Orthogonal Projection Loss

Kanchana Ranasinghe†*, Muzammal Naseerο†, Munawar Hayat§
Salman Khan†ο, Fahad Shahbaz Khan†‡
†Mohamed bin Zayed University of AI, UAE   οAustralian National University, Australia
§Monash University, Australia   ‡Linköping University, Sweden   *Stony Brook University, USA
kranasinghe@cs.stonybrook.edu

Abstract

Deep neural networks have achieved remarkable performance on a range of classification tasks, with softmax cross-entropy (CE) loss emerging as the de-facto objective function. The CE loss encourages features of a class to have a higher projection score on the true class-vector compared to the negative classes. However, this is a relative constraint and does not explicitly force different class features to be well-separated. Motivated by the observation that ground-truth class representations in CE loss are orthogonal (one-hot encoded vectors), we develop a novel loss function termed ‘Orthogonal Projection Loss’ (OPL) which imposes orthogonality in the feature space. OPL augments the properties of CE loss and directly enforces inter-class separation alongside intra-class clustering in the feature space through orthogonality constraints on the mini-batch level. As compared to other alternatives of CE, OPL offers unique advantages e.g., no additional learnable parameters, does not require careful negative mining and is not sensitive to the batch size. Given the plug-and-play nature of OPL, we evaluate it on a diverse range of tasks including image recognition (CIFAR-100), large-scale classification (ImageNet), domain generalization (PACS) and few-shot learning (mini-ImageNet, CIFAR-FS, tiered-ImageNet and Meta-dataset) and demonstrate its effectiveness across the board. Furthermore, OPL offers better robustness against practical nuisances such as adversarial attacks and label noise. Code is available at: https://github.com/kahnchana/opl.

1. Introduction

Recent years have witnessed great success across a range of computer vision tasks owing to progress in deep neural networks (DNNs) [23]. Effective loss functions for DNNs training have been a crucial component of these advancements [15]. In particular, the softmax cross entropy (CE) loss, commonly used for tackling classification problems, has been pivotal for stable and efficient training of DNNs.

Multiple variants of CE have been explored to enhance discriminativity and generalizability of feature representations learned during training. Contrastive [16] and triplet [43] loss functions are a common class of methods that have gained popularity on tasks requiring more discriminative features. At the same time, methods like centre loss [57] and contrastive centre loss [37] have attempted to explicitly enforce inter-class separation and intra-class clustering through Euclidean margins between class prototypes. Angular margin based losses [32, 31, 7, 53, 52] compose another class of objective functions that increase inter-class margins through altering the logits prior to the CE loss.
While these methods have proven successful at promoting better inter-class separation and intra-class compactness, they do possess certain drawbacks. Contrastive and triplet loss functions [16, 43] are dependent on carefully designed negative mining procedures, which are both time-consuming and performance-sensitive. Methods based on centre loss [57, 37], that work together with CE loss, promote margins in Euclidean space which is counter-intuitive to the intrinsic angular separation enforced through CE loss [31]. Further, these methods introduce additional learnable parameters in the form of new class centres. Angular margin based loss functions [31, 32] which are highly successful for face recognition tasks, make strong assumptions for face embeddings to lie on the hypersphere manifold, which does not hold universally for all computer vision tasks [45]. Some loss designs are also specific to certain architecture classes e.g., [45] can only work with DNNs which output Class Activation Maps [65].

In this work, we explore a novel direction of simultaneously enforcing inter-class separation and intra-class clustering through orthogonality constraints on feature representations learned in the penultimate layer (Fig. 1). We propose Orthogonal Projection Loss (OPL), which can be applied on the feature space of any DNN as a plug-and-play module. We are motivated by how image classification inherently assumes independent output classes and how orthogonality constraints in feature space go hand in hand with the one-hot encoded (orthogonal) label space used with CE. Furthermore, orthogonality constraints provide a definitive geometric structure in comparison to arbitrarily increasing margins which are prone to change depending on the selected batch, thus reducing sensitivity to batch composition. Finally, simply maximizing the margins can cause negative correlation between classes and thereby unnecessarily focus on well-separated classes while we tend to ensure independence between different class features to successfully disentangle the class-specific characteristics.

Compared with contrastive loss functions [16, 43], OPL operates directly on mini-batches, eliminating the requirement of complex negative sample mining procedures. By enforcing orthogonality through computing dot-products between feature vectors, OPL provides a natural augmentation to the intrinsic angular property of CE, as opposed to methods [57, 37, 17] that enforce an Euclidean margin in feature space. Furthermore, OPL introduces no additional learnable parameters unlike [57, 51, 37], operates independent of model architecture unlike [45], and in contrast to losses operating on the hypersphere manifold [31, 32, 7, 53], performs well on a wide range of tasks.

In summary, our main contributions are as follows:

- We propose a novel loss, OPL, that directly enforces inter-class separation and intra-class clustering via orthogonality constraints with no learnable parameters.
- Our orthogonality constraints are efficiently formulated compared to existing methods [26, 46], allowing minibatch processing without the need to explicitly obtain singular values. This leads to a simple vectorized implementation of OPL directly integrating with CE.
- We extensively evaluate on a diverse range of image classification tasks highlighting the discriminative ability of OPL. Further, our results on few-shot learning (FSL) and domain generalization (DG) datasets establish the transferability and generalizability of features learned with OPL. Finally, we establish the improved robustness of learned features to adversarial attacks and label noise.

2. Related Work

Generalizable Representations. Recent works explore the transferability of features learned via supervised training [60], e.g. FSL [49, 4, 14] and DG [20, 9] tasks. Tian et al. [49] establish a strong FSL baseline using only standard (non-episodic) supervised pre-training. Adaptation of supervised pre-trained models to the episodic evaluation setting of FSL tasks is explored in [59, 4]. Goldblum et al. [14] show the importance of margin-based regularization methods for FSL. Our work differs by building on orthogonality constraints to learn more transferable features, and is more compatible with CE as opposed to [14]. Multiple DG methods also explore constraints on feature spaces [20, 9] to boost cross-domain performance. In particular, [9] explores inter-class separation and intra-class clustering through contrastive and triplet loss functions. OPL improves on these while eliminating the need for compute expensive and complex sample mining procedures.

Orthogonality. Orthogonality of kernels in DNNs is well explored with an aim to diversify the learned weight vectors [33, 54]. The idea of orthogonality is also used for disentangled representations such as in [55] and to stabilize network training since orthogonalization ensures energy preservation [8, 41]. Orthogonal weight initializations have also shown their promise towards improving learning behaviours [36, 58]. However, all of these works operate in the parameter space while our work applies orthogonality on a feature space to enforce better inter-class separation and intra-class clustering. Remarkably, the previous formulations to achieve orthogonality in the feature space generally depend on computing singular value decomposition [26, 46], which can be numerically unstable, difficult to estimate for rectangular matrices, and undergoes an iterative process [2]. In contrast, our orthogonal constraints are enforced in a novel manner, realized via decomposition on the sample-to-sample relationships within a mini-batch, while simultaneously avoiding tedious pair/triplet computations.
3. Proposed Method

Maximizing inter-class separation while enhancing intra-class compactness is highly desirable for classification. While the commonly used cross-entropy (CE) loss encourages logits of the same class to be closer together, it does not enforce any margin amongst different classes. There have been multiple efforts to integrate max-margin learning with CE. For example, Large-margin softmax [32] enforces inter-class separability directly on the dot-product similarity while SphereFace [31] and ArcFace [7] enforce multiplicative and additive angular margins on the hypersphere manifold, respectively. Directly enforcing max-margin constraints to enhance discriminability in the angular domain is ill-posed and requires approximations [32]. Some works turn to the Euclidean space to enhance feature space discrimination. For example, centre loss [57] clusters penultimate layer features using Euclidean distance. Similarly, Affinity Loss [17] forms uniformly shaped equi-distant class-wise clusters based upon Gaussian distances in the Euclid space. Margin-maximizing objective functions in the Euclid space are not ideally suited to work alongside CE loss, since CE seeks to separate output logits in the angular domain. By enforcing orthogonality constraints, our proposed OPL loss maximally separates intermediate features in the angular domain, thus complementing cross-entropy loss which enhances angular discriminability in the output space. In the following discussion, we revisit CE loss in the context of max-margin learning, and argue why OPL loss is ideally suited to supplement CE.

3.1. Revisiting Softmax Cross Entropy Loss

Consider a deep neural network $H$, which can be decomposed into $H = H_o \cdot H_y$, where $H_o$ is the feature extraction module and $H_y$ is the classification module. Given an input-output pair $(x, y)$, let $f = H_o(x), f \in \mathbb{R}^d$ be the intermediate features and $\hat{y} = H_y(f), \hat{y} \in \mathbb{R}^k$ be the output predictions. For brevity, let us define the classification module as a linear layer $H_y = W = [w_1, \ldots, w_c]$ with no unit-biases, where $w_i \in \mathbb{R}^d, i = 1 \cdots c$ are class-wise learnable projection vectors for $c$ classes. The traditional CE loss can then be defined in terms of discrepancy between the predicted $\hat{y}$ and ground-truth label $y$, by projecting the features $f \in \mathbb{R}^d$ onto the weight matrix $W \in \mathbb{R}^{d \times c}$.

$$L_{CE}(\hat{y}, y) = - \log \frac{\exp (f^T w_y)}{\sum_j \exp (f^T w_j)}$$

$$\propto \sum_{j \neq y} \exp (f^T w_j - f^T w_y)$$

$$\propto \sum_{j \neq y} \exp (\|f\|_2 \cdot \|w_j\|_2 \cos(\theta_j) - \|f\|_2 \cdot \|w_y\|_2 \cos(\theta_y))$$

Since CE does not explicitly enforce any margin between each class pair, previous works have shown that the learned class regions for some class samples tend to be bigger compared to others [31, 7, 17]. To counter this effect and ensure all classes are equally separated, efforts have been made to introduce a margin $m$ between different classes by modifying the term $\cos(\theta_y)$ as $\cos (m + \theta_y)$ for additive angular margin [7], $\cos(m\theta_y)$ for multiplicative angular margin [31] and $m + \cos (\theta_y)$ as additive cosine margin [53]. The gradient propagation for such margin-based softmax loss formulations is hard, and previous works rely on approximations. Instead of introducing any margins to ensure uniform separation between different classes, our proposed loss function simply enforces all classes to be orthogonal to each other with simultaneous clustering of within-class samples, using an efficient vectorized implementation with straightforward gradient computation and propagation.

By considering the $w_y \in W$ vectors as individual class prototypes, the CE loss can be viewed as aligning the feature vectors $f$ along its relevant class prototype. The cosine similarity in the form of the dot product $(f^T w_y)$ gives CE an intrinsic angular property, which is observed in Fig. 2 where features naturally separate in the polar coordinates with CE only. Moreover, during the standard SGD based optimization, the CE loss is applied on mini-batches. We note that there is no explicit enforcement of feature separation or clustering across multiple samples within the mini-batch. Given the opportunity to enforce such constraints since supervised training is commonly conducted adopting random mini-batch based iterations, we explore the possibility of within mini-batch constraints aimed at augmenting the intrinsic discriminative characteristics of CE loss.
3.2. Orthogonal Projection Loss

The CE loss with one-hot-encoded ground-truth vectors seeks to implicitly achieve orthogonality between different classes in the output space. Our proposed OPL loss, ameliorates CE loss, by enforcing class-wise orthogonality in the intermediate feature space. Given an input-output pair \( \{x_i, y_i\} \) in the dataset \( D \), let \( f_i = H_\theta(x_i) \) be the features output by an intermediate layer of the network. Our objective is to enforce constraints to cluster the features that simultaneously ensures intra-class clustering and inter-class orthogonality within a mini-batch as follows:

\[
s = \sum_{i,j \in B, y_i = y_j} \langle f_i, f_j \rangle / \sum_{i,j \in B, y_i = y_j} 1 \quad (2)
\]

\[
d = \sum_{i,k \in B, y_i \neq y_k} \langle f_i, f_k \rangle \text{git} / \sum_{i,k \in B, y_i \neq y_k} 1 \quad (3)
\]

\[
\mathcal{L}_{\text{OPL}} = (1 - s) + |d| \quad (4)
\]

where \( \langle \cdot, \cdot \rangle \) is the cosine similarity operator applied on two vectors, \( |\cdot| \) is the absolute value operator, and \( B \) denotes mini-batch size. Note that the cosine similarity operator used in Eq. 2 and 3 involves normalization of features (projection to a unit hyper-sphere) as follows:

\[
\langle x_i, x_j \rangle = \frac{x_i \cdot x_j}{\|x_i\|_2 \cdot \|x_j\|_2} \quad (5)
\]

where \( \|\cdot\|_2 \) refers to the \( \ell_2 \) norm operator. This normalization is key to aligning the outcome of OPL with the intrinsic angular property of CE loss.

In Eq. 4, our objective is to push \( s \) towards 1 and \( d \) towards 0. Since \( 1 - s > 0 \) already, we take the absolute value of \( d \) given \( d \in (-1, 1) \). This in turn restricts the overall loss such that \( \mathcal{L}_{\text{OPL}} \in (0, 3) \). When minimizing this overall loss, the first term \( (1 - s) \) will ensure clustering of same class samples, while the second term \( |d| \) will ensure the orthogonality of different class samples. The loss can be implemented efficiently in a vectorized manner on the mini-batch level, avoiding any loops (see Algorithm 1).

We further note that the ratio of contribution to the overall loss of each individual term in Eq. 4 can be controlled to re-prioritize between the two objectives of inter-class separation and intra-class compactness. While the unweighted combination of \( s \) and \( d \) alone performs well, specific use-cases could benefit from weighted combinations. We re-formulate Eq. 4 as follows:

\[
\mathcal{L}_{\text{OPL}} = (1 - s) + \gamma \cdot |d| \quad (6)
\]

where \( \gamma \) is the hyper-parameter controlling the weight for the two different constraints.

Algorithm 1 Pytorch style pseudocode for OPL

```python
def forward(features, labels):
    features = F.normalize(features, p=2, dim=1)
    masks for same and diff class features
    mask = torch.eq(labels, labels.t())
    eye = torch.eye(mask.shape[0])
    mask_pos = mask.masked_fill(eye, 0)
    mask_neg = 1 - mask
    # s & d calculation
    dot_prod = torch.matmul(features, features.t())
    pos_total = (mask_pos * dot_prod).sum()
    neg_total = torch.abs(mask_neg * dot_prod).sum()
    pos_mean = pos_total / (mask_pos.sum() + 1e-6)
    neg_mean = neg_total / (mask_neg.sum() + 1e-6)
    # total loss
    loss = (1.0 - pos_mean) + neg_mean
    return loss
```

Since OPL acts only on intermediate features, we apply cross entropy loss over the outputs of the final classifier \( H_\theta \). The overall loss used is a weighted combination of CE and OPL. We note that our proposed loss can also be used together with other common image classification losses, such as Guided Cross Entropy, Label Smoothing or even task specific loss functions in different computer vision tasks. The overall loss \( \mathcal{L} \) can be defined as:

\[
\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{OPL}} \quad (7)
\]

where \( \lambda \) is a hyper-parameter controlling the OPL weight.

3.3. Interpretation and Analysis

Overall Objective: Consider \( F_c \) is a set of mini-batch samples comprising of normalized features from the same class \( c \) in a given dataset \( D \). The overall OPL constraints can be viewed as a minimization of the following objective to update the network \( H_\phi \) over the random variables \( F_c \):

\[
\min_{\phi} \sum_{i=1}^{c} \sum_{j=1}^{c} \mathbb{E}_{(F_i, F_j) \sim D} \left[ |F_i F_j^T| - \epsilon \right] \quad (8)
\]

where \( \epsilon \) is the absolute value operator and \( \epsilon \) is the Iverson bracket operator. We refer to the term defined in Eq. 8 as the expected inter-class orthogonality. The behaviour of OPL in terms of minimizing this expectation is visualized in Fig. 4 where average per-class feature vectors over the CIFAR-100 dataset are calculated for ResNet-56 models trained under ‘CE-only’ and ‘CE+OPL’ settings. Clear improvements over the CE baseline in terms of minimizing the expected inter-class orthogonality can be observed. Moreover, the stochastic mini-batch based application of OPL prevents naively pushing all non-diagonal values to zero as...
We compare feature orthogonality as measured by OPL and feature similarity as measured by cosine similarity and plot their convergence during training. Feature similarity is initially high because all features are random immediately after initialization. OPL simultaneously enforces higher inter-class similarity and intra-class dissimilarity in comparison with the CE baseline.

We present matrices illustrating the orthogonality of average per class features computed over the CIFAR-100 test set. See Appendix B.3 for more analysis. This translates to allowing necessary inter-class relationships not encoded in one-hot labels of a dataset to be captured within the learned features.

Taking a step further, we decompose OPL into its sub-components, $s$ and $d$, as defined in Eq. 4. While $s$ computes the pair-wise cosine similarity between all same-class features within the mini-batch, $d$ calculates the similarity between different class features. These measures can directly be adopted to quantify the inter-class separation and intra-class compactness in any given features space. Moreover, the unweighted OPL formulation in Eq. 4 can be considered a measure of the overall feature orthogonality within any given embedding space. It will be interesting to compare the contribution of OPL towards inter-class separation and intra-class clustering of a feature space in contrast to the generic CE based training scenario. We present this comparison by training ResNet-56 on CIFAR-100 dataset in Fig. 3. This separation of features achieved through OPL translates to performance improvements not only within the standard classification setting, but also in tasks requiring transferable or generalizable features. Goldblum et al. [14] explore the significance of inter-class separation and intra-class clustering for better performance when transferring a feature embedding to a few-shot learning task. Similar notions regarding discriminative features are explored in [9] for domain generalization. We explore the effects of OPL in few-shot learning settings, and visualize the novel class embeddings learned with OPL in Appendix B using LDA [35] to preserve the inter-class to intra-class variance ratio as suggested in [14].

**Why orthogonality constraints?** One may wonder what benefit orthogonality in the feature space can provide in comparison to simply maximizing margin between classes. Our reasoning is twofold: reducing sensitivity to batch composition and avoiding negative correlation constraints. Within the random mini-batch based training setting, the orthogonality objective provides a definitive geometric structure irrespective of the batch composition, while the optimal max-margin separation is dependent on the batch composition. Furthermore, in the common case where the output space feature dimension $d > c$ (c number of classes), maximizing angular margin between normalized features on a unit hyper-sphere will lead to negative correlation among the class prototypes (considering a maximal and equi-angular separation). We argue that this is an undesired constraint since the categorical classification task itself assumes non-existence of ordinal relationships between classes (e.g. use of orthogonal one-hot encoded labels). Moreover, extending the constraints to additionally cause negative correlation between classes unnecessarily focuses on already well-separated classes during training whereas our constraint which tends to ensure independence provides a more balanced objective to disentangle the class-specific characteristics of even more fine-grained classes.

### 4. Experiments

We extensively evaluate our proposed loss function on multiple tasks including image classification (Tables 1 & 2), robustness against label noise (Table 3), robustness against adversarial attacks (Table 4) and generalization to domain shifts (Table 5). We further observe the enhanced transferability of orthogonal features e.g. in the case of few shot

---

**Figure 3: Feature Analysis:** We compare feature orthogonality as measured by OPL and feature similarity as measured by cosine similarity and plot their convergence during training. Feature similarity is initially high because all features are random immediately after initialization. OPL simultaneously enforces higher inter-class similarity and intra-class dissimilarity in comparison with the CE baseline.

**Figure 4: Orthogonality Visualization:** We present matrices illustrating the orthogonality of average per class features computed over the CIFAR-100 test set. See Appendix B.3 for more analysis.
learning (Tables 6 & 7). Our approach shows consistent improvements and highlights the advantages of orthogonal features on this diverse set of tasks and datasets with various deep network backbones. Additionally, we demonstrate the plug-and-play nature of OPL by showing benefits of its use over CE, Truncated Loss (for noisy labels) [64], RSC [20] and various adversarial learning baselines.

4.1. Image Classification

We evaluate the effectiveness of orthogonal features in the penultimate layer on image classification using our proposed training objective (Eq. 7). Competitive results are achieved showing consistent improvements (Tables 1 & 2) on two datasets: CIFAR-100 [21], and ImageNet [22]. CIFAR-100 consists of 60,000 natural images spread across 100 classes, with 600 images per class. We apply OPL over a cross-entropy baseline for supervised classification on CIFAR-100 (following the experiment setup in [45]), and compare our results against other loss functions which impose margin constraints [31, 32, 53, 7], introduce regularization [45, 26, 29, 6], or promote clustering [57, 63] to enhance separation among classes in Table 1. Despite its simplicity, our method performs well against state-of-the-art loss functions. Note that HNC [45] is dependant on class activation maps, RBF [63] and LGM [51] involve learnable parameters, and CB Focal Loss [6] specifically solves class-imbalance. In contrast, OPL has a simple formulation easily integratable to any network architecture, involves no learnable parameters, and targets general classification. Additionally, we note how OPL has higher performance gains with respect to top-1 accuracy (in comparison to top-5 accuracy) which is the more challenging metric. We attribute this to the fact that increased separation through OPL mostly helps in classifying difficult samples. Further, we note top-5 is not a preferred measure for CIFAR-100 since most classes are different in nature as opposed to e.g., ImageNet with several closely related classes (where our gain is much pronounced for top-5 accuracy, as discussed next).

ImageNet is a standard large-scale dataset used in visual recognition tasks, containing roughly 1.2 million training images and 50,000 validation images. We experiment with OPL by integrating it to common backbone architectures used in image classification tasks: ResNet18 and ResNet50. We train 1 the models for 90 epochs using SGD with momentum (initial learning rate 0.1 decayed by 10 every 30 epochs). Results for these experiments are presented in Table 2 and Fig. 5. We note that simply enforcing our orthogonality constraints increases the top-1 (%) accuracy of ResNet50 from 76.15% to 76.98% without any additional bells and whistles. Moreover, given the large number of fine-grained classes among the 1000 categories of ImageNet (e.g., multiple dog species) which can be viewed as difficult cases, the better discriminative features learned by OPL obtains notable improvements in top-5 accuracy as well.

4.2. Robustness against Label Noise

Given the rich representation capacity of deep neural networks, especially considering how most can even fit random labels or noise perfectly [62], errors in the sample labels pose a significant challenge for training. In most practical applications, label noise is almost impossible to avoid, in particular, when it comes to large-scale datasets requiring millions of human annotations. Multiple works [13, 64] explore modifications to common objective functions aimed at building robustness to label noise. Despite the explicit inter-
class separation constraints on the feature space enforced by OPL, we argue that the random mini-batch based optimization exploited by OPL negates the effects of noisy labels. This hypothesis is supported by our experiments presented in Table 3 which show additional robustness of OPL against label noise. We simply integrate OPL over the approach followed in [64], without any task-specific modifications.

### 4.3. Robustness against Adversarial Attacks

Adversarial attacks modify a given benign sample by adding adversarial noise such that the deep neural network is deceived [48]. Adversarial examples are out-of-distribution samples and remain a challenging problem to solve. Adversarial training [34] emerges as an effective defense where adversarial examples are generated and added into the training set. We enforced orthogonality on such adversarial examples in the feature space while optimizing the model weights and show our benefit on different adversarial training mechanisms [34, 19, 56]. Important to note that all the considered adversarial training schemes [34, 19, 56] are different in nature e.g. training in Madry et al. [34] is based on cross-entropy only, Hendrycks et al. [19] propose to exploit pre-training, while Wang et al. [56] introduce a surrogate loss along with cross-entropy. Our orthogonality constraint help maximizing adversarial robustness in all cases showing the generic and plug-and-play nature of our proposed loss. In order to have reliable evaluation, we report robustness gains against Auto-Attack (AA) [5] in Table 4. On CIFAR10, our method increased robustness of [34] by 5.11%, [19] by 0.81% and [56] by 2.2%.

### 4.4. Domain Generalization (DG)

The DG problem aims to train a model using multi-domain source data such that it can directly generalize to new domains without the need of retraining. We argue that the feature space constraints of OPL tend to capture more general semantic features in images which generalize better across domains. This is verified by the performance improvements for DG that we obtain by integrating OPL with the state-of-the-art approach in [20] and evaluating on the popular PACS dataset [28]. The results presented in Table 5 indicate that integrating OPL with [20] sets new state-of-the-art across all four domains as compared to [20].

---

**Table 3: Results on CIFAR-100 for Noisy Labels:** We explore the effect of noisy labels when training with OPL for image classification tasks. We use the method in [64] as a baseline comparison with 0.4 noise level and ResNet18 backbone.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Uniform</th>
<th>Class Dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>TL              [64]</td>
<td>87.62%</td>
<td>82.28%</td>
</tr>
<tr>
<td></td>
<td>TL + OPL</td>
<td>88.45%</td>
<td>87.02%</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>TL              [64]</td>
<td>62.64%</td>
<td>47.66%</td>
</tr>
<tr>
<td></td>
<td>TL + OPL</td>
<td>65.62%</td>
<td>53.94%</td>
</tr>
</tbody>
</table>

**Table 4: OPL performance on Adversarial Robustness:** We show the impact of enforcing orthogonality on the robust features. We adversarially train baseline methods [34, 19, 56] by adding OPL constraint during training. Robust features obtained with OPL leads to better accuracy and show clear improvements over the baseline. Top-1 accuracy is reported against Auto-Attack [5] in whitebox setting (attacker has full knowledge of the model architecture and pretrained weights).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Clean</th>
<th>Advers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>Madry et al. [ICLR'18] [34]</td>
<td>87.14</td>
<td>44.04</td>
</tr>
<tr>
<td></td>
<td>Madry et al. [ICLR'18] + OPL</td>
<td>87.76</td>
<td>49.15</td>
</tr>
<tr>
<td></td>
<td>Hendrycks et al. [ICLR'18] [19]</td>
<td>87.11</td>
<td>54.92</td>
</tr>
<tr>
<td></td>
<td>Hendrycks et al. [ICLR'18] + OPL</td>
<td>87.51</td>
<td>55.73</td>
</tr>
<tr>
<td></td>
<td>MART [50] + OPL</td>
<td>84.49</td>
<td>54.10</td>
</tr>
<tr>
<td></td>
<td>MART [56] + OPL</td>
<td>84.41</td>
<td>56.23</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>Madry et al. [ICLR'18] [34]</td>
<td>60.20</td>
<td>20.60</td>
</tr>
<tr>
<td></td>
<td>Madry et al. [ICLR'18] + OPL</td>
<td>61.13</td>
<td>23.01</td>
</tr>
<tr>
<td></td>
<td>Hendrycks et al. [ICLR'18] [19]</td>
<td>59.23</td>
<td>28.42</td>
</tr>
<tr>
<td></td>
<td>Hendrycks et al. [ICLR'18] + OPL</td>
<td>61.00</td>
<td>30.05</td>
</tr>
<tr>
<td></td>
<td>MART [ICLR'20] [36]</td>
<td>58.90</td>
<td>23.40</td>
</tr>
<tr>
<td></td>
<td>MART [56] + OPL</td>
<td>58.01</td>
<td>25.74</td>
</tr>
</tbody>
</table>

**Table 5: Results on PACS dataset:** We integrate OPL with [20], gaining improvements for domain generalization tasks (*best replicated value).

<table>
<thead>
<tr>
<th>Method</th>
<th>Art</th>
<th>Cartoon</th>
<th>Sketch</th>
<th>Photo</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>JiGe [CVPR'19] [3]</td>
<td>86.20</td>
<td>78.70</td>
<td>70.63</td>
<td>97.66</td>
<td>83.29</td>
</tr>
<tr>
<td>MASF [NeurIPS'19] [10]</td>
<td>82.89</td>
<td>80.49</td>
<td>72.29</td>
<td>95.01</td>
<td>82.67</td>
</tr>
<tr>
<td>MetaReg [NeurIPS'18] [11]</td>
<td>87.20</td>
<td>79.20</td>
<td>70.30</td>
<td>97.60</td>
<td>83.60</td>
</tr>
<tr>
<td>RSC [ECCV'20] [20]</td>
<td>87.89</td>
<td>82.16</td>
<td>83.35</td>
<td>96.47</td>
<td>87.47</td>
</tr>
<tr>
<td>RSC + OPL</td>
<td><strong>88.28</strong></td>
<td><strong>84.64</strong></td>
<td><strong>84.17</strong></td>
<td><strong>96.83</strong></td>
<td><strong>88.48</strong></td>
</tr>
</tbody>
</table>

**Table 6: Results on FSL tasks:** In this section, we explore the transferability of features learned with our loss function in relation to FSL tasks. We evaluate OPL on three benchmark few-shot classification datasets: miniImageNet, tieredImageNet, and CIFAR-FS. We run additional experiments on Meta-Dataset [50] which is a large-scale benchmark for evaluating FSL methods in more diverse and challenging settings. Similar to [40], we expand Meta-Dataset by adding three additional datasets, MNIST, CIFAR10, and CIFAR100. In light of work that shows promise of learning strong features for FSL [49], we experiment with OPL using it as an auxiliary loss on the feature space during the supervised training. Quantitative results highlighting the performance improvements are presented in Table 6. Our results on Meta-Dataset are obtained using the train on all setting presented in [50]. We integrate OPL over the method presented in [11], train on the first 8 datasets of Meta-Dataset, and evaluate on the rest (including the three additional datasets from [40]). Results are presented in Table 7. See Appendix A.4 for robustness of OPL features against sample noise in FSL tasks.
### Variation with Batch Size

<table>
<thead>
<tr>
<th>Method</th>
<th>New Loss</th>
<th>Cifar:1shot</th>
<th>Cifar:5shot</th>
<th>Mini:1shot</th>
<th>Mini:5shot</th>
<th>Tier:1shot</th>
<th>Tier:5shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML(_\text{PMLR'17}) [12]</td>
<td>-</td>
<td>58.90±1.9</td>
<td>71.50±1.0</td>
<td>48.70±1.84</td>
<td>63.11±0.92</td>
<td>51.67±1.81</td>
<td>70.30±1.75</td>
</tr>
<tr>
<td>PN (_\text{NIPS'17}) [44]</td>
<td>-</td>
<td>55.50±0.7</td>
<td>72.00±0.6</td>
<td>49.42±0.78</td>
<td>68.20±0.66</td>
<td>53.31±0.89</td>
<td>72.69±0.74</td>
</tr>
<tr>
<td>RNet(_\text{CVPR'18}) [47]</td>
<td>-</td>
<td>55.00±1.0</td>
<td>69.30±0.8</td>
<td>50.44±0.82</td>
<td>65.32±0.70</td>
<td>54.48±0.93</td>
<td>71.32±0.78</td>
</tr>
<tr>
<td>Shot-Free(_\text{ECCV'19}) [39]</td>
<td>-</td>
<td>69.20±N/A</td>
<td>84.70±N/A</td>
<td>59.04±N/A</td>
<td>77.64±N/A</td>
<td>63.52±N/A</td>
<td>82.59±N/A</td>
</tr>
<tr>
<td>MetaOptNet(_\text{CVPR'19}) [25]</td>
<td>-</td>
<td>72.60±0.7</td>
<td>84.30±0.5</td>
<td>62.64±0.61</td>
<td>78.63±0.46</td>
<td>65.99±0.72</td>
<td>81.56±0.53</td>
</tr>
<tr>
<td>RFS(_\text{ECCV'20}) [49]</td>
<td>✓</td>
<td>71.45±0.8</td>
<td>85.95±0.5</td>
<td>62.02±0.60</td>
<td>79.64±0.44</td>
<td>69.74±0.72</td>
<td>84.41±0.55</td>
</tr>
<tr>
<td>RFS + OPL (Ours)</td>
<td>✓</td>
<td>73.02±0.4</td>
<td>86.12±0.2</td>
<td>63.10±0.36</td>
<td>79.87±0.26</td>
<td>70.20±0.41</td>
<td>85.01±0.27</td>
</tr>
<tr>
<td>NAML(_\text{CVPR'20}) [27]</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>65.42±0.25</td>
<td>75.48±0.34</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Neg-Cosine(_\text{ECCV'20}) [30]</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>63.85±0.81</td>
<td>81.57±0.56</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SKD(_\text{Arxiv'20}) [38]</td>
<td>✓</td>
<td>74.50±0.9</td>
<td>88.00±0.6</td>
<td>65.93±0.81</td>
<td>83.15±0.54</td>
<td>71.69±0.91</td>
<td>86.66±0.60</td>
</tr>
<tr>
<td>SKD + OPL (Ours)</td>
<td>✓</td>
<td>74.94±0.4</td>
<td>88.06±0.3</td>
<td>66.90±0.37</td>
<td>83.23±0.25</td>
<td>72.10±0.41</td>
<td>86.70±0.27</td>
</tr>
</tbody>
</table>

Table 6: Few-Shot Learning Improvements: We obtain performance improvements using OPL over the RFS [49] baseline and SKD baseline [38] containing ResNet-12 backbones. Our loss is simply plugged in to their supervised feature learning phase. Results reported for our experiment are averaged over 3000 episodic runs. Note that [38, 27, 30] are recent loss functions specific to FSL.

### Table 7: Results on Meta-Dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CNAPs(_\text{[40]})</th>
<th>SUR(<em>\text{[11]})(</em>\text{(NeurIPS'19)})</th>
<th>SUR + OPL(_\text{[Ours]})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imagenet</td>
<td>52.3±1.0</td>
<td>56.4±1.2</td>
<td>56.5±1.1</td>
</tr>
<tr>
<td>Omniglot</td>
<td>88.4±0.7</td>
<td>88.5±0.8</td>
<td>89.8±0.7</td>
</tr>
<tr>
<td>Aircraft</td>
<td>80.5±0.6</td>
<td>79.5±0.8</td>
<td>79.6±0.7</td>
</tr>
<tr>
<td>Birds</td>
<td>72.2±0.9</td>
<td>76.4±0.9</td>
<td>76.9±0.7</td>
</tr>
<tr>
<td>Textures</td>
<td>58.3±0.7</td>
<td>73.1±0.7</td>
<td>72.7±0.7</td>
</tr>
<tr>
<td>Quick Draw</td>
<td>72.5±0.8</td>
<td>75.7±0.7</td>
<td>75.7±0.7</td>
</tr>
<tr>
<td>Fungi</td>
<td>47.4±1.0</td>
<td>48.2±0.9</td>
<td>50.1±1.0</td>
</tr>
<tr>
<td>VGG Flower</td>
<td>86.0±0.5</td>
<td>90.6±0.5</td>
<td>90.9±0.5</td>
</tr>
<tr>
<td>MSOCO</td>
<td>42.6±1.1</td>
<td>52.1±1.0</td>
<td>52.0±1.0</td>
</tr>
<tr>
<td>MNIST</td>
<td>92.7±0.4</td>
<td>93.2±0.4</td>
<td>94.3±0.4</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>61.5±0.7</td>
<td>66.4±0.8</td>
<td>66.6±0.7</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>50.1±1.0</td>
<td>57.1±1.0</td>
<td>57.6±1.0</td>
</tr>
<tr>
<td>Average</td>
<td>67.0</td>
<td>71.4</td>
<td>71.9</td>
</tr>
</tbody>
</table>

Table 7: Results on Meta-Dataset: OPL is integrated with the SUR-PNF method in [11] for the Meta-Dataset train on all setting. Traffic Signs dataset has been omitted in comparisons due to an error in Meta-Dataset possibly affecting prior work.

### Table 8: Hyper-parameter search

<table>
<thead>
<tr>
<th>hyper-parameter</th>
<th>γ=2</th>
<th>γ=1</th>
<th>γ=0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ = 0.05</td>
<td>70.48</td>
<td>70.66</td>
<td>72.02</td>
</tr>
<tr>
<td>λ = 0.1</td>
<td>70.12</td>
<td>70.94</td>
<td>71.30</td>
</tr>
<tr>
<td>λ = 0.5</td>
<td>70.26</td>
<td>71.18</td>
<td>70.66</td>
</tr>
<tr>
<td>λ = 1</td>
<td>69.78</td>
<td>70.48</td>
<td>72.20</td>
</tr>
<tr>
<td>λ = 2</td>
<td>67.64</td>
<td>69.58</td>
<td>70.52</td>
</tr>
</tbody>
</table>

Table 8: Hyper-parameter search: We report the top-1 accuracy values on CIFAR-100 held-out validation set, using ResNet-56 after training using OPL with various pairs of λ and γ.

### 4.6. Ablative Study

OPL in its full form (Eq. 6) contains two hyper-parameters, λ and γ. We conduct a hyper-parameter search over a held-out validation set of CIFAR-100 (see Table 8). The optimum values selected from these experiments are kept consistent across all other tasks and used when reporting test performance. Furthermore, we evaluate the performance of OPL on the test split of CIFAR-100 for varying λ values keeping γ fixed, to illustrate the minimal sensitivity of our method to different λ values in Fig. 6a. Next, we consider how OPL operates on random mini-batches, and evaluate its performance against varying batch sizes (CIFAR-100 dataset). These results presented in Fig. 6b exhibit how OPL consistently achieves performance boosts over a CE baseline across varying batch-sizes.

### 5. Conclusion

We present a simple yet effective loss function to enforce orthogonality on the output feature space and show its improvements for a wide range of classification tasks. Our loss operates in conjunction with the softmax CE loss, and can be easily integrated with any DNN. We also explore a variety of characteristics of the features learned with OPL illustrating its benefit for few-shot learning, domain-generalization and robustness against adversarial attacks and label noise. In future, we hope to explore other variants of OPL including its adaptation to unsupervised learning.

Acknowledgements: This work is partially supported by ARC DECRA Fellowship DE200101100.
References


[57] Y. Wen, Kaipeng Zhang, Z. Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In ECCV, 2016. 1, 2, 3, 6


[63] Xiao Zhang, Rui Zhao, Yu Qiao, and Hongsheng Li. Rbf-softmax: Learning deep representative prototypes with radial basis function softmax. In ECCV, 2020. 6, 12
