including Caltech 256 [5], MIT indoor 67 [6], CIFAR-10 [7] and ImageNet [8], show significant improvement made by **GrOD** for all three algorithms in all cases.

ACM Reference Format:
Haoyi Xiong, Ruosi Wan, Jian Zhao, Zeyu Chen, Xingjian Li, Zhanxing Zhu, and Jun Huan. 2022. **GrOD**: Deep Learning with Gradients Orthogonal Decomposition for Knowledge Transfer, Distillation, and Adversarial Training. 1, 1 (April 2022), 25 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Both authors contributed equally to this work.
†Corresponding Author: Jian Zhao zhaojian90@u.nus.edu.

This work was partially supported by the National Key R&D Program of China (No. 2021ZD0110303) to Haoyi Xiong, Zeyu Chen and Xingjian Li, and National Science Foundation of China (No. 62006244), Young Elite Scientist Sponsorship Program of China Association for Science and Technology (No. YESS20200140) to Jian Zhao. An earlier version of this work that focuses on the regularization for deep transfer learning only has been accepted for publication as "Towards Making Deep Transfer Learning Never Hurt" [1] in the 2019 IEEE International Conference on Data Mining (ICDM-19).
Authors' addresses: Haoyi Xiong, Baidu, Inc. Baidu Technology Park, Haidian, Beijing, China; Ruosi Wan, School of Mathematical Sciences, Peking University, Haidian, Beijing, China, Baidu, Inc. Baidu Technology Park, Haidian, Beijing, China; Jian Zhao, Institute of North Electronic Equipment, Haidian, Beijing, China; Zeyu Chen, Baidu, Inc. Baidu Technology Park, Haidian, Beijing, China; Xingjian Li, Baidu, Inc. Baidu Technology Park, Haidian, Beijing, China; Zhanxing Zhu, School of Mathematical Sciences, Peking University, Haidian, Beijing, China; Jun Huan, Baidu, Inc. Haidian, Beijing, China.

XXXX-XXXX/2022/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Deep learning has been widely used as a major workhorse for a variety of pattern recognition applications, such as image classification [7, 8], face recognition [9, 10], human parsing [11, 12], biomarker identification [13, 14], and spatiotemporal pattern mining [15, 16]. In many real-world practices of deep learning [17], regularization has been frequently used to improve the performance of deep neural networks training through incorporating explicit regularization terms beyond empirical loss minimization (see also in Chapter 7 of [17]). To regularize the training of deep neural networks, a simple yet effective approach is to build a regularization term to augment the empirical loss through weighted linear combination. Such weights are typically used to make trade-off between empirical loss and model complexity.

Compared to the regularized statistical learning [18] that was originally proposed to avoid over-fitting, regularization nowadays is redesigned in deep learning to enable a wide range of novel learning applications, such as knowledge transfer from pre-trained neural networks [2, 19–21], knowledge distillation via Teacher-Student training [3, 22], and adversarial learning for robustness [4, 23]. While the use of regularization yielding deep learning with better performance and new functionalities, regularized deep learning might sometimes hurt the performance deep neural network and achieve even worse performance than empirical loss minimization (ERM) [18], especially when regularizer weight is inappropriately large.

1.1 Our Observations

While one can fix the over-regularization issue through lowering the regularizer weight for statistical learning, such problem is still tough for regularized deep learning [17]. For example, using the starting point as the reference (SPAR) (i.e., incorporating a L^2-norm regularizer that constrains the distance between the parameter and the starting point of optimization [2]) is frequently used to fine-tune deep neural networks for deep transfer learning. Using an inappropriate pre-trained model for SPAR leads to even worse performance than the one from scratch [24, 25], as the L^2-norm regularization with the start point of optimization would affect the local minimum points that the learning procedure finally converges to, while the selection of poor local minimum points may significantly hurt the generalization performance of deep learning.

We specify above observation using an example based on L^2-SP [2] shown in Figure 1. The Black Line refers to the empirical loss descent flow of common gradient-based learning algorithms with pre-trained weights as the start point. It shows that with the empirical loss gradients as the descent direction, such method quickly converges to a local minimum in a narrow cone, which is usually considered as an over-fitting solution. In the meanwhile, the Blue line demonstrates the possible empirical loss descending path of L^2-SP algorithm, where a strong regularization blocks the learning algorithm to continue lowering the empirical loss while traversing the area around the point of pre-trained weights. An ideal case has been illustrated as the red line, where L^2-SP regularizer helps the learning algorithm to avoid the over-fitting solutions. The overall descent direction adapting L^2-SP regularizer with respect to empirical loss leads to generalizable solutions. There thus needs a method to make both empirical loss and regularizer continue descending to boost the performance of deep transfer learning. Thus, the new technique to balance the regularization term and the empirical loss in the deep learning procedure is needed.

1.2 Our Contributions

Motivated by above observations, simple yet effective training paradigm, *namely Gradients Orthogonal Decomposition* (**GrOD**), which provides a new descent direction estimator for the regularized learning of over-parameterized deep neural networks. **GrOD** follows a simple *Empirical Risk*

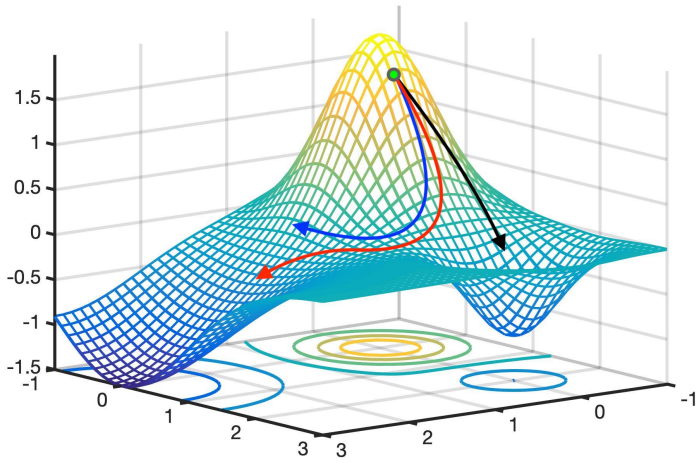


Fig. 1. Flows of descent directions on the Empirical Loss. **Black Line**: the flow via gradient descent direction of empirical loss (i.e., gradient of empirical loss) from a common starting point, where the descent direction quickly leads the learning procedure converged to a local minimum without considering the regularization term and hence may over-fit; **Blue Line**: the flow via the descent direction linearly combining gradients of empirical loss and the regularization term, where the regularization term diminishes the minimization of empirical loss; **Red Line**: the flow via the descent direction balancing the gradients of empirical loss and the regularzation term, where the descent direction leads to a flat area with low empirical loss (i.e., potentials of improved generalizability).

Minimization (ERM) Preserved descent direction principle — in every iteration of the learning procedure, the empirical loss of regularized deep learning should descend as fast as the one based on ERM.Specifically, **GrOD** decomposes the gradients of the regularizer term and removes the part that is opposed to the empirical loss gradients. With remaining parts combined, the regularizer and empirical loss terms are expected to be *"minimized simultaneously"* while the minimization of empirical loss is more preferred in **GrOD**.

Inspired by the observation that regularizer may hurt the model's fitting by preventing empirical loss descending, we proposed a novel regularized deep learning framework **GrOD** that improves regularized deep learning with a better balance between the gradients of loss function and the regularization term — i.e., with better fitness to the training data without simply degrading the weight of regularization term. During the learning procedure,when the angle between the empirical loss gradient and the regularizer gradient is large (larger than 90°), **GrOD** decomposes the regularizer gradients into two components: hurting part (parallel to empirical loss gradients) and safely regularized part (vertical to empirical loss gradients), discards the hurting part and preserves the safely regularized part. In this way, it will preserve regularization affects without preventing empirical loss descending.

In terms of methodology, the most relevant work to our study is Gradient Episodic Memory (GEM) for continual learning [26], which continuously learn the new task using the well-trained models for past tasks. In terms of objectives, **GrOD** aims at lowering the effects of regularization from hurting empirical risk minimization, while GEM prevents the empirical risk minimization from hurting regularization effects (i.e., the accuracy on old tasks). In terms of algorithms, in every

iteration of learning, GEM estimates the descent direction with respect to the gradients of the new task and all past tasks using a time-consuming Quadratic Program (QP), while **GrOD** re-estimates the descent direction from the gradients of the regularizer term and the empirical loss term with low-complexity orthogonal decomposition. All in all, GEM can be considered as a special case of **GrOD** using L^2-SP regularizer [2] based on two tasks.

Extensive experiments have been performed using state-of-the-art algorithms for deep transfer learning [2], knowledge distillation [3] and adversarial learning [4] tasks, on top of a wide range of deep learning benchmark datasets including Caltech, MIT indoor 67, CIFAR-10, Fashion MNIST and ImageNet. The experiments show that **GrOD** always improves the performance of the three tasks. Specifically, for transfer learning tasks, **GrOD** has improved the L^2-SP algorithm [2] with 0.1% – 7% higher accuracy (even transferring from the network pre–trained by inappropriate datasets). Through knowledge distillation [3], the network trained by **GrOD** outperforms the vanilla one with 0.5% – 5% higher accuracy through aligning the generated feature maps. For adversarial learning task, **GrOD** has been evaluated to enhance the state-of-the-art algorithm [4] with significant Pareto-improvement in both accuracy and robustness. Besides, the gradients direction analysis based on the experiments verified our assumptions about the descent direction's performance during neural network's training process.

2 RELATED WORK AND BACKGROUNDS

In this section, we first introduced the preliminary setting of regularized deep learning, then introducing the regularization term for deep transfer learning, knowledge distillation and adversarial learning that would be used in our studies.

2.1 Regularized Deep Learning

Deep convolutional networks usually consist of a great number of parameters that need fit to the dataset. For example, ResNet-110 has more than one million free parameters. The size of free parameters causes the risk of over-fitting. Regularized deep learning is the technique to reduce this risk by constraining the parameters within a limited space with respect to a set of regularization terms. The general learning problem is usually formulated as follow.

Definition 1 (Regularized Deep Learning). *Let's first denote the dataset for the desired task as* $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3) \ldots, (\mathbf{x}_n, y_n)\}$*, where totally n tuples are offered and each tuple* $(\mathbf{x}_i, y_i)$ *refers to the input image and its label in the dataset,* $\mathbf{x} \in \mathbb{R}^D$*,* $y \in \{1, 2, \ldots, N\}$ *for multi-class classificationïijŇand D is the dimensionality of the input data. We then denote* $\omega \in \mathbb{R}^d$ *be the d-dimensional parameter vector containing all d parameters of the training model. Further, given a regularization term* $\Omega(\omega) : \mathbb{R}^d \to \mathbb{R}^d$*, one estimates the parameter of target network through the regularized deep learning paradigms. The optimization object with* ***regularized deep learning*** *is to obtain the minimizer of* $\mathcal{L}(\omega)$

$$\min_{w} \mathcal{L}(\omega) = \left\{ \frac{1}{n} \sum_{i=1}^{n} L(z(\mathbf{x}_i, \omega), y_i) + \lambda \cdot \Omega(\omega) \right\} \tag{1}$$

where (i) the first term $\sum_{i=1}^{n} L(z(\mathbf{x}_i, \omega), y_i)$ *refers to the empirical loss of data fitting while (ii) the second term* $\Omega(\omega)$ *characterizes the affects for transfer learning, knowledge distillation, adversarial learning and so on. z maps* $\mathbb{R}^D \times \mathbb{R}^d$ *to* $\{1, 2, \ldots, N\}$*. The tuning parameter* $\lambda > 0$ *balances the trade-off between the empirical loss and the regularizer.*

2.2 Transfer Learning

When the training dataset size is relatively small, we often need to transfer knowledge learned from large datasets to small tasks [27–31]. Given the weights of a deep neural network pre–trained by a large dataset (e.g., ImageNet), a recent work [2] proposed to first use pre–trained weights as the starting point of the training procedure, then leverages the squared Euclid distance from the training weights to the pre–trained weights as the regularization term for deep transfer learning. Such approach "helps" the training procedure find a generalizable solution with higher accuracy, even based on a small set.

In terms of regularization, given the weights (denoted as Ω_s) of a neural network pre–trained from a large dataset, L^2-SP [2] algorithm uses the squared-euclidean distance from ω to the pre-trained weights $\omega_\mathbf{s}$ of source network (listed in Eq 2) to constrain the learning procedure where

$$\Omega(\omega) = \|\omega - \omega_s\|_2^2 \,. \tag{2}$$

In terms of optimization procedure, L^2-SP makes the learning procedure start from the pre-trained weights (i.e., using $\omega_\mathbf{s}$ to initialize the learning procedure).

In addition to above regularization, other methods have been used for deep transfer learning, including [19, 32–36]. As early as in 2014, authors in [32] reported their observation of significant performance improvement through directly reusing weights of the pre-trained source network to the target task, when training a large CNN with tremendous number of filters and parameters. However, in the meanwhile of reusing all pre-trained weights, the target network might be overloaded by learning tons of inappropriate features (that cannot be used for classification in the target task), while the key features of the target task have been probably ignored. In this way, Yosinki *et al.* [37] proposed to understand whether a feature can be transffered to the target network, through quantifying the "transferability" of features from each layer considering the performance gain. Furthermore, Huh *et al.* [19] made empirical study on analyzing features that CNN learned from ImageNet dataset to other computer vision tasks, so as to detail the factors effecting deep transfer learning accuracy. In recent days, this line of research has been further developed with increasing number of algorithms and tools that can improve the performance of deep transfer learning, including subset selection [33, 38], sparse transfer [34], filter distribution constraining [35], parameter transfer [36], and transfer learning over manifolds [39]. Moreover, [29] studies the memorability of images using transfer learning, while authors in [30, 40] work on the knowledge transfer crossing the modalities. The overall survey on transfer learning can be found in [25, 41]

2.3 Knowledge Distillation

To achieve similar goals, instead of adopting the weights in a straightforward approach, authors [3] propose to use so–called *knowledge distillation* mechanism, where given a pre–trained network as the teacher network it considers the training objective as a student network that learns from the teacher. More specific, the squared Euclid distance between feature maps generated by the convolutional layers of the teacher and student networks are used as regularization [22]. The *feature-wise knowledge distillation* algorithm proposed in [3] enables effective knowledge transfer through learning the behaviors of the pre-trained network, as a gift of knowledge distillation. Similar mechanism is also used for neural network compression [42], using the original network as the teacher and the compression target model as the student with feature map quantization.

Given the training dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and N filters in the teacher/student networks for knowledge distillation, the knowledge distillation algorithm [3] models the regularization as the aggregation of squared-euclidean distances between feature maps outputted by the N filters of

the teacher/student networks, such that

$$\Omega(\omega) = \frac{1}{n}\sum_{j=1}^{N}\sum_{i=1}^{n}\left\|\boldsymbol{F}_j(\omega, \mathbf{x}_i) - \boldsymbol{F}_j(\omega_{\mathbf{s}}, \mathbf{x}_i)\right\|_2^2, \tag{3}$$

where $\boldsymbol{F}_j(\omega, \mathbf{x}_i))$ refers to the feature map outputted by the j^{th} filter ($1 \leq j \leq N$) of the target network based on weight $\mathbf{w}$ using input image $\mathbf{x}_i$ ($1 \leq i \leq n$).

In terms of methodologies, the knowledge distillation was originally proposed to compress deep neural networks [22, 31, 43] through teacher-student network training, where the teacher and student networks are usually based on the same task [22]. In terms of inductive transfer learning, authors in [3] were first to investigate the possibility of using the distance of intermediate results (e.g., feature maps generated by the same layers) of source and target networks as the regularization term. Further, [44] proposed to use the distance between activation maps as the regularization term for so-called "attention transfer". Notice in our experiment settings, we mainly focus on the applications of knowledge distillation to knowledge transfer, i.e. the source model is pre-trained using other datasets.

2.4 Adversarial Learning

In addition to the accuracy improvement, there frequently needs to enhance the robustness of deep learning under adversarial attacks [45, 46]. With a strategy to perturb the training data for adversarial samples generation, [46] proposed to incorporate the training loss based on the generated adversarial samples via Fast Gradient Sign Method (FGSM) as the regularization term to augment the loss for deep adversarial learning. [4] indicated that instead of FGSM, using adversarial examples generated by Projected Gradient Descent (PGD, [47]) on the negative loss function will obtain a more robust model.

Given the training dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, one state-of-the-art adversarial learning algorithm [4], studied in this paper, first synthesizes the adversarial samples set $\{(\mathbf{x}'_1, y_1), \ldots, (\mathbf{x}'_n, y_n)\}$, through perturbation. Then the algorithm proposes to use the empirical loss based on adversarial samples as the objective function to minimize where

$$\mathcal{L}(\omega) = \frac{1}{n}\sum_{i=1}^{n} L(z(\mathbf{x}'_i, \omega), y_i). \tag{4}$$

Using first-order taylor expansion, we can approximate (4) as the regularized form

$$\begin{aligned}
\mathcal{L}(\omega) =& \frac{1}{n}\sum_{i=1}^{n} L(z(\mathbf{x}'_i, \omega), y_i) \\
\approx& \frac{1}{n}\sum_{i=1}^{n}[L(z(\mathbf{x}_i, \omega), y_i) + (\mathbf{x}'_i - \mathbf{x}_i)\frac{\partial}{\partial x}L(z(\mathbf{x}_i, \omega), y_i)] \\
=& \frac{1}{n}\sum_{i=1}^{n} L(z(\mathbf{x}_i, \omega), y_i) \\
&+ \left\{\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}'_i - \mathbf{x}_i)\frac{\partial}{\partial x}L(z(\mathbf{x}_i, \omega), y_i)\right\} \\
=& \frac{1}{n}\sum_{i=1}^{n} L(z(\mathbf{x}_i, \omega), y_i) + \lambda \cdot \Omega(\omega)
\end{aligned} \tag{5}$$

Thus the regularization part of adversarial training is $\Omega(\omega) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}'_i - \mathbf{x}_i)\frac{\partial}{\partial x}L(z(\mathbf{x}_i, \omega), y_i)$ with $\lambda = 1$. Specifically, a pre–trained model using the original dataset is frequently required as the target network for defense, where the gradients and/or Hessian matrices of the loss function are used to perturb the input space of the training data with optional noise to the labels. One can also generate the perturbations for adversarial learning under black–box/derivative–free settings [48, 49]. In addition to the empirical loss over the perturbed set, knowledge distillation over feature maps can also be adopted for defense [50, 51]. More definitions and details could be found in a comprehensive survey [52].

2.5 Discussion on the Connection to Our Work

Compared to above work and other transfer learning studies, our work aims at providing a *generic descent direction estimation strategy* that improves the performance of regularization-based deep transfer learning. The intuition of **GrOD** is, per iteration during the learning procedure, re-estimating a new direction of loss descending that addresses the affect of regularizers while making the empirical loss reduction/minimization not hurt. In our work, we demonstrated the capacity of **GrOD** working with two most recent deep transfer learning regularizers–L^2-SP [2] and Knowledge distillation [3], which are based on two typical deep learning philosophies (i.e., constraining weights and feature maps respectively), using a wide range of transfer learning tasks. The consistent performance boosts with **GrOD** in all cases of experiments suggests that **GrOD** can improve above regularization-based deep transfer learning with higher accuracy.

Other techniques, including continual learning [20, 21], attention mechanism for CNN models [44, 53–55] and so on, can also improve the performance of knowledge transfer between tasks. We believe our work made complementary contributions in this area. All in all, we appreciate the contributions made by these studies. Furthermore, compared to the earlier version of this manuscript [1], we have made significant contributions to extend the previous work that primarily focuses on deep transfer learning, to improve the regularized deep learning in general cases. New regularized deep learning applications, such as knowledge distillation and adversarial learning, have been studied here. This manuscript includes our most recent efforts on improving deep transfer learning, adversarial learning and network distilling with **GrOD**, from both theoretical and empirical aspects. Additional experiments with new results have been provided to demonstrate our new findings.

3 GROD: GRADIENT ORTHOGONAL DECOMPOSITION

In this section, we formalize the technical details of our research, then present the design of our solution **GrOD**.

3.1 Definitions, Intuitions and Assumptions

Prior to presenting of the algorithms, this section introduces the settings of the problem.

Definition 2 (descent directions). *Gradient-based learning algorithms are frequently used for regularized deep learning to minimize the loss function listed in Eq. 1. In each iteration of learning procedure, the algorithms estimate a descent direction* $\mathbf{d}(\omega)$*, such as stochastic gradient, based on the optimization objective* ω *that approximates the gradient, such that*

$$\begin{aligned}\mathbf{d}(\omega) &\approx \nabla\mathcal{L}(\omega)\\ &= \sum_{i=1}^{n}\nabla L(z(\mathbf{x}_i, \omega), y_i) + \lambda\nabla\Omega(\omega)\\ &= \nabla J(\omega) + \lambda\cdot\nabla\Omega(\omega),\end{aligned} \tag{6}$$

where $\nabla J(\omega) = \sum_{i=1}^{n} \nabla L(z(\mathbf{x}_i, \omega), y_i)$ *refers to the gradient of empirical loss based on training set and* $\nabla\Omega(\omega)$ *is the gradient of regularization term all based on optimization objective* ω.

Following the above definition, we reduce our research problem as finding a new descent direction based on the gradients of the empirical loss and the regularizer term. The new descent direction is expected to preserve the effects of regularization, while avoiding to hurt the empirical loss minimization. Due to the affect of regularization $\Omega(\omega, \omega_s)$, the angle between the actual descent direction $\mathbf{d}(\omega)$ and the gradient of empirical loss $\nabla J(\omega)$, i.e., $\measuredangle(\mathbf{d}(\omega), \nabla J(\omega))$, would be large. It is intuitive to state that when $\measuredangle(\mathbf{d}(\omega), \nabla J(\omega)$ is large, the descent direction cannot effectively lower the empirical loss and causes the potential performance bottleneck of deep transfer learning. We thus formulate the technical problem with following assumptions specified.

Assumption 1 (Effective Empirical Loss Minimization). *It is reasonable to assume that the actual descent direction* $\widehat{\mathbf{d}}(\omega)$ *having a smaller angle with the gradient of empirical loss, i.e., a smaller* $\measuredangle(\widehat{\mathbf{d}}(\omega), \nabla J(\omega))$*, can lower the empirical loss more efficiently.*

Assumption 2 (Regularization Effect Preservation). *It is also reasonable to assume the actual descent direction* $\widehat{\mathbf{d}}(\omega)$ *having a smaller angle with the gradient of regualrizer's term, i.e., a smaller* $\measuredangle(\widehat{\mathbf{d}}(\omega), \nabla\Omega(\omega))$*, could strengthen the effects of regularization for deep learning.*

3.2 Problem Formulation

Based on above definitions and assumptions, in our research we propose a new direction descent algorithm—every iteration of the algorithm re-estimates a new descent direction $\widehat{\mathbf{d}}$ to effectively lower the training loss based on the optimization object ω while preserving the effect of regularizer $\Omega(\omega)$ (**Assumption 2**). Note that, to avoid the use of any threshold for bounding the two angles between $\widehat{\mathbf{d}}$ and $\nabla J(\omega)$ and between $\widehat{\mathbf{d}}$ and $\nabla Omega(\omega)$, we formulate the research problem as follows.

3.2.1 ERM-Effective descent direction. We formulate the research problem as finding an ERM-Effective Descent Direction as follow.

Definition 3 (ERM-Effective Descent Direction). *We define the ERM-Effective descent direction* $\mathbf{d}(\omega)$ *as a direction derived from the overall loss gradient* $\nabla\mathcal{L}(\omega)$ *and could descend the empirical loss* $J(\omega)$ *as fast as the one using the gradient of empirical loss* $\nabla J(\omega)$*. Such that*

$$\mathbf{d}(\omega) = \arg\min_{\mathbf{d}} \|\mathbf{d} - \nabla\mathcal{L}(\omega)\|_2^2 \quad s.t. \quad J(\omega - \epsilon\mathbf{d}) \leq J(\omega - \epsilon\nabla J(\omega)), \tag{7}$$

where ϵ *denotes the learning rate.*

Such ERM-effective descent direction $\mathbf{d}(\omega)$ can be estimated by solving the constrained optimization problem (7). Intuitively, optimization (7) aims at finding the descent direction which is close to the overall loss gradients and lowers the empirical loss no less than using the empirical loss gradient as the descent direction in the iteration with ω.

3.2.2 Low-Complexity ERM-Effective Descent Direction via Relaxed Constraint Programming. While the proposed descent direction straightforwardly meets our assumptions, the computation complexity to solve the constrained programming is high. Thus, we relax the constrained programming problem through the first-order approximation.

Assumption 3 (Relaxation with First-order Taylor Approximation). *For simplicity, we assume the loss function would enjoy a tight first-order approximation based on Taylor expansion, such that with* $\|\Delta\|_2^2$ *close to zero,* $J(\omega + \Delta) \approx J(\omega) + \langle\nabla J(\omega), \Delta\rangle + o(\|\Delta\|_2^2)$*,where* $\langle\cdot,\cdot\rangle$ *denotes the inner product. Thus, with a varnishing learning rate* ϵ *and any descent direction* $\mathbf{d}$*, there should have* $J(\omega - \epsilon\mathbf{d}) \approx J(\omega) - \langle\mathbf{d}, \nabla J(\omega)\rangle$.

Definition 4 (ERM-Effective Descent Direction via Relaxed Constraint Programming). *Based on above assumption, we can rewrite the problem in 7 into the relaxed constraint programming problem as follow.*

$$\mathbf{d}(\omega) = \arg\min_{\mathbf{d}} \ \|\mathbf{d} - \nabla\mathcal{L}(\omega)\|_2^2 \quad s.t. \quad \langle \mathbf{d}, \nabla J(\omega)\rangle \geq \|\nabla J(\omega)\|_2^2 \ . \tag{8}$$

In this work, we intend to solve the problem in (8) by orthogonal decomposition of regularization gradients $\nabla\Omega(\omega)$.

Algorithm 1 GrOD: descent direction Estimation

1: **procedure GrOD**($\mathbf{D}, \omega_\mathbf{t}, b, \lambda$)
2: /*Stochastic Gradients Estimations*/
3: $\mathbf{B}_t \sim \mathbf{D}$ sampling a mini-batch of b random samples from the training dataset
4: $\nabla\widehat{J}_t$ estimating stochastic gradient of $J(\omega)$ at the point $\omega_\mathbf{t}$ using the mini-batch $\mathbf{B}_t$
5: $\nabla\widehat{\Omega}_t$ estimating stochastic gradient of $\Omega_t(\omega)$ at the point $\omega_\mathbf{t}$ using the mini-batch $\mathbf{B}_t$
6: /*descent direction Correction*/
7: **if** $\measuredangle(\nabla\widehat{J}_t, \nabla\widehat{\Omega}_t) \leq 90°$ **then**
8: $\widehat{\mathbf{d}}_\mathbf{t} \leftarrow \nabla J_t + \lambda \cdot \nabla\widehat{\Omega}_t$
9: **else**
10: $\widehat{\mathbf{d}}_\mathbf{t} \leftarrow \nabla\hat{J}_t + \lambda \cdot [\nabla\hat{\Omega}_t - \frac{\langle \nabla\hat{J}_t, \nabla\hat{\Omega}_t\rangle}{\|\nabla\hat{J}_t\|_2^2} \cdot \nabla\hat{J}_t]$
11: **end if**
12: **return** $\widehat{\mathbf{d}}_\mathbf{t}$
13: **end procedure**

3.3 GrOD: Descent Direction Estimation via Orthogonal Decompositions

In this section, we presented the design of **GrOD** as a descent direction estimator that solves the relaxed constraint programming problem addressed in Section 3.1.2. Given the empirical loss function $J(\omega)$, the regularization term $\Omega(\omega)$, the set of training data $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, the mini-batch size b and the regularization coefficient λ, we propose to use Algorithm 1 to estimate the descent direction at the point ω_t for the t^{th} iteration of regularized deep learning.

With such descent direction estimator, the learning algorithm is capable of replacing the original stochastic gradient estimators used in stochastic gradient descent (SGD), Momentum and/or Adam for deep learning. Specifically, for each (e.g., the t^{th}) iteration of learning procedure, **GrOD** estimates the gradients of empirical loss and regularization term (i.e., $\nabla\widehat{J}_t$ and $\nabla\widehat{\Omega}_t$) separately as follows.

- *Acute Angle:* When the angle between gradients of empirical loss and regularization term is acute i.e., $\measuredangle(\nabla\widehat{J}_t, \nabla\widehat{\Omega}_t) \leq 90°$, **GrOD** uses the original stochastic gradient as the descent direction (such as line 8 in Algorithm 1). In such case, we believed the effect of regularization might not over−penalize the empirical loss minimization procedure.
- *Obtuse Angle:* On the other hand, when the angle is obtuse, **GrOD** decomposes the gradient of regularization term $\nabla\widehat{\Omega}_t$ to two orthogonal directions, where the first direction is orthogonal with the gradient of empirical loss while the second direction parallelizing with the empirical loss gradient (i.e., $\frac{\langle \nabla\hat{J}_t, \nabla\hat{\Omega}_t\rangle}{||\nabla\hat{J}_t||_2^2} \cdot \nabla\hat{J}_t$). **GrOD** truncates the direction against the gradient of empirical loss (i.e., $\nabla\hat{\Omega}_t - \frac{\langle \nabla\hat{J}_t, \nabla\hat{\Omega}_t\rangle}{||\nabla\hat{J}_t||_2^2} \cdot \nabla\hat{J}_t$), and further compose the orthogonal direction with gradient of empirical loss as the actual descent direction (as shown in Line 10. of Algorithm 1).

Note that the complexity of **GrOD** for descent direction estimation is low. Given the two gradient vectors $\nabla\widehat{J}_t$ and $\nabla\widehat{\Omega}_t$, **GrOD** uses Line 10 of Algorithm 1 to estimate the descent direction, where the inner-product of two vectors, scalar-vector-product, and some vector addition/subtraction operations are used. Thus the computational complexity of Line 10 of **GrOD** for descent direction estimation is $O(d)$, where d refers to the number of dimensions.

To understand the theoretical properties of **GrOD**, please refer to Section 3.4 for our analysis. For empirical validation, please refer to Section 4.5.2, where the experiment results validate the effect of the gradient orthogonal decomposition to the regularized deep learning.

3.4 Understanding the Effects of GrOD for Regularization

Based on the algorithm introduced in Algorithm 1, we can make lemmas as follow.

Lemma 1 (Acute Angle with Gain). *In the t^{th} iteration of* **GrOD**, *given (1) an positive regularizer's weight $\lambda > 0$, (2) the empirical loss gradient $\nabla\hat{J}_t$, (3) the regularizer's gradient $\nabla\widehat{\Omega}_t$, and (4) the actual descent direction $\widehat{\mathbf{d}}_\mathbf{t}$ computed by* **GrOD**, *the angle between the actual descent direction $\widehat{\mathbf{d}}_t$ and the empirical loss gradient $\nabla\hat{J}_t$ is acute, and the inner product of $\widehat{\mathbf{d}}_t$ and $\nabla\hat{J}_t$ is larger than the squared norm of $\nabla\hat{J}_t$, such that*

$$\left\langle \widehat{\mathbf{d}}_\mathbf{t}, \nabla\hat{J}_t \right\rangle \geq \left\| \nabla\hat{J}_t \right\|_2^2 . \tag{9}$$

Above lemma could be obtained using the proof as follow.

Proof. We prove above two lemmas in two cases

- When $\measuredangle(\nabla\widehat{\Omega}_t, \nabla\hat{J}_t) \leq 90^\circ$ (i.e., $\left\langle \nabla\widehat{\Omega}_t, \nabla\hat{J}_t \right\rangle \geq 0$), then $\widehat{\mathbf{d}}_\mathbf{t} = \nabla\widehat{\Omega}_t + \lambda \cdot \nabla\hat{J}_t$ and $\left\langle \widehat{\mathbf{d}}_\mathbf{t}, \nabla\hat{J}_t \right\rangle = \left\langle \nabla\hat{J}_t, \nabla\hat{J}_t \right\rangle + \lambda \cdot \left\langle \nabla\hat{J}_t, \nabla\widehat{\Omega}_t \right\rangle \geq \|\nabla\hat{J}_t\|_2^2 \geq 0$.
- Else when $\measuredangle(\nabla\widehat{\Omega}_t, \nabla\hat{J}_t) > 90^\circ$ ((i.e., $\left\langle \nabla\widehat{\Omega}_t, \nabla\hat{J}_t \right\rangle < 0$)), then $\widehat{\mathbf{d}}_\mathbf{t} = \nabla\hat{J}_t + \lambda \cdot [\nabla\hat{\Omega}_t - \frac{\langle \nabla\hat{J}_t, \nabla\hat{\Omega}_t \rangle}{\|\nabla\hat{J}_t\|_2^2} \cdot \nabla\hat{J}_t]$ and $\left\langle \widehat{\mathbf{d}}_\mathbf{t}, \nabla\hat{J}_t \right\rangle = \left\langle \nabla\hat{J}_t, \nabla\hat{J}_t \right\rangle + \lambda \cdot \left[\left\langle \nabla\hat{J}_t, \nabla\widehat{\Omega}_t \right\rangle - \frac{\langle \nabla\hat{J}_t, \nabla\hat{\Omega}_t \rangle}{\|\nabla\hat{J}_t\|_2^2} \cdot \left\langle \nabla\hat{J}_t, \nabla\hat{J}_t \right\rangle \right] = \|\nabla\hat{J}_t\|_2^2 \geq 0$.

In above two cases, there has $\left\langle \widehat{\mathbf{d}}_\mathbf{t}, \nabla\hat{J}_t \right\rangle \geq \|\nabla\hat{J}_t\|_2^2 \geq 0$ and thus $\measuredangle(\widehat{\mathbf{d}}_\mathbf{t}, \nabla\hat{J}_t) \leq 90^\circ$ (acute angle). □

Lemma 2 (Strengthened Descent Direction). *There has $\left\| \widehat{\mathbf{d}}_t \right\|_2^2 \geq \|\nabla\hat{J} + \lambda \cdot \nabla\widehat{\Omega}_t\|_2^2$ — i.e., the norm of* **GrOD** *descent direction is longer than the original loss gradient.*

Proof.
- When $\measuredangle(\nabla\widehat{\Omega}_t, \nabla\hat{J}_t) \leq 90^\circ$ (i.e., $\left\langle \nabla\widehat{\Omega}_t, \nabla\hat{J}_t \right\rangle \geq 0$), then $\widehat{\mathbf{d}}_\mathbf{t} = \nabla\widehat{\Omega}_t + \lambda \cdot \nabla\hat{J}_t$. Thus $\left\| \widehat{\mathbf{d}}_t \right\|_2^2 = \|\nabla\hat{J} + \lambda \cdot \nabla\widehat{\Omega}_t\|_2^2$.
- Else when $\measuredangle(\nabla\widehat{\Omega}_t, \nabla\hat{J}_t) > 90^\circ$ ((i.e., $\left\langle \nabla\widehat{\Omega}_t, \nabla\hat{J}_t \right\rangle < 0$)), then $\widehat{\mathbf{d}}_\mathbf{t} = \nabla\hat{J}_t + \lambda \cdot [\nabla\hat{\Omega}_t - \frac{\langle \nabla\hat{J}_t, \nabla\hat{\Omega}_t \rangle}{\|\nabla\hat{J}_t\|_2^2} \cdot \nabla\hat{J}_t]$. Let decompose $\widehat{\Omega}_t$ into two orthogonal vectors $\widehat{\Omega}_x = \frac{\left\langle \nabla\hat{J}_t, \widehat{\Omega}_t \right\rangle}{\|\nabla\hat{J}_t\|_2^2} \nabla\hat{J}_t$ and $\widehat{\Omega}_y = \widehat{\Omega}_t - \frac{\left\langle \nabla\hat{J}_t, \widehat{\Omega}_t \right\rangle}{\|\nabla\hat{J}_t\|_2^2} \nabla\hat{J}_t$

subject to the direction and the orthogonal direction of $\nabla\hat{J}_t$. Then we have

$$
\begin{aligned}
&\left\|\nabla\hat{J}+\lambda\cdot\nabla\widehat{\Omega}_t\right\|_2^2\\
&=\left\|\nabla\hat{J}+\lambda\cdot\nabla\widehat{\Omega}_x+\lambda\cdot\nabla\widehat{\Omega}_y\right\|_2^2\ ,\ \text{Consider the orthogonal directions}\\
&=\left\|\nabla\hat{J}+\lambda\cdot\nabla\widehat{\Omega}_x\right\|_2^2+\left\|\lambda\cdot\nabla\widehat{\Omega}_y\right\|_2^2\\
&=\left\|\left(1+\lambda\frac{\left\langle\nabla\hat{J}_t,\widehat{\Omega}_t\right\rangle}{\|\nabla\hat{J}_t\|_2^2}\right)\cdot\nabla\hat{J}_t\right\|_2^2+\left\|\lambda\cdot\nabla\widehat{\Omega}_y\right\|_2^2\ ,\ \text{as}\ \left\langle\nabla\widehat{\Omega}_t,\nabla\hat{J}_t\right\rangle<0\\
&\leq\left\|\nabla\hat{J}_t\right\|_2^2+\left\|\lambda\cdot\nabla\widehat{\Omega}_y\right\|_2^2\ ,\ \text{Consider the orthogonal directions}\\
&=\left\|\widehat{\mathbf{d}}_t\right\|_2^2
\end{aligned}
\tag{10}
$$

□

Finally, we could obtain our theoretical result as follow.

Proposition 1 (The **GrOD** descent direction is an ERM-Effective descent direction via Relaxed Constraint Programming). *We argue that in every t^{th} iteration, given an positive regualrizer's weight λ, suppose the empirical loss gradient $\nabla\hat{J}_t$ and the regularizer's gradient $\nabla\widehat{\Omega}_t$, the actual descent direction $\widehat{\mathbf{d}}_\mathbf{t}$ computed by* **GrOD** *should be a solution of problem (8) in Definition 4.*

Proof. Lemma 1 proves that the **GrOD** descent direction $\widehat{\mathbf{d}}_\mathbf{t}$ satisfies $\measuredangle(\widehat{\mathbf{d}}_\mathbf{t},\nabla\hat{J}_t)\leq 90^\circ$ — the constraint of problem (8) in Definition 4. Thus, here, we reduce our poof to test whether $\widehat{\mathbf{d}}_\mathbf{t}$ is a minimizer of $\|\mathbf{d}-(\nabla\hat{J}_t+\lambda\cdot\nabla\widehat{\Omega}_t)\|_2^2$ among all possible vectors satisfying the constraint. We test this proposition in following two cases.

- When $\measuredangle(\nabla\widehat{\Omega}_t,\nabla\hat{J}_t)\leq 90^\circ$, then $\widehat{\mathbf{d}}_t=\nabla\hat{J}_t+\lambda\cdot\nabla\widehat{\Omega}_t$ (as line 8 in Algorithm 1). Thus, $\|\widehat{\mathbf{d}}_t-(\nabla\hat{J}_t+\lambda\cdot\nabla\widehat{\Omega}_t)\|_2^2=0$ (minimal) in this case.
- Else when $\measuredangle(\nabla\widehat{\Omega}_t,\nabla\hat{J}_t)>90^\circ$, then $\widehat{\mathbf{d}}_\mathbf{t}=\nabla\hat{J}_t+\lambda\cdot[\nabla\hat{\Omega}_t-\frac{\langle\nabla\hat{J}_t,\nabla\hat{\Omega}_t\rangle}{\|\nabla\hat{J}_t\|_2^2}\cdot\nabla\hat{J}_t]$ (as line 10 in Algorithm 1). Let decompose $\widehat{\Omega}_t$ into two orthogonal vectors $\widehat{\Omega}_x=\frac{\left\langle\nabla\hat{J}_t,\widehat{\Omega}_t\right\rangle}{\|\nabla\hat{J}_t\|_2^2}\nabla\hat{J}_t$ and $\widehat{\Omega}_y=\widehat{\Omega}_t-\frac{\left\langle\nabla\hat{J}_t,\widehat{\Omega}_t\right\rangle}{\|\nabla\hat{J}_t\|_2^2}\nabla\hat{J}_t$ subject to the direction and the orthogonal direction of $\nabla\hat{J}_t$, and thus,

$$
\left\|\widehat{\mathbf{d}}_\mathbf{t}-(\nabla\hat{J}_t+\lambda\cdot\nabla\widehat{\Omega}_t)\right\|_2^2=\left\|\lambda\cdot\frac{\left\langle\nabla\hat{J}_t,\nabla\hat{\Omega}_t\right\rangle}{\|\nabla\hat{J}_t\|_2^2}\cdot\nabla\hat{J}_t\right\|_2^2=\lambda^2\left\|\widehat{\Omega}_x\right\|_2^2\ .
\tag{11}
$$

In the meanwhile, we can obtain an inequality as follow.

$$
\begin{aligned}
&\left\|\widehat{\mathbf{d}}_t - (\nabla \hat{J}_t + \lambda \cdot \nabla \widehat{\Omega}_t)\right\|_2^2 \text{ , Consider Lemma 2 and triangle} \\
\geq &\left\|\widehat{\mathbf{d}}_t\right\|_2^2 - \left\|\nabla \hat{J}_t + \lambda \cdot \nabla \widehat{\Omega}_t\right\|_2^2 \\
= &\left\|\widehat{\mathbf{d}}_t\right\|_2^2 - \left\|\nabla \hat{J}_t + \lambda \cdot \nabla \widehat{\Omega}_x + \lambda \cdot \nabla \widehat{\Omega}_y\right\|_2^2 \text{ , Consider } (\nabla \hat{J}_t + \lambda \cdot \nabla \Omega_x) \perp \nabla \widehat{\Omega}_y \\
= &\left\|\widehat{\mathbf{d}}_t\right\|_2^2 - \left(\left\|\nabla \hat{J}_t + \lambda \cdot \nabla \widehat{\Omega}_x\right\|_2^2 + \left\|\lambda \cdot \nabla \widehat{\Omega}_y\right\|_2^2\right) \text{ , Consider } \frac{\left\langle \nabla \hat{J}_t, \widehat{\Omega}_t \right\rangle}{\|\nabla \hat{J}_t\|_2^2} < 0 \text{ in this case} \\
= &\left\|\widehat{\mathbf{d}}_t\right\|_2^2 - \left(\left\|\nabla \hat{J}_t\right\|_2^2 - \left\|\lambda \cdot \nabla \widehat{\Omega}_x\right\|_2^2 + \left\|\lambda \cdot \nabla \widehat{\Omega}_y\right\|_2^2\right) \text{ , Consider } \nabla \hat{J}_t \perp \nabla \widehat{\Omega}_y \\
= &\left\|\widehat{\mathbf{d}}_t\right\|_2^2 - \left(\left\|\nabla \hat{J}_t + \lambda \cdot \nabla \widehat{\Omega}_y\right\|_2^2 - \left\|\lambda \cdot \nabla \widehat{\Omega}_x\right\|_2^2\right) \\
= &\lambda^2 \left\|\nabla \widehat{\Omega}_x\right\|_2^2
\end{aligned}
\tag{12}
$$

Consider (11) and (12), we can conclude that $\widehat{\mathbf{d}}_t$ is the solution of problem (8) while $\lambda^2 \left\|\nabla \widehat{\Omega}_x\right\|_2^2 = \left\{\min_{\mathbf{d}} \ \|\mathbf{d} - \nabla \mathcal{L}(\omega)\|_2^2 \ s.t. \ \langle \mathbf{d}, \nabla J(\omega) \rangle \geq \|\nabla J(\omega)\|_2^2\right\}$. □

In this way, we could conclude **GrOD** is the solution that we desire in problem (8). Furthermore, from the perspectives of descent directions, we also find that the behavior of **GrOD** is not achievable through tuning the weight of the regularizer alone.

Proposition 2 (**GrOD** is not achievable by tuning the weight of the regularizer). *In the t^{th} iteration of* **GrOD**, *given (1) any two positive weights of the regularizer $\forall \lambda_1, \ \lambda_2 > 0$ for* **GrOD** *and the vanilla loss for regularized deep learning respectively, (2) the empirical loss gradient $\nabla \hat{J}_t$ with $\|\nabla \hat{J}_t\|_2^2 > 0$, and (3) the regularizer's gradient $\nabla \widehat{\Omega}_t$ with with $\|\nabla \widehat{\Omega}_t\|_2^2 > 0$, we denote the* **GrOD** *descent direction based on λ_1 as $\widehat{\mathbf{d}}_t(\lambda_1)$ and the vanilla regularized loss as $\nabla \hat{J}_t + \lambda_2 \cdot \nabla \widehat{\Omega}_t$. We argue that, when $\measuredangle(\nabla \hat{J}_t, \nabla \widehat{\Omega}_t) > 90^\circ$, for any positive weights λ_1 and λ_2, there has*

$$
\widehat{\mathbf{d}}_t(\lambda_1) \neq \nabla \hat{J}_t + \lambda_2 \cdot \nabla \widehat{\Omega}_t \ . \tag{13}
$$

In this way, we say the descent direction of **GrOD** *is not achievable by vanilla gradient of loss for regularized deep learning for any pair of weights for regularizers.*

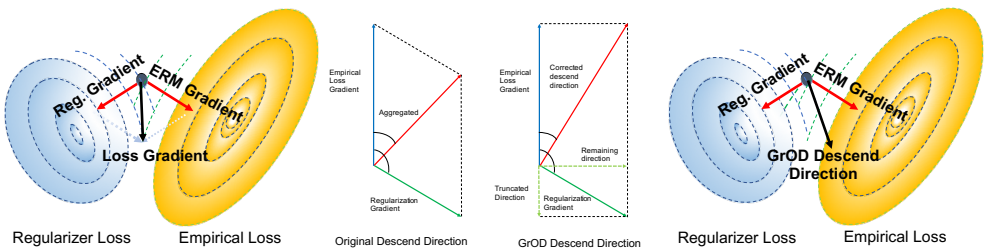
Proof. Let decompose $\widehat{\Omega}_t$ into two orthogonal vectors $\widehat{\Omega}_x = \frac{\left\langle \nabla \hat{J}_t, \widehat{\Omega}_t \right\rangle}{\|\nabla \hat{J}_t\|_2^2} \nabla \hat{J}_t$ and $\widehat{\Omega}_y = \widehat{\Omega}_t - \frac{\left\langle \nabla \hat{J}_t, \widehat{\Omega}_t \right\rangle}{\|\nabla \hat{J}_t\|_2^2} \nabla \hat{J}_t$ subject to the direction and the orthogonal direction of $\nabla \hat{J}_t$. Since $\measuredangle(\nabla \hat{J}_t, \nabla \widehat{\Omega}_t) > 90^\circ$ and $\|\hat{J}_t\|_2^2 > 0$ and $\|\widehat{\Omega}_t\|_2^2 > 0$, there thus has $\left\langle \nabla \hat{J}_t, \widehat{\Omega}_t \right\rangle < 0$ and $\|\widehat{\Omega}_x\|_2^2 > 0$. Given any two positive

weights $\forall \lambda_1,\ \lambda_2 > 0$ we can obtain the inequality as follow.

$$
\begin{aligned}
&\left\| \widehat{\mathbf{d}}_t(\lambda_1) - (\nabla \hat{J}_t + \lambda_2 \cdot \nabla \widehat{\Omega}_t) \right\|_2^2 ,\ \text{Consider } \measuredangle(\nabla \hat{J}_t, \nabla \widehat{\Omega}_t) > 90^\circ \\
&= \left\| \lambda_1 \cdot (\nabla \hat{\Omega}_t - \frac{\langle \nabla \hat{J}_t, \nabla \hat{\Omega}_t \rangle}{\|\nabla \hat{J}_t\|_2^2} \cdot \nabla \hat{J}_t) - \lambda_2 \cdot \nabla \widehat{\Omega}_t \right\|_2^2 \\
&= \left\| (\lambda_1 - \lambda_2) \cdot (\nabla \hat{\Omega}_x + \nabla \hat{\Omega}_y) - \lambda_1 \cdot \widehat{\Omega}_x \right\|_2^2 \\
&= \left\| (\lambda_1 - \lambda_2) \cdot \nabla \widehat{\Omega}_y - \lambda_2 \cdot \nabla \widehat{\Omega}_x \right\|_2^2 ,\ \text{Consider } \nabla \widehat{\Omega}_x \perp \nabla \widehat{\Omega}_y \\
&= \left\| (\lambda_1 - \lambda_2) \cdot \nabla \widehat{\Omega}_y \right\|_2^2 + \left\| \lambda_2 \cdot \nabla \widehat{\Omega}_x \right\|_2^2 \\
&> 0.
\end{aligned}
\tag{14}
$$

In this way, we can conclude that $\widehat{\mathbf{d}}_t(\lambda_1) \neq \nabla \hat{J}_t + \lambda_2 \cdot \nabla \widehat{\Omega}_t$, for $\forall \lambda_1,\ \lambda > 0$, when $\measuredangle(\nabla \hat{J}_t, \nabla \widehat{\Omega}_t) > 90^\circ$, $\|\hat{J}_t\|_2^2 > 0$ and $\|\widehat{\Omega}_t\|_2^2 > 0$. □

To interpret the theoretical results, we use an example to visualize our intuition. Figure 2 illustrates an example of **GrOD** descent direction estimation, when the angles between gradients of empirical loss and regularization term is obtuse ($> 90^\circ$). As shown in Figure 2 (a), the effect of regularization term forms a direction that might slow down the empirical loss descending. As shown in Figure 2 (b), **GrOD** decomposes the gradient of regularization term and truncates the conflicted direction for the actual descent direction estimation. On the other hand, the angle between the actual descent direction and regularization gradient and the angle between the actual descent direction and empirical loss gradient are both acute ($\leq 90^\circ$), so as to secure the regularization effect while ensuring empirical loss descending. In this way, we can understand **GrOD** as the solution to the relaxed constraint programming problem for ERM-preserved descent direction estimation addressed in Section 3.2. Furthermore, in this example, due to the truncated direction of the regularizer's gradient, the **GrOD** descent direction cannot be achieved by any linear combinations of ERM loss gradient and the regularizer's gradient.



(a) Original Gradient (b) Orthogonal Decomposition of the Regularizer's Gradients (c) Descent Direction of **GrOD**

Fig. 2. Example of **GrOD** descent direction Estimation

4 EXPERIMENT

In this section, we report our experiment results for **GrOD** with three types of regularized deep learning paradigms, i.e., L^2-SP for transfer learning [2], knowledge distillation [3], and adversarial training [46].

4.1 Data Set and Experiment Setups

In transfer learning and knowledge distillation experiments we used the ResNet-18 [56] as our base model with three widely-used source datasets including ImageNet [8], Places 365 [57], and Stanford Dogs 120 [58] for weights pre-training. To evaluate the performance of transfer learning, we selected four datasets as the target data sets. These are Caltech 256 [5], MIT Indoors 67 [6], Flowers 102 [59] and CIFAR-10 [7]. Note that, we follow the same settings used in [2] for Caltech 256 setup, where 30 or 60 samples randomly drawn from each category for training with 20 remaining samples for testing. In adversarial training experiments we used a small model consisting of two cnn layers and two fully connected layers with Fashion MNIST [60], while the ResNet-18 with CIFAR-10. Table 1 presents the statistics on some basic facts of all the datasets used in experiments.

4.1.1 Source/Target Tasks Pairing. Above configuration leads to 15 source/target task pairs, where regularization would hurt the performance of transfer learning in some of these cases. For example, the image contents of ImageNet and CIFAR-10 are quite similar, in this way, the knowledge transfer from ImageNet to CIFAR-10 could improve the performance. On the other hand, the images in Stanford Dog 120 and MIT Indoor 67 are quite different, e.g., dogs v.s. indoor scenes; then the regularization based on pre-trained weights of Stanford Dog 120 task would hurt the learning of MIT Indoor 67 task.

4.1.2 Pre-trained Models and Weights. Furthermore, to obtain the pre-trained weights of all source tasks, we adopt the pre-trained models of ImageNet [1], Place 365 [2], and Stanford Dog 120[3] released online. We found an interesting fact that the pre-trained models of Place 365 and Stanford Dog 120 were trained from the pre-trained model of ImageNet. In this way, the pre-trained models for Place 365 and Stanford Dog 120 have been already enhanced by the ImageNet.

4.1.3 Image Classification Tasks Setups. In transfer learning and knowledge distillation task all images are re-sized to 256×256 and re-scaled to $[-2, 2]$ for each channel, following with data augmentation operations of random mirror and random crop to 224×224. We use a batch size of 64, SGD with the momentum of 0.9 is used for optimizing all models [61]. The learning rate for base model starts with 0.01 and is divided by 10 after 6,000 iterations. The Training is terminated with 8,000 iterations for Caltech 256, MIT Indoor 67 and Flowers 102, terminates with 20,000 iterations for CIFAR-10 (i.e., 18 epochs). The pre-trained weights obtained from the source task were not only used as the initialization of the model, i.e., starting point of optimization. Under the best configuration, each experiment is repeated five times. We report the average accuracy with standard deviations. In adversarial training task, all images are resized to 224×224 and re-scaled to $[-2, 2]$, with random horizontal flips.

4.1.4 Hyper-parameter Tuning for Regularizer Weights. The regularizer weights for all experiments have been tuned best using cross validation or follow the default settings from the officially release codes of models. Our latter experiments would show that, with the same hyper-parameter settings, **GrOD** does not always outperform the overall loss gradients descent. To compare with varying

[1]https://github.com/pytorch/vision/tree/master/torchvision/models
[2]https://github.com/CSAILVision/places365
[3]https://github.com/stormy-ua/dog-breeds-classification

Table 1. Statistics on Datasets

Datasets	Domains	# Train/Test
ImageNet	Visual objects	1,419K+/100K
Place 365	Indoor scenes	10,000K+
Stanford Dog 120	Dogs	12K/8.5K
CIFAR-10	Visual objects	50K/10K
Caltech 256	Visual objects	30K+
MIT Indoors 67	Indoor scenes	5K+/1K+
Flowers 102	Flowers	1K+/6K+
Fashion-Mnist	Clothes	50K/60K

hyper-parameters, our experiment results addressed in Section 4.4 will demonstrate the effectiveness of **GrOD** that outperforms common regularized deep learning dominantly for adversarial learning with varying regularizer weights.

4.2 Performance of GrOD on Transfer Learning with L^2-SP [2]

In this section, we report the results of overall performance comparison based on the above tasks using L^2-SP [2] and its variant based on **GrOD** for knowledge transfer from pre−trained models. We primarily focus on evaluating the performance improvement contributed by **GrOD** on top of L^2-SP, comparing to the vanilla implementations. Both source and target tasks are trained on a typical ResNet-18 architecture. The knowledge transfer from ImageNet to all target tasks seems all good, as ImageNet contains more than 1000 classes of images with more categories covered and rich features offered. However, the performance of knowledge transfer from Stanford Dog 120 to MIT Indoor 67 might be quite limited or even negatively affected the learning procedure, as these two datasets contain quite different images—dogs v.s. indoor scenes. Further discussion on the negative transfer effects would be addressed in Section 4.2.2.

4.2.1 Overall Comparison. We present accuracy of all source/target pairs in Table 2. **GrOD** improves the performance of deep transfer learning in all of the above cases. For example, for the CIFAR-10 (target task) with ImageNet (source task), L^2-SP algorithm achieved 93.30% accuracy, while **GrOD** (L^2-SP) has improved the accuracy to 96.41% (with more than 3.1% accuracy improvement). To the best of our knowledge, it has the best known result [62] for CIFAR-10 training on ResNet-18 from ImageNet sources with only 18 epochs. Even, using Stanford Dog 120 as the source task can perform similar as the ones sourcing from ImageNet, since the pre−trained model of Standford Dog 120 was pre−pre−trained from ImageNet. Overall **GrOD** significantly improves the performance of L^2-SP in all transfer learning settings that we evaluated.

An interesting facts observed in the experiments is that, on top of the both algorithms and 15 source/task pairs, using Stanford Dog 120 as the source task can perform similar as the ones sourcing from ImageNet. We consider it is due to the reason that the public release of Stanford Dog 120 pre-trained model is pre-trained from ImageNet, while the size of Stanford Dog 120 dataset is relatively small (i.e., it cannot "wash out" the knowledge obtained from ImageNet while preserving the knowledge from the both ImageNet/Stanford Dog 120 datasets). In this way, knowledge transferring from Stanford Dog 120 can be as good as those based on ImageNet. In the meanwhile, **GrOD** can still improve the performance of L^2-SP, gaining 0.12%∼2.2% higher accuracy with low variance, even given the well−trained Stanford Dog 120 model.

Table 2. Accuracy Comparison on Knowledge Transfer from Different Source Datasets

	Caltech 256	MIT Indoors 67	Flowers 102	CIFAR-10
Source Dataset **ImageNet** [8]				
Fine-Tune [37]	82.68 ± 0.20	76.73 ± 0.77	90.24 ± 0.31	96.40 ± 0.4
L^2-SP [2]	83.69 ± 0.09	75.11 ± 0.43	88.96 ± 0.21	93.30 ± 0.16
GrOD +L^2-SP [2]	**84.14 ± 0.08**	**77.46 ± 0.29**	**90.68 ± 0.31**	**96.41 ± 0.11**
Source Dataset **Places 365** [57]				
Fine-Tune [37]	73.13 ± 0.20	82.64 ± 0.16	83.77 ± 0.68	89.35 ± 0.59
L^2-SP [2]	66.99 ± 0.20	84.09 ± 0.09	77.66 ± 0.13	89.78 ± 0.05
GrOD + L^2-SP [2]	**73.32 ± 0.1**	84.09 ± 0.09	**84.11 ± 0.06**	**90.85 ± 0.11**
Source Dataset **Stanford Dogs 120** [58]				
Fine-Tune [37]	82.29 ± 0.04	75.69 ± 0.21	90.20 ± 0.39	96.34 ± 0.13
L^2-SP [2]	83.44 ± 0.23	74.64 ± 0.07	88.14 ± 0.06	94.16 ± 0.10
GrOD + L^2-SP [2]	**83.84 ± 0.08**	**76.46 ± 0.22**	**89.98 ± 0.04**	**96.39 ± 0.08**

4.2.2 Performance with Negative Transfer Effect. According to the results presented in Tables 2, we find negative transfer may happen in the cross-domain cases "Visual Objects/Dogs ⇔ Indoor Scenes" (please refer to the domain definitions in Table 1), while **GrOD** can improve the performance of L^2-SP to relieve such negative effects. Two detailed cases are addressed as follow.

- **Cases of Negative Transfer** For both L^2-SP algorithms, when using ImageNet and Stanford Dogs 120 as the source task while transferring to MIT Indoors 67 as the target task, we can observe significant performance degradation comparing to knowledge transfer from Place 365 to MIT Indoor 67. For example (**Case I**), the accuracy of MIT Indoor 67 using L^2-SP is 84.09% based on pre-trained weights of Place 365, while the accuracy would be degraded to 75.11% and 74.64% under the same settings with ImageNet and Stanford Dog 120 as the pre-trained models respectively. Furthermore, we also observe the similar negative transfer effects, when using Place 365 as source while transfer to the target tasks based on Caltech 256, Flower 102 and CIFAR-10. For example (**Case II**), the accuracy on Flowers 102 is 77.66% using Place 365 as source, while sourcing from ImageNet and Stanford Dog can achieve as high as 88.96% and 88.14% respectively, all based on L^2-SP.
- **Relieving Negative Transfer Effects.** We believe performance degradation appeared in **Cases I** and **II** is due to the negative transfer, as the domains of these datasets are quite different. **GrOD** can however relieve such negative transfer cases. **GrOD**+L^2-SP [2] can achieve 84.11% on Flowers 102 dataset even when sourcing from Place 365 — i.e., achieving 7% accuracy improvement, comparing to vanilla L^2-SP under the same settings. For the rest negative transfer cases, **GrOD** can still improve the performance, with around 2% higher accuracy, comparing to the vanilla implementations of L^2-SP. In this way, we conclude that **GrOD** can improve the performance of L^2-SP in negative transfer cases with higher accuracy.

Note that we don't intend to claim that **GrOD** could eliminate the negative transfer effects in parts. It, however, improves the performance of regularization-based deep transfer learning, even with inappropriate source/target pairs. Such accuracy improvement can marginally solve the problem of negative transfer effects.

4.3 Performance of GrOD on Feature-wise Knowledge Distillation with [3]

We report the results of overall performance comparison based on the aforementioned tasks using Feature-wise Knowledge Distillation [3, 22] and its variant based on **GrOD** for Teacher–Student

Table 3. Classification Accuracy Comparison for Knowledge Distillation from Teacher Networks Pre-Trained by Various Datasets

	Caltech 256	MIT Indoors 67	Flowers 102	CIFAR-10
Distilling a Teacher Network Pre-trained by **ImageNet** [8]				
KnowDist [3]	82.93 ± 0.08	78.05 ± 0.32	90.43 ± 0.4	96.43 ± 0.08
GrOD +KnowDist [3]	**83.27 ± 0.4**	**78.77 ± 0.31**	**90.91 ± 0.4**	**96.57 ± 0.2**
Distilling a Teacher Network Pre-trained by **Places 365** [57]				
KnowDist [3]	72.8 ± 0.22	83.29 ± 0.42	83.50 ± 0.26	94.96 ± 0.05
GrOD + KnowDist [3]	**73.18 ± 0.24**	**84.40 ± 0.41**	**84.12 ± 0.56**	**95.02 ± 0.13**
Distilling a Teacher Network Pre-trained by **Stanford Dogs 120** [58]				
KnowDist [3]	82.73 ± 0.26	76.36 ± 0.19	89.86 ± 0.07	96.11 ± 0.53
GrOD + KnowDist [3]	**82.85 ± 0.27**	**76.74 ± 0.26**	**90.29 ± 0.34**	**96.41 ± 0.18**

training of deep neural networks. We also focus on evaluating the performance improvement contributed by **GrOD** on top of Knowledge Distillation (denoted as "KnowDist" in the paper), comparing to [3]. We use a ResNet-18 pre–trained on ImageNet as the Teacher Network.

We present the overall accuracy comparisons in Table 3. **GrOD** improves the performance of Teacher–Student training in all Student networks (also ResNet-18) training. For example, to train a Student network using CIFAR-10 dataset, common Knowledge distillation achieves 96.43% accuracy while **GrOD** further improves the accuracy to 96.57%. These two numbers are quite closed to the state of the art performance of ResNet-18 on CIFAR-10 datasets (without using additional training or data augmentation methods) [62]. As it has been shown in 3, GrOD brings significant improvement in all tasks. We test the performance of **GrOD** based on other Teacher networks (based on different datasets). **GrOD** achieves performance improvement in all cases, on top of [3].

4.4 Performance of GrOD on Adversarial Learning with Advt [4]

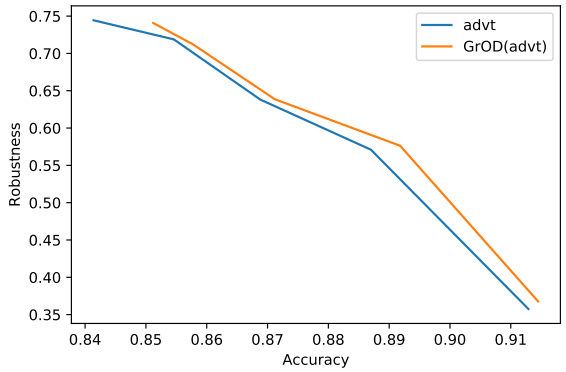
In this section, we report the results of performance comparison based on Fashion MNIST and CIFAR-10 under, adversarial learning settings, using advt [4] and its variant based on **GrOD**. We also focus on evaluating the performance improvement contributed by **GrOD** on top of advt. We use a simple two–layer CNN [4] and a ResNet-18 for this experiments.

4.4.1 Adversarial Learning Setups with **GrOD**. The experiment setups for adversarial learning with **GrOD** are a bit different from previous settings. [46] found training with an adversarial objective function with regularization in Eq (4). To generate the perturbation more efficiently, [4] provides the state of the art of adversarial learning that uses projected gradient descent(PGD, [47]) to generate adversarial examples, where two key factors ϵ and λ control the level of noise in adversarial attacks and strength of regularization affects to the adversarial learning.

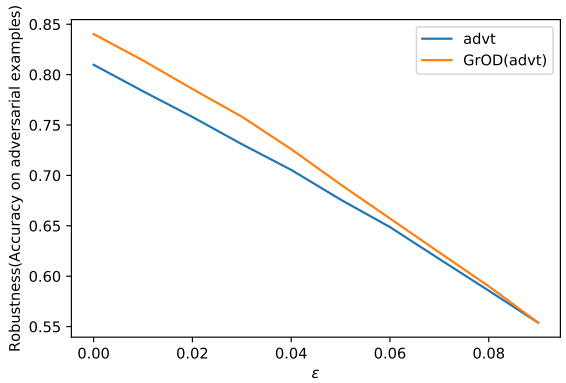
In this way, one can make trade–off between accuracy and robustness of the model through tuning the regularization coefficient λ and attacking strength ϵ. We can further adopt **GrOD** to get better accuracy while reserve robustness. Note that we model the gradient of regularization term as the $\nabla\Omega(\omega) = \frac{1}{n}\sum_{i=1}^{n}\nabla_\omega L(z(\boldsymbol{x}_i,\omega),y_i) - \frac{1}{n}\sum_{i=1}^{n}\nabla_\omega L(z(\boldsymbol{x}_i,\omega),y_i)$ to remove the major component in parallel with the original loss.

4.4.2 Experimental Results. We tested the adversarial training with **GrOD**on Fashion-MNIST [60] and CIFAR-10 respectively. All the images' are re-scaled to $[-2, 2]$. All adversarial examples are generated via [4] with 7 steps.

[4]https://github.com/ashmeet13/FashionMNIST-CNN



(a) Fashion MNIST with varying regularizer weight λ



(b) CIFAR-10 with varying perturbation size ϵ

Fig. 3. Overall Performance of Adversarial Learning with **GrOD**: **(a)** Robustness **vs** Accuracy of the models on Fashion MNIST datasets with varying λ (regularizer weight) and fixed perturbation size $\epsilon = 0.05$; **(b)** Robustness of the model trained on CIFAR-10 with varying perturbation size $\epsilon \in [0, 0.1]$

For Fashion MNIST dataset, we set step size for noise $\epsilon = 0.05$ with varying regulariztion coefficient λ, so as to see whether **GrOD** can improve the performance of advt [4] with enhanced robustness (i.e., the accuracy based on adversarial samples) and accuracy (i.e., the accuracy based on original testing samples). The results show that **GrOD** can achieve "Pareto–improvement" on top of advt. When, both algorithms achieve the same accuracy, **GrOD** leads to higher robustness. When they behave with same robustness, advt(**GrOD**) outperforms advt with higher accuracy. Note that, without adversarial attack, the evaluated CNN can obtain 0.92 accuracy on testing sets.

For the experiments based on the CIFAR-10 dataset, we fix the regularization coefficient while varying the step size ϵ for adversarial attack from 0 to 0.1. The experiment shows that advt(**GrOD**) can always outperform advt under the same level of noise (perturbation) for adversarial attacks. Generally, experiments based on both datasets demonstrated significant improvement of **GrOD** in adversarial learning tasks on top of state of the art [4].

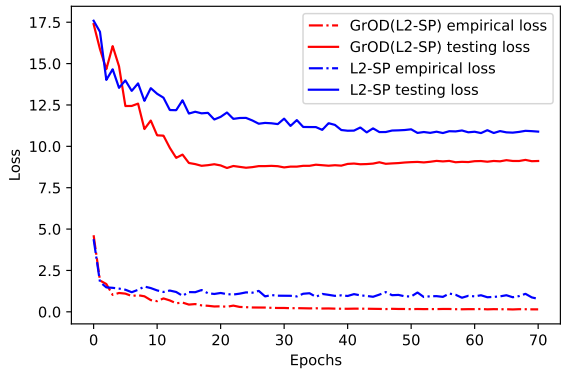


Fig. 4. Empirical Loss Minimization

4.5 Case Studies

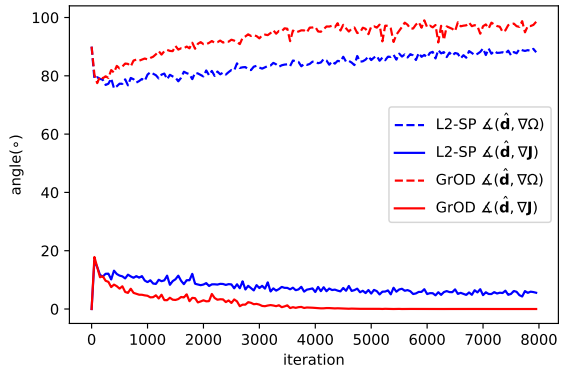
We report the results of the following two case studies that provide further evidences supporting that **GrOD** works in the way that we assumed.

4.5.1 Empirical Loss Minimization. As was elaborated in the Introduction section, we suspect that using regularizer might restrict the learning procedure from lowering the empirical loss. Such restriction helps the regularized deep learning to avoid over-fitting, but in the meanwhile, hurts the learning procedure. Following the insight we hope to study trends of empirical loss part minimization with and without **GrOD** in L^2-SP [2] case. Note that the empirical loss here is NOT the training loss, it refers to the data fitting error part of the training loss.

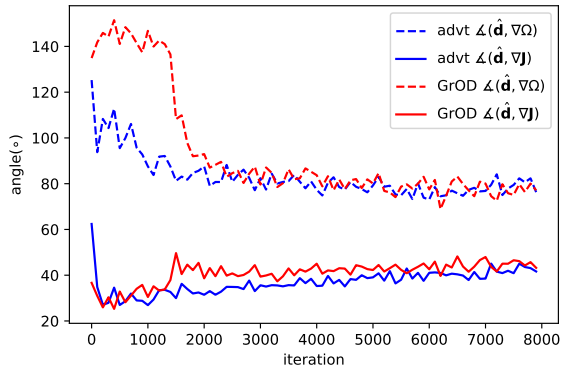
Figure 4 illustrates the trends of both empirical loss and testing loss, with increasing number of iterations, based on both L^2-SP and **GrOD**(L^2-SP), for Places 365 $\Longrightarrow$ MIT Indoors 67 case. As was expected, the empirical loss of both vanilla L^2-SP and **GrOD**(L^2-SP) reduces with the number of iterations, while the empirical loss of L^2-SP is always higher than that of **GrOD**(L^2-SP). In the meanwhile, **GrOD**(L^2-SP) always enjoys a lower testing loss than vanilla L^2-SP. The phenomena indicates that, comparing vanilla L^2-SP to **GrOD**(L^2-SP), the L^2-SP regularization term would restrict the procedure of empirical loss minimization and finally hurt the learning procedure with lower testing accuracy. Furthermore, we also observed that **GrOD** could be further improved through early stopping.

4.5.2 Angles Between Descent Directions. The intuition of **GrOD** design is based on the two assumptions made in section 3.2— it is possible to find a new descent direction that is very closed to the direction of empirical loss gradient (**Assumption 1**), while always shares a angle with the gradient of regularization term as small as the original descent direction (**Assumption 2**).

Figure 5 plots the comparison of the two types of angles with the L^2-SP and advt algorithms with and without **GrOD**. The results showed that with **GrOD** both algorithms always enjoy a smaller angle between the actual descent direction and the (stochastic) gradient of empirical loss i.e.,$\measuredangle(\widehat{\mathbf{d}}(\omega), \nabla J(\omega))$ of both algorithms with **GrOD** is smaller then the vanilla ones. We thus confirm the validity of Assumption 1. To demonstrate the validity of Assumption 2., we measure $\measuredangle(\widehat{\mathbf{d}}(\omega), \nabla\Omega(\omega))$ for the two cases using L^2-SP and advt algorithms on CIFAR-10. It shows that no matter whether **GrOD** is used, the trends of angles over the number of iterations are quite similar for the same algorithm under the same settings. Please note that values of angles highly depend on



(a) Knowledge Transfer: L^2-SP [2] vs. **GrOD**+L^2-SP



(b) Adversarial Training: Advt [4] vs. **GrOD**+ Advt

Fig. 5. $\angle(\widehat{\mathbf{d}}(\omega_t), \nabla\Omega(\omega_t))$ **vs.** $\angle(\widehat{\mathbf{d}}(\omega_t), \nabla J(\omega_t))$ over the Number of Iterations t with varying ω_t: **Blue line** represents vanilla implementations of L^2-SP and advt, **Red line** represents the **GrOD**-based solutions; Dash line represents $\angle(\widehat{\mathbf{d}}(\omega_t), \nabla\Omega(\omega_t))$, Solid line represents $\angle(\widehat{\mathbf{d}}(\omega_t), \nabla J(\omega_t))$.

the choice of hyperparameters (e.g., λ for L^2-SP). However, we still can verify that, by design, the angles between the **GrOD**'s actual descent direction and the empirical loss's gradient are always acute.

5 DISCUSSION

In this paper, we proposed **GrOD**, which can improve regularized deep learning through orthogonal decomposition of loss gradients. We have included extensive experiments using regularized learning paradigms [2–4] for knowledge transfer, knowledge distillation, and adversarial training. In this section, we discuss several open issues in this work.

5.1 Performance Improvement of GrOD and Analysis

All in all, the performance improvement made by **GrOD** on top of L^2-SP [2], KnowDist [3], and Adversarial Learning [4] is quite marginal. However, we hope to point out that the performance improvements consistently exist in all cases, especially relieving the "negative transfer" where the use of regularizer hurts the transfer learning. Please note that, in our study, we include the experiments based on inappropriate pairs of source and target datasets/pre-traind models for knowledge transfer and knowledge distillation (e.g., in Tables 2 and 3) while most of existing works uses ImageNet as the source datasets or pre-trained models only.

With inappropriate pairs of source and target datasets/pre-traind models, the regularized learning with L^2-SP or Distll might hurts the performance compared to directly fine-tuning from pre-trained weights. In all negative transfer cases, **GrOD** improves the performance of regularized knowledge transfer or knowledge distillation while always achieving performance better than vanilla fine-tuning. Furthermore, for the adversarial training with Advt regularization [4], **GrOD** achieves better trade-off between accuracy and robustness, with Pareto dominance in these two factors, under varying strength of perturbation. Again our theoretical analysis in Section 3.4 clearly states how **GrOD** could ensure the effectiveness of ERM learning procedure while preserving the regularization effects (Proposition 1), which solicits the performance improvement. Note that our theoretical analysis relies on two assumptions made in Section 3.1, we conducted case studies with experiments to validate these two assumptions empirically.

5.2 Stability of GrOD Performance

Though **GrOD** enjoys higher accuracy on average, in some cases, it also incorporates higher variances. For example, in Table 2, with weights pre-trained by ImageNet, **GrOD**(L^2-SP) achieves 90.86% with 0.31% STD for the target task based on Flower 102, while L^2-SP achieves 88.96% with 0.21% STD in the same settings. It is obvious that **GrOD** incorporates with higher variances, however the lower bound of confidence intervals of **GrOD** is still higher than the upper bound of confidence interval of the original algorithm. Further, we also tried to hack the weight decay using **GrOD**, the results showed that **GrOD** cannot improve weight decay (Note that the weight decay, i.e., the L^2-regularization, has been frequently considered as a stabilizer [63, 64] of the training procedure in a regularization of Ridge-style.). We believe both of these observations are due to the use of orthogonal decomposition on stochastic gradient. In practical deep learning, stochastic gradients – the noisy evaluation of loss functions' derivatives, have been used to accelerate the learning procedure with mini-batch sampling. However, the gradient noise [65] after orthogonal decomposition might perturbate the training procedure and leads to instability.

5.3 Hyper-parameters Tuning and Fair Comparisons

In our experiments, to enable fair comparisons, we use hyper-parameters, including learning rates and the weights of regularizers, according to the default settings released from the open-source implementation of the algorithms [2–4] (most of which were tuned best through cross validations in their research). In the same set of experiments, both **GrOD** and the original algorithms used the same set of hyper-parameters, especially the weights of regularizers for fair comparisons. Note that the performance of **GrOD** could be further improved through tuning the hyper-parameters (rather than the use of default ones).

Furthermore, our theoretical analysis also shows that the descent direction of **GrOD** is not achievable through tuning the weight of regularizers' term (Proposition 2). That means, no matter how one sets the hyper-parameters for vanilla regularized deep learning, the algorithm based on the vanilla loss gradient of regularized deep learning could not behave as same as **GrOD**.

5.4 Connections to Optimization Algorithms

Note that **GrOD** strategy is derived from the common stochastic gradient estimation used in stochastic gradient based learning algorithms, such as SGD, Momentum, conditioned SGD, Adam and so on. It can be consider as an alternative approach for descent direction estimation on top of vanilla stochastic gradient estimation, where you can still use natural gradient-alike method to condition the descent direction or adopt Momentum-alike acceleration methods to replace the weight updating mechanism. We are not intending to compare **GrOD** with any gradient-based learning algorithms, as the contributions are complementary. One can freely use **GrOD** to improve any gradient-based optimization algorithms with the descent direction corrected.

Furthermore, according to the ERM-Effective descent direction assumption, **GrOD** can furhter lower the empirical loss while preserving regularization effects simultaneously, as the finally descent direction will be close to the both empirical loss gradient and regularizer gradient. Our later on experiment based on adversarial learning will show that no matter how regularizer weights are fine-tuned, **GrOD** can still outperform the traditional regularized deep learning algorithms that linearly combine the gradients of the two terms as the descent direction. In future work, we intend to study the asymptotic properties and convergence performance of **GrOD**, using Neural Tangent Kernel as the proxy [66] to lower the complexity in non-convex optimization analysis.

5.5 Improving Advanced Regularization Methods

Please be advised that the regularized deep learning algorithms for transfer learning (L^2-SP) [2], knowledge distillation [3], and adversarial training [4] are not the state-of-the-art algorithms in the fields. In future work we are interested in incorporating with more advanced methods, such as DELTA [24], BSS [67], Co-Tuning [68], and learning without forgetting [21], where more advanced and complicated regularizers have been proposed incorporating constrained features, singular value decomposition, category relationship, and so on.

6 CONCLUSIONS

In this paper, we studied a descent direction estimation strategy **GrOD** that improves the common regularized deep learning techniques with applications to transfer learning [2], knowledge distillation [3], and adversarial learning [4]. Significant improvements have been observed compared to the existing methods that simply aggregates empirical loss for data fitting and regularization terms through linear combination, such as [2–4].

Specifically, we designed a new method to re-estimate a new direction for loss descending based on the (stochastic) gradient estimation of empirical loss and regularizers, where orthogonal decomposition has been made on the gradient of regularization terms, so as to eliminate the conflicted direction against the empirical loss descending. The design of the algorithm is based on an intuitive assumption made by us, namely *ERM-preserved descent direction*, where in the every iteration of the learning procedure, the empirical loss of regularized deep learning is expected to descend as fast as the one based on empirical loss minimization. We have conducted extensive experiments to evaluate **GrOD** using several real-world datasets based on classical convolutional neural networks. The experiment results and comparisons show that **GrOD** significantly improves the state-of-the-art algorithms for the three applications with higher accuracy and robustness.

REFERENCES

[1] Ruosi Wan, Haoyi Xiong, Xingjian Li, Zhanxing Zhu, and Jun Huan. Towards making deep transfer learning never hurt. In *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM'19)*. IEEE, 2019.

[2] Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. *Thirty-fifth International Conference on Machine Learning*, 2018.

[3] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[4] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[5] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[6] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009.

[7] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[9] Qingzhong Wang, Pengfei Zhang, Haoyi Xiong, and Jian Zhao. Face. evolve: A high-performance face recognition library. *arXiv preprint arXiv:2107.08621*, 2021.

[10] Jian Zhao, Yu Cheng, Yan Xu, Lin Xiong, Jianshu Li, Fang Zhao, Karlekar Jayashree, Sugiri Pranata, Shengmei Shen, Junliang Xing, et al. Towards pose invariant face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2207–2216, 2018.

[11] Jian Zhao, Jianshu Li, Hengzhu Liu, Shuicheng Yan, and Jiashi Feng. Fine-grained multi-human parsing. *International Journal of Computer Vision*, 128(8):2185–2203, 2020.

[12] Lutao Chu, Yi Liu, Zewu Wu, Shiyu Tang, Guowei Chen, Yuying Hao, Juncai Peng, Zhiliang Yu, Zeyu Chen, Baohua Lai, and Haoyi Xiong. Pp-humanseg: Connectivity-aware portrait segmentation with a large-scale teleconferencing video dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 202–209, 2022.

[13] Jingbo Zhou, Shuangli Li, Liang Huang, Haoyi Xiong, Fan Wang, Tong Xu, Hui Xiong, and Dejing Dou. Distance-aware molecule graph attention network for drug-target binding affinity prediction. *arXiv preprint arXiv:2012.09624*, 2020.

[14] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 975–985, 2021.

[15] Congxi Xiao, Jingbo Zhou, Jizhou Huang, An Zhuo, Ji Liu, Haoyi Xiong, and Dejing Dou. C-watcher: A framework for early detection of high-risk neighborhoods ahead of covid-19 outbreak. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4892–4900, 2021.

[16] Jindong Han, Hao Liu, Haoyi Xiong, and Jing Yang. Semi-supervised air quality forecasting via self-supervised hierarchical graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[17] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[18] Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[19] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, page 201611835, 2017.

[21] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[23] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

[24] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan. DELTA: DEEP LEARNING TRANSFER USING FEATURE MAP WITH ATTENTION FOR CONVOLUTIONAL NETWORKS. In *International Conference on Learning Representations*, 2019.

[25] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[26] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

[27] Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12856–12864, 2020.

[28] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[29] P. Jing, Y. Su, L. Nie, and H. Gu. Predicting image memorability through adaptive transfer learning from external sources. *IEEE Transactions on Multimedia*, 19(5):1050–1062, 2017.

[30] C. Yan, L. Li, C. Zhang, B. Liu, Y. Zhang, and Q. Dai. Cross-modality bridging and knowledge transferring for image understanding. *IEEE Transactions on Multimedia*, 21(10):2675–2685, 2019.

[31] S. Lin, R. Ji, C. Chen, D. Tao, and J. Luo. Holistic cnn compression via low-rank decomposition with knowledge transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):2889–2905, 2019.

[32] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[33] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4109–4118, 2018.

[34] Jiaming Liu, Yali Wang, and Yu Qiao. Sparse deep transfer learning for convolutional neural network. In *AAAI*, pages 2245–2251, 2017.

[35] Mehmet Aygun, Yusuf Aytar, and Hazim Kemal Ekenel. Exploiting convolution filter patterns for transfer learning. In *ICCV Workshops*, pages 2674–2680, 2017.

[36] Yinghua Zhang, Yu Zhang, and Qiang Yang. Parameter transfer unit for deep neural networks. *arXiv preprint arXiv:1804.08613*, 2018.

[37] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[38] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10–19, 2017.

[39] Yuwu Lu, Wenjing Wang, Chun Yuan, Xuelong Li, and Zhihui Lai. Manifold transfer learning via discriminant regression analysis. *IEEE Transactions on Multimedia*, 2020.

[40] M. Yuan and Y. Peng. Ckd: Cross-task knowledge distillation for text-to-image synthesis. *IEEE Transactions on Multimedia*, 22(8):1955–1968, 2020.

[41] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[42] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations*, 2018.

[43] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[44] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[45] Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647. ACM, 2005.

[46] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[47] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[48] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[49] Haishan Ye, Zhichao Huang, Cong Fang, Chris Junchi Li, and Tong Zhang. Hessian-aware zeroth-order optimization for black-box adversarial attack. *arXiv preprint arXiv:1812.11377*, 2018.

[50] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[51] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[52] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Commun. ACM*, 61(7):56–66, June 2018.

[53] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[54] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[55] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[57] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[58] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2, page 1, 2011.

[59] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[60] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[61] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[62] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

[63] Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger B. Grosse. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*, 2018.

[64] Aditya Sharad Golatkar, Alessandro Achille, and Stefano Soatto. Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence. In *Advances in Neural Information Processing Systems*, volume 32, pages 10677–10687, 2019.

[65] Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, Vladimir Braverman, and Zhanxing Zhu. On the noisy gradient descent that generalizes as SGD. In Hal DaumÃĺ III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10367–10376. PMLR, 13–18 Jul 2020.

[66] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 6–6, 2021.

[67] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. *Advances in Neural Information Processing Systems*, 32:1908–1918, 2019.

[68] Kaichao You, Zhi Kou, Mingsheng Long, and Jianmin Wang. Co-tuning for transfer learning. *Advances in Neural Information Processing Systems*, 33, 2020.