Orthogonal Projection Loss

Kanchana Ranasinghe, Muzammal Naseer, Munawar Hayat,
Salman Khan, Fahad Shahbaz Khan

Code publicly available: https://kahnchana.github.io/opl

We propose a novel loss function

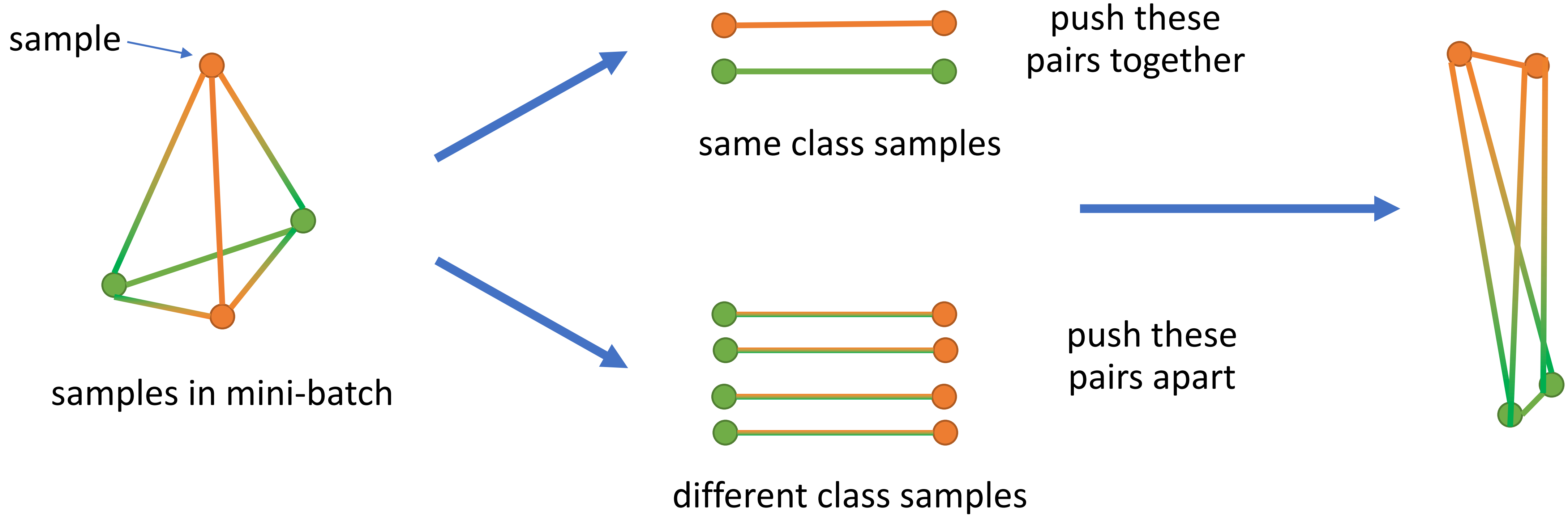
Task	Dataset	Baseline	OPL	Metric
Classification	CIFAR-100	72.40%	73.52%	acc@1
Classification	ImageNet	78.31%	79.26%	acc@1
Few Shot Classification	CIFAR-FS	71.45%	73.02%	1-shot
Few Shot Classification	MiniImageNet	62.02%	63.10%	1-shot
Few Shot Classification	TieredImageNet	69.74%	70.20%	1-shot
Few Shot Classification	MetaDataset (avg)	71.4%	71.9%	varying shot
Domain Generalization	PACS (avg)	87.47%	88.48%	acc@1
Label Noise	CIFAR-10	87.62%	88.45%	acc@1
Label Noise	CIFAR-100	62.64%	65.62%	acc@1
Adversarial Robustness	CIFAR-10	54.92%	55.73%	acc@1
Adversarial Robustness	CIFAR-100	28.42%	30.05%	acc@1

Results across tasks

- Orthogonal Projection Loss (OPL) applies feature space constraints
- same-class feature clustering and different-class feature separation
- features learned by OPL are more ***discriminative*** and ***generalizable***

Orthogonal Projection Loss

feature space



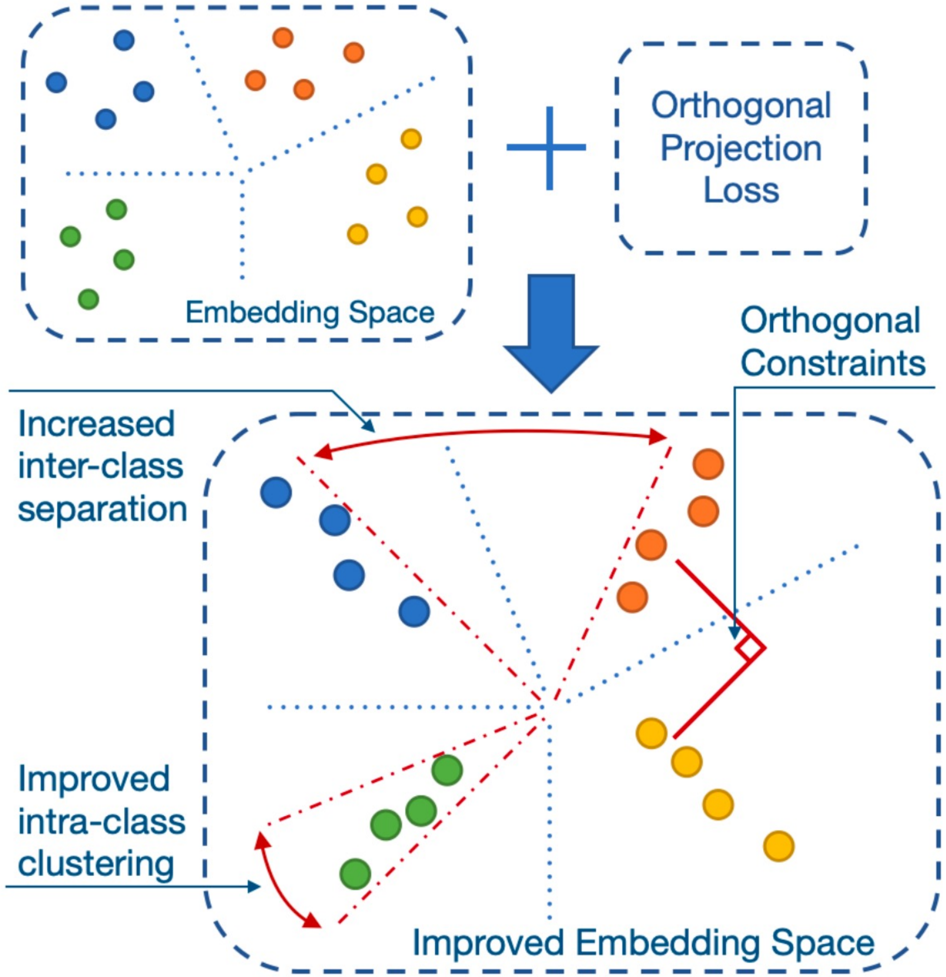
Proposed Method

$$s = \sum_{\substack{i,j \in B \\ y_i = y_j}} \langle \mathbf{f}_i, \mathbf{f}_j \rangle / \sum_{\substack{i,j \in B \\ y_i = y_j}} 1$$

$$d = \sum_{\substack{i,k \in B \\ y_i \neq y_k}} \langle \mathbf{f}_i, \mathbf{f}_k \rangle / \sum_{\substack{i,k \in B \\ y_i \neq y_k}} 1$$

$$\mathcal{L}_{\text{OPL}} = (1 - s) + \gamma * |d|$$

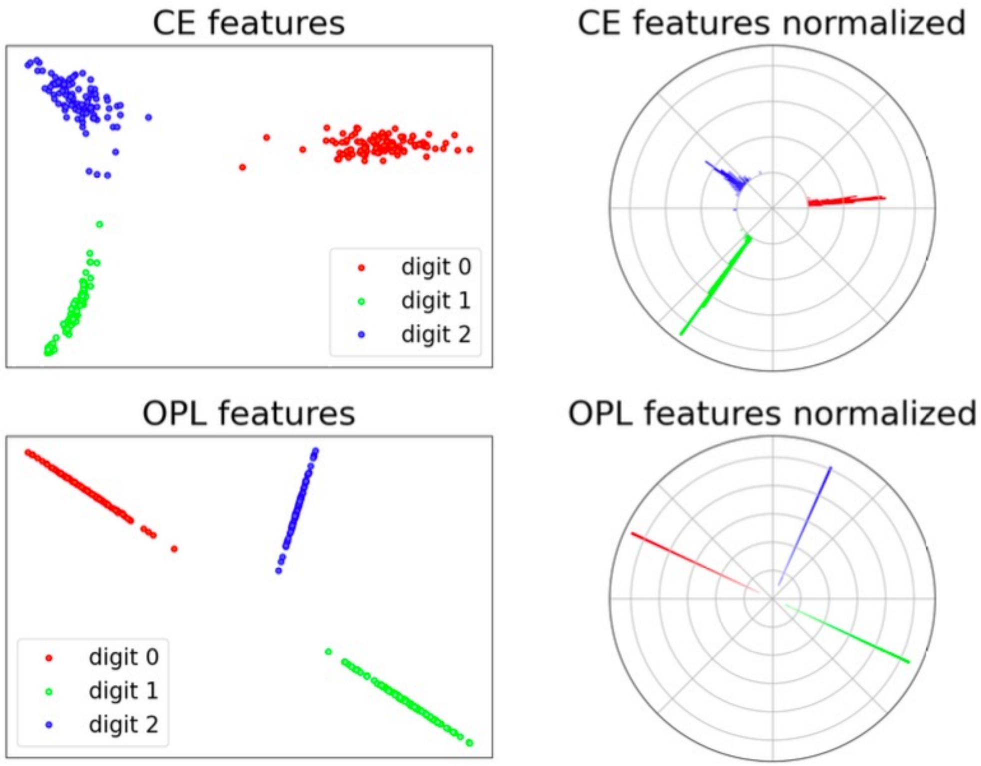
$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{OPL}}$$



- Within each mini-batch, in feature space we make:
  - same class samples similar (s-term)
  - different class samples orthogonal (d-term)

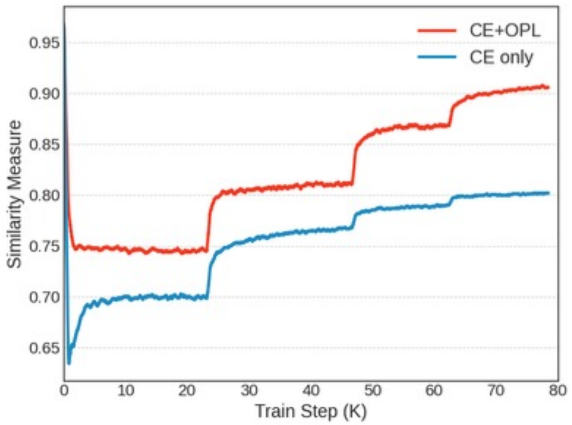
Orthogonality?

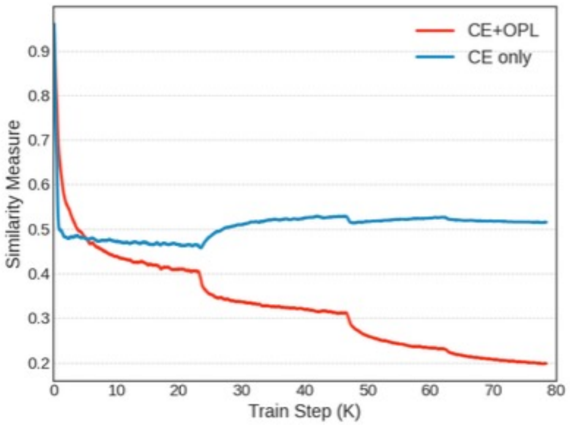
- For random mini-batch based training setting, orthogonality objective provides a definitive geometric structure independent of the batch composition
  - more stable training
  - works with small batch sizes
- Methods like optimal max-margin separation is dependent on the batch composition
- Orthogonality also avoids negative correlation constraints



how OPL changes sample distribution in feature space (left: 3-dim feature space / right: histogram of features)

Training





- OPL has an efficient and straightforward vectorized implementation
- During training, the cosine similarity between same class samples reaches one (left) while that between different class samples reaches zero (right)

Algorithm 1 Pytorch style pseudocode for OPL

```
def forward(features, labels):
    """
    features:   features shaped (B, D)
    labels:     targets shaped (B, 1)
    """
    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
```

Detailed Evaluations

Method	ResNet-18		ResNet-50	
	top-1	top-5	top-1	top-5
CE (Baseline)	69.91%	89.08%	76.15%	92.87%
CE + OPL (ours)	**70.27%**	**89.60%**	**76.98%**	**93.30%**

ImageNet: classification accuracy

Dataset	Method	Clean	Advers.
CIFAR10	Madry *et al.* (ICLR'18) [34]	87.14	44.04
	Madry *et al.* [34] + OPL	**87.76**	**49.15**
	Hendrycks *et al.* (PMLR'19) [19]	87.11	54.92
	Hendrycks *et al.* [19] + OPL	**87.51**	**55.73**
	MART [56] (ICLR'20)	**84.49**	54.10
	MART[56] + OPL	84.41	**56.23**
CIFAR100	Madry *et al.* (ICLR'18) [34]	60.20	20.60
	Madry *et al.* [34] + OPL	**61.13**	**23.01**
	Hendrycks *et al.* (PMLR'19) [19]	59.23	28.42
	Hendrycks *et al.* [19] + OPL	**61.00**	**30.05**
	MART (ICLR'20) [56]	**58.90**	23.40
	MART[56] + OPL	58.01	**25.74**

Adversarial attacks: classification accuracy

Dataset	CNAPs [40] (NeurIPS'19)	SUR [11] (ECCV'20)	SUR + OPL (Ours)
Imagenet	52.3±1.0	56.4±1.2	**56.5±1.1**
Omniglot	88.4±0.7	88.5±0.8	**89.8±0.7**
Aircraft	**80.5±0.6**	79.5±0.8	79.6±0.7
Birds	72.2±0.9	76.4±0.9	**76.9±0.7**
Textures	58.3±0.7	**73.1±0.7**	72.7±0.7
Quick Draw	72.5±0.8	75.7±0.7	**75.7±0.7**
Fungi	47.4±1.0	48.2±0.9	**50.1±1.0**
VGG Flower	86.0±0.5	90.6±0.5	**90.9±0.5**
MSCOCO	42.6±1.1	**52.1±1.0**	52.0±1.0
MNIST	92.7±0.4	93.2±0.4	**94.3±0.4**
CIFAR10	61.5±0.7	66.4±0.8	**66.6±0.7**
CIFAR100	50.1±1.0	57.1±1.0	**57.6±1.0**
Average	67.0	71.4	**71.9**

Few Shot Learning: classification accuracy

Method	Top-1 (%)	Top-5 (%)
ResNet50-D [18]	78.31→**79.26**	94.09→**94.62**

ImageNet: classification accuracy

Detailed Evaluations

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

Few Shot Learning: classification accuracy

Dataset	Method	Uniform	Class Dependent
CIFAR10	TL (NeurIPS'18) [64]	87.62%	82.28%
	TL[64] + OPL	**88.45%**	**87.02%**
CIFAR100	TL (NeurIPS'18) [64]	62.64%	47.66%
	TL[64] + OPL	**65.62%**	**53.94%**

Label Noise: classification accuracy

Method	Art	Cartoon	Sketch	Photo	Avg
JiGen(CVPR'19) [3]	86.20	78.70	70.63	97.66	83.29
MASF(NeurIPS'19) [10]	82.89	80.49	72.29	95.01	82.67
MetaReg(NeurIPS'18) [1]	87.20	79.20	70.30	97.60	83.60
RSC(ECCV'20) [20]	87.89	82.16	83.35	96.47*	87.47
RSC + OPL	**88.28**	**84.64**	**84.17**	**96.83**	**88.48**

Domain Generalization: classification accuracy

Thank You!

Project Page: https://kahnchana.github.io/opl