Unleashing the Power of Contrastive Self-Supervised Visual Models via Contrast-Regularized Fine-Tuning

Yifan Zhang[1], Bryan Hooi[1], Dapeng Hu[1], Jian Liang[2], Jiashi Feng[3]

[1]National University of Singapore [2]Chinese Academy of Sciences [3]SEA AI Lab

October 16, 2021

Contents

Contents

Deep Learning

Deep learning has achieved great success in many computer vision tasks, e.g., image classification, object detection, semantic segmentation

Demands for abundant training data

- Deep neural networks (DNNs) have massive model parameters
- Hence, they require a large number of annotated data for training

Year	Model	#Layers	#Parameter	FLOPs	ImageNet Top-5 Error
2012	AlexNet	5+3	60M	725M	16.4%
2013	Clarifai	5+3	60M	–	11.7%
2014	MSRA	5+3	200M	–	8.06%
2014	VGG-19	16+3	143M	20G	7.32%
2014	GooLeNet	22	6.8M	2G	6.67%
2015	ResNet	152	60.2M	11G	5.79%

Why self-supervision?

- However, supervised learning with large-scale labeled data is impractical
  1. Annotating large-scale data for each new task is very expensive
  2. Some areas are data-starved, e.g., in the medical area
- Some areas have a vast number of unlabeled image/video/graph data (e.g., Facebook: 1 billion images uploaded per day)
- Self-supervised learning is an alternatives to supervised learning

Self-supervised learning

- Self-supervised learning is based on a pre-training and fine-tuning scheme
  1. Pre-train DNNs based on some specific **self-supervised pretext tasks** by using unlabeled data as self-supervision
  2. Transfer the pre-trained DNNs to solve downstream tasks

- Contrastive self-supervised learning has become the most popular one
  1. Pre-train DNNs based on an instance discrimination contrastive task[1]
  2. Transfer the contrastive pre-trained DNNs to solve downstream tasks

[1] Wu, Z., et al. Unsupervised featurelearning via non-parametric instance discrimination. In CVPR, 2018

Research issue

- Despite substantial studies on pre-training, few have explored fine-tuning
- The common practice is to directly fine-tune the pre-trained model with the cross-entropy loss on downstream tasks
- Fine-tuning with only cross-entropy may not be an optimal strategy
  1. Cross-entropy has limited ability to reduce intra-class feature variations
  2. Fine-tuning DNNs with cross-entropy may suffer from overfitting when the data number of downstream task is limited
- Our goal is to investigate how to better fine-tune contrastive self-supervised visual models on downstream tasks

Contents

Contrastive Learning

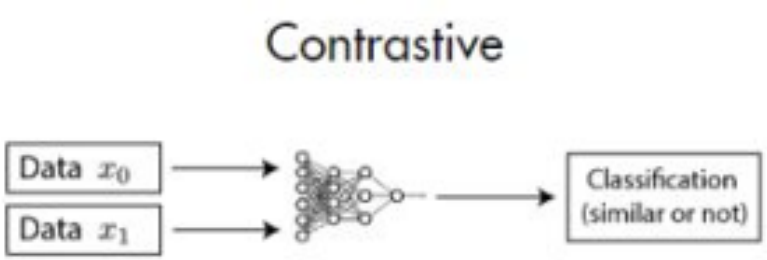
- Learning from **paired data** instead of single data
- Pull positive (similar) pairs closer and push negative (dissimilar) pairs apart
- Positive pairs: two transformations of an image or two images from a class

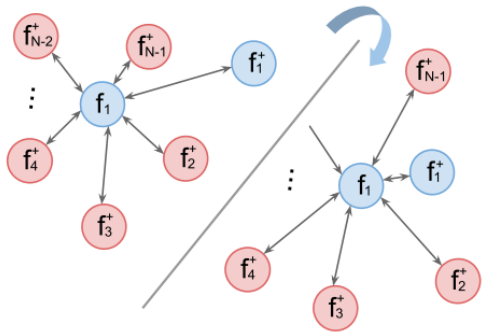
Contrastive self-supervised learning

- Formulate the contrastive learning as a multi-class classification problem
- **Contrastive loss** (InfoNCE loss[2]):

$$L = -\log \frac{\exp(f^\top f^+)}{\exp(f^\top f^+) + \sum_{i=1}^{n-1} \exp(f^\top f_i^-)},$$

where f, f^+, f^- denote the features of the anchor image, the positive pair and negative pairs of the anchor

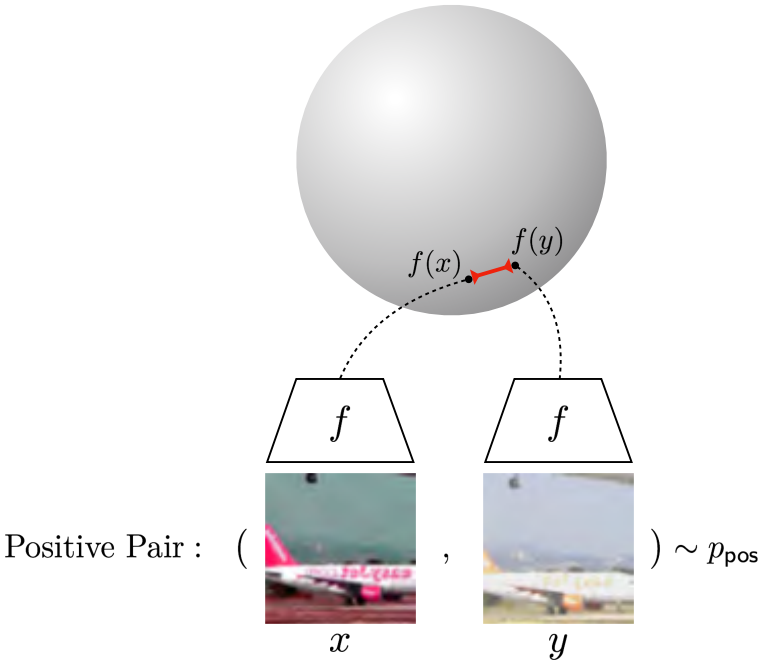




[2] Oord, A. V. D., et al. Representation learning with contrastive predictive coding, arxiv, 2018.

What contrastive self-supervised learning learns?

- There are two findings in existing work[3]:
  1. Instance alignment: similar samples have similar features
  2. Instance uniformity: different data are pushed away



Alignment: Similar samples have similar features. (Figure inspired by Tian et al. (2019).)

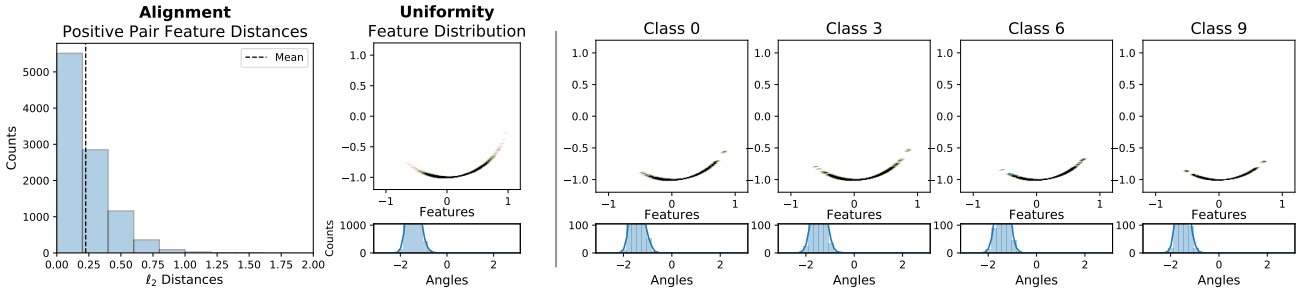
Uniformity: Preserve maximal information.

Figure 1: Illustration of alignment and uniformity of feature distributions on the output unit hypersphere. STL-10 (Coates et al., 2011) images are used for demonstration.

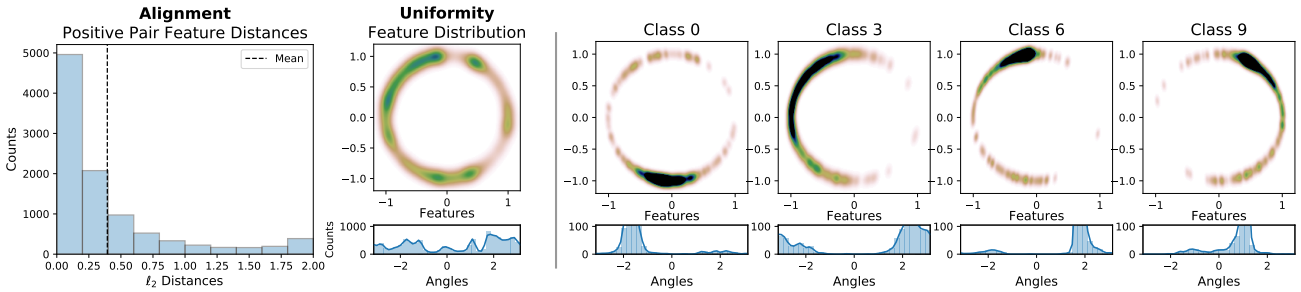
[3] Wang., et al. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere, In ICML, 2020.

What contrastive self-supervised learning learns?

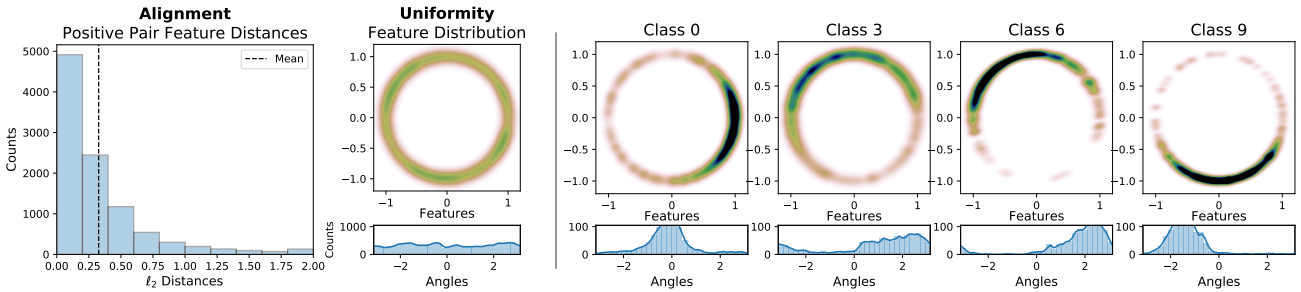
- The learned features are uniformly distributed in the feature space[4]



(a) **Random Initialization.** Linear classification validation accuracy: 12.71%.



(b) **Supervised Predictive Learning.** Linear classification validation accuracy: 57.19%.



(c) **Unsupervised Contrastive Learning.** Linear classification validation accuracy: 28.60%.

[4] Wang., et al. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere, In ICML, 2020.

Why not fine-tune with only cross-entropy?

- Although cross-entropy tends to separate inter-class features, the resulted models still have limited capability of reducing intra-class feature variations

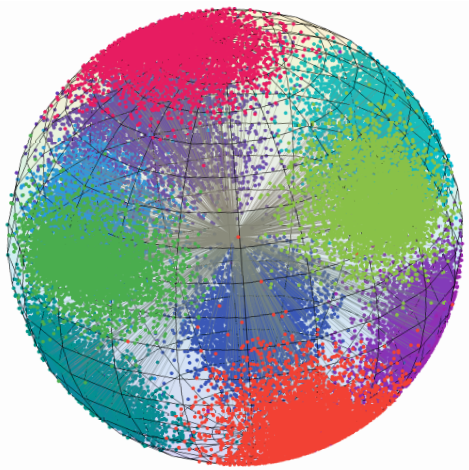


Figure: fine-tuning with only cross-entropy

- How to better fine-tune contrastive self-supervised visual models on downstream tasks?

Contents

Motivations

- **Contrastive self-supervised learning**: optimizing unsupervised contrastive loss during pre-training yields models with instance alignment property
- Whether applying contrastive learning to fine-tuning would bring benefits?
- To answer this, we begin by analyzing supervised contrastive loss:

$$\mathcal{L}_{con} = -\frac{1}{n|P_i|}\sum_{i=1}^{n}\sum_{z_j \in P_i} \log \frac{e^{(z_i^\top z_j/\tau)}}{\sum_{z_k \in A_i} e^{(z_i^\top z_k/\tau)}},$$

where features z are ℓ_2-normalized, and τ is a temperature factor, while P_i and A_i denote the positive pair set and full pair set of the anchor z_i

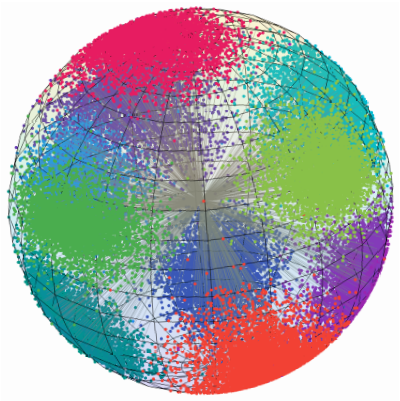
Regularization effectiveness on representation learning

Theorem 1 *Assuming the features are ℓ_2-normalized and the classes are balanced with equal data number, minimizing the contrastive loss is equivalent to minimizing the class-conditional entropy $\mathcal{H}(Z|Y)$ and maximizing the feature entropy $\mathcal{H}(Z)$:*

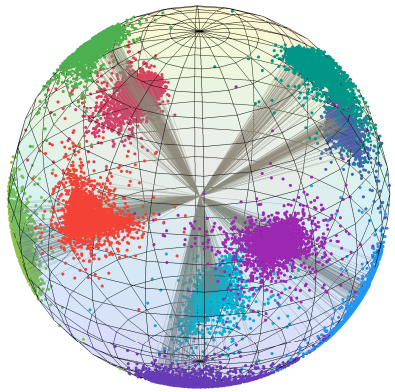
$$\mathcal{L}_{con} \propto \mathcal{H}(Z|Y) - \mathcal{H}(Z)$$

- This theorem shows that $\mathcal{L}_{con}$ explicitly regularizes representation learning
  1. Minimizing $\mathcal{L}_{con}$ will minimize $\mathcal{H}(Z|Y)$, which encourages to learn a low-entropy cluster for each class (i.e., high intra-class compactness)
  2. Minimizing $\mathcal{L}_{con}$ will maximize $\mathcal{H}(Z)$ and tends to learn a high-entropy feature space (i.e., large inter-class separation degree)

Regularization effectiveness on representation learning



(a) Training with $\mathcal{L}_{ce}$



(b) Training with $\mathcal{L}_{ce}+\mathcal{L}_{con}$

- Based on the results, we confirm:
  1. Minimizing $\mathcal{L}_{con}$ encourages to learn high intra-class compactness
  2. Minimizing $\mathcal{L}_{con}$ tends to learn a feature space with large inter-class separation degree

Optimization effectiveness on model training

Theorem 2 *Assuming the features are ℓ_2-normalized and the classes are balanced, the contrastive loss is positive proportional to the infimum of conditional cross-entropy $\mathcal{H}(Y;\hat{Y}|Z)$, where the infimum is taken over classifiers:*

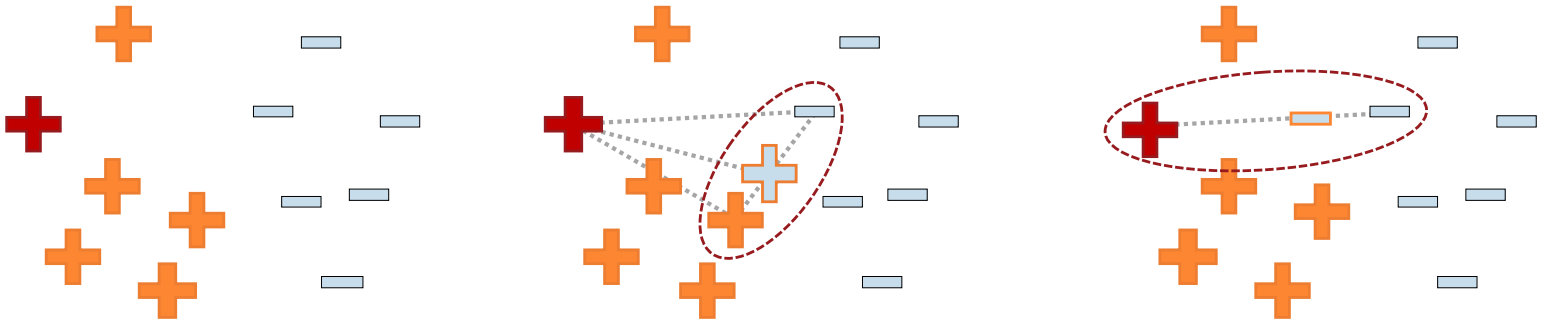
$$\mathcal{L}_{con} \propto \inf \underbrace{\mathcal{H}(Y;\hat{Y}|Z)}_{\textit{Conditional CE}} - \mathcal{H}(Y)$$

- This theorem shows that $\mathcal{L}_{con}$ boosts the model optimization
  - The label Y is given by datasets, so its entropy $\mathcal{H}(Y)$ is a constant
  - Minimizing $\mathcal{L}_{con}$ minimizes the infimum of conditional cross-entropy $\mathcal{H}(Y;\hat{Y}|Z)$
  - Pulling positives together and pushing negatives further apart makes the predicted label distribution closer to the ground truth distribution

Challenges

- Based on the above theoretical analysis, we propose to use contrastive learning to enhance the fine-tuning of contrastive pre-trained models
- Instead of simply adding contrastive loss to the objective function, we further consider **two practical challenges**:
  1. How to mine hard sample pairs for better contrastive fine-tuning
  2. How to improve the generalizability of the model

Challenges I

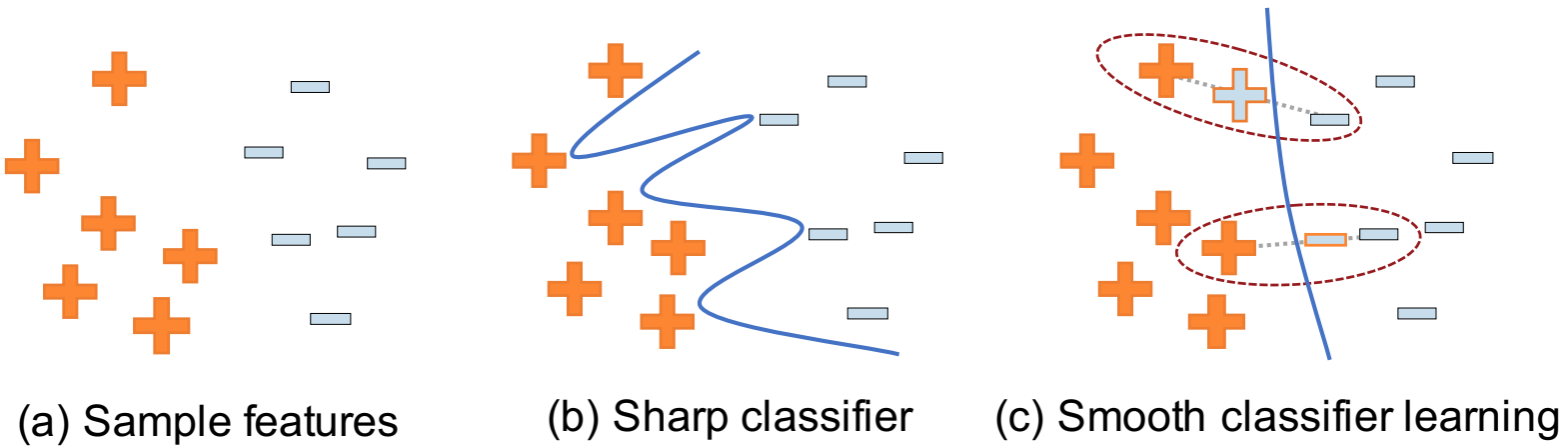


(a) Sample features (b) Hard positive mixup (c) Hard negative mixup

- The majority of sample are easy-to-contrast in training[5], and may produce negligible contrastive loss gradients without performance contributions
- Our solution: to generate both hard positive and hard negative sample pairs based on feature mixup

[5] Hardwood, etc. Smart mining for deep metric learning. In ICCV, 2017.

Challenges II

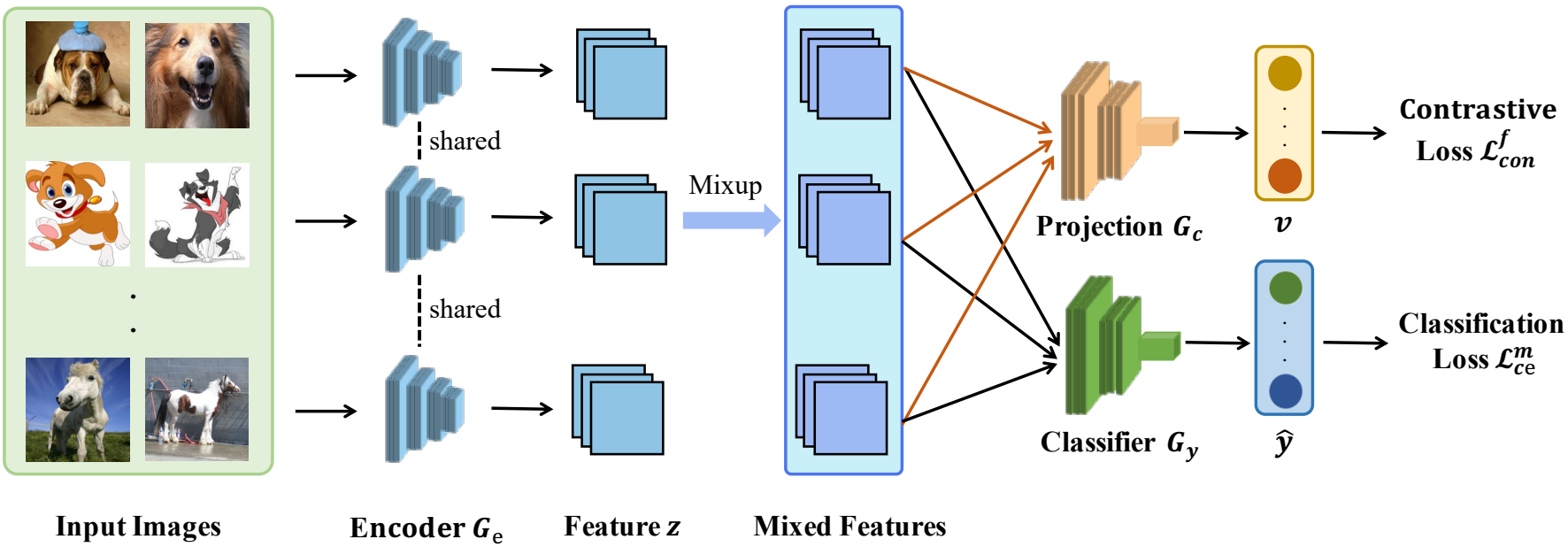


(a) Sample features (b) Sharp classifier (c) Smooth classifier learning

- The decision boundary trained with cross-entropy is often sharp and close to training data[6], especially when the data number is limited
- This may lead to incorrect yet confident predictions when evaluated on slightly different test samples
- Our solution: to smooth classifiers by using the mixed features

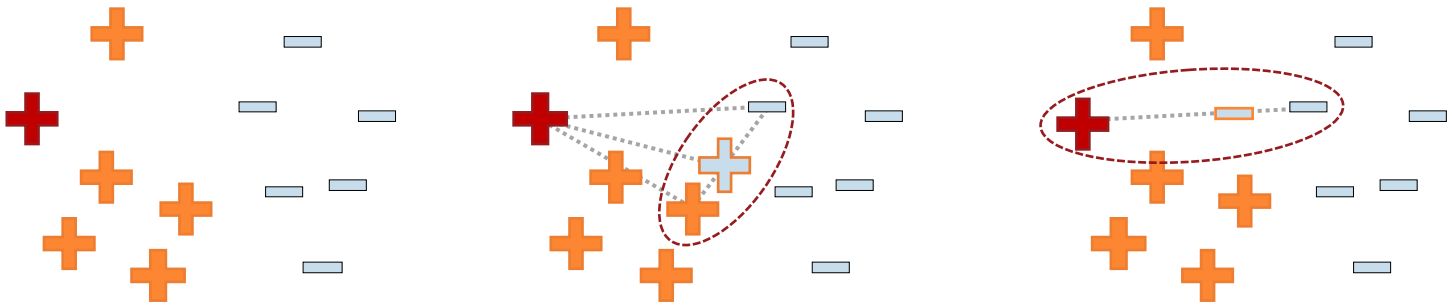
[6] Bengio, Y., etc. Manifold mixup: Better rep-resentations by interpolating hidden states. In ICML, 2019.

Overall scheme



- We propose a contrast-regularized tuning (Core-tuning) method
- Core-tuning consists of two main components:
  1. Hard sample pair mining
  2. Smooth classifier learning

Hard positive sample pair generation



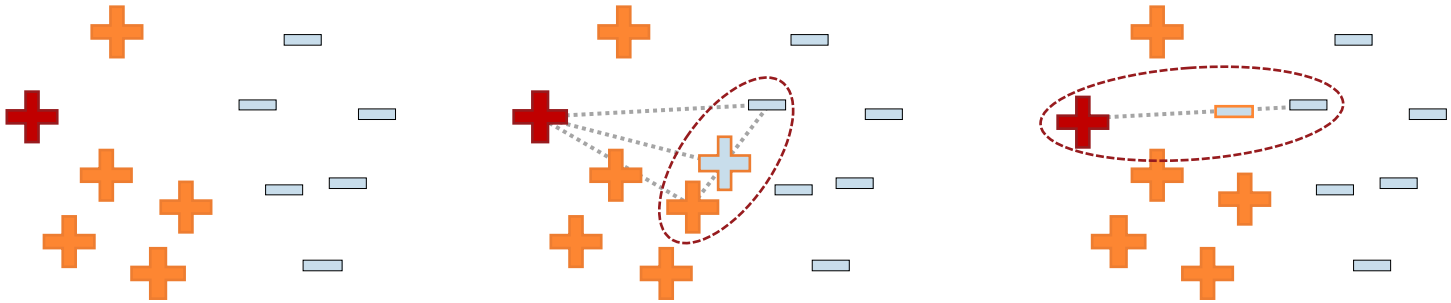
(a) Sample features (b) Hard positive mixup (c) Hard negative mixup

- Given a feature anchor z_i, we first find out its hardest positive sample z_i^{hp} and hardest negative sample z_i^{hn} based on cosine similarity
- We generate hard positive pair as a convex combination of z_i^{hp} and z_i^{hn}:

$$z_i^+ = \lambda z_i^{hp} + (1-\lambda) z_i^{hn}$$

- $\lambda \sim \text{Beta}(\alpha, \alpha) \in [0, 1]$, and α is a parameter to decide the beta distribution

Hard negative sample pair generation



(a) Sample features (b) Hard positive mixup (c) Hard negative mixup

- Given a feature anchor z_i, we randomly select a negative sample (z_i^n, y_i^n) to synthesize the hard negative pair:

$$z_i^- = (1-\lambda)z_i + \lambda z_i^n; \qquad y_i^- = (1-\lambda)y_i + \lambda y_i^n$$

- We clip $\lambda \geq 0.8$ to make the generated negative closer to negatives
- Hardness: the generated pairs are located between positives/negatives

Hard pair reweighting

- We first use a two-layer MLP head G_c to obtain ℓ_2-normalized contrastive features $v_i{=}G_c(z_i)/\|G_c(z_i)\|_2$
- Since hard positives are more informative for contrastive learning, we propose to assign higher importance weights to them
- Inspired by the focal loss[7], we find hard positive pairs generally lead to a low prediction probability: $p_{ij}{=}\frac{\exp(v_i^\top v_j/\tau)}{\sum_{v_k\in A_i}\exp(v_i^\top v_k/\tau)}$
- Focal contrastive loss:

$$\mathcal{L}_{con}^{f}{=}-\frac{1}{n|P_i|}\sum_{i=1}^{n}\sum_{v_j\in P_i}(1{-}p_{ij})\log\frac{e^{(v_i^\top v_j/\tau)}}{\sum_{v_k\in A_i}e^{(v_i^\top v_k/\tau)}},$$

where P_i and A_i denote the anchor's positive pair set and full pair set, both of which contain the generated hard pairs

[7] Lin, T-T, etc. Focal loss for dense object detection. In ICCV, 2017

Smooth classifier learning

- The classifier trained with cross-entropy is often sharp and close to data, leading to limited generalization performance
- Inspired by that mixup helps learn a smoother classifier[8], we further use the mixed data from the generated hard pair set $\mathcal{B}$ for training

$$\mathcal{L}_{ce}^{m} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) - \frac{1}{|\mathcal{B}|}\sum_{(z_j, y_j)\in\mathcal{B}} y_j \log(G_y(z_j))$$

[8] Bengio, Y., etc. Manifold mixup: Better representations by interpolating hidden states. In ICML, 2019.

Contents

Experiment objectives

1. Verify Core-tuning on image classification and semantic segmentation
2. Evaluate how Core-tuning affects model generalization to new domains and model robustness to adversarial examples

Ablation studies on Image Classification

1. Each component in Core-tuning improves the performance
2. Mixup (row 3) is expected to outperform mixup-hard w.r.t. classification, but Core-tuning still shows obvious improvement via contrast regularizer

$\mathcal{L}_{ce}$	$\mathcal{L}_{con}$	$\mathcal{L}_{con}^{f}$	mixup	mixup-hard	ImageNet-20	CIFAR10	CIFAR100	Caltech101	DTD	Aircraft	Cars	Pets	Flowers	Avg.
√					88.28	94.70	80.27	91.87	71.68	86.87	88.61	89.05	98.49	87.76
√	√				89.29	95.33	81.49	92.84	72.73	87.44	89.37	89.71	98.65	88.54
√			√		90.67	95.43	81.03	92.68	73.31	88.37	89.06	91.37	98.74	88.96
√	√			√	92.20	97.01	83.89	93.22	74.78	88.88	89.79	91.95	98.94	90.07
√		√		√	**92.59**	**97.31**	**84.13**	**93.46**	**75.37**	**89.48**	**90.17**	**92.36**	**99.18**	**90.45**

Results on Image Classification

1. MoCo-v2 pre-trained models perform worse than supervised ones
2. Fine-tuning methods for supervised pre-trained models cannot perform well
3. Using contrastive loss for fine-tuning may also performs unsatisfactory
4. Core-tuning achieves state-of-the-art fine-tuning performance

Method	ImageNet-20	CIFAR10	CIFAR100	Caltech101	DTD	Aircraft	Cars	Pets	Flowers	Avg.
SL-CE-tuning	91.01	94.23	83.40	93.39	74.40	87.03	89.77	92.17	98.78	89.35
CE-tuning	88.28	94.70	80.27	91.87	71.68	86.87	88.61	89.05	98.49	87.76
L2SP (Li et al., 2018)	88.49	95.14	81.43	91.98	72.18	86.55	89.00	89.43	98.66	88.10
M&M (Zhan et al., 2018)	88.53	95.02	80.58	92.91	72.43	87.45	88.90	89.60	98.57	88.22
DELTA (Li et al., 2019)	88.35	94.76	80.39	92.19	72.23	87.05	88.73	89.54	98.65	87.99
BSS (Chen et al., 2019)	88.34	94.84	80.40	91.95	72.22	87.18	88.50	89.50	98.57	87.94
RIFLE (Li et al., 2020)	89.06	94.71	80.36	91.94	72.45	87.60	89.72	90.05	98.70	88.29
SCL (Gunel et al., 2021)	89.29	95.33	81.49	92.84	72.73	87.44	89.37	89.71	98.65	88.54
Bi-tuning (Zhong et al., 2021)	89.06	95.12	81.42	92.83	73.53	87.39	89.41	89.90	98.57	88.58
Core-tuning (ours)	**92.59**	**97.31**	**84.13**	**93.46**	**75.37**	**89.48**	**90.17**	**92.36**	**99.18**	**90.45**

Results on semantic segmentation

- Core-tuning contributes to fine-tuning performance of all pre-trained models
- Self-supervised models have already outperformed supervised ones
- This inspires us to explore unsupervised contrastive regularizers in the future

Pre-training	Fine-tuning	MPA	FWIoU	MIoU
Supervised	CE	87.10+/-0.20	89.12+/-0.17	76.52+/-0.34
InsDis	CE	83.64+/-0.12	88.23+/-0.08	74.14+/-0.21
	ours	**84.53+/-0.31**	**88.67+/-0.07**	**74.81+/-0.13**
PIRL	CE	83.16+/-0.26	88.22+/-0.24	73.99+/-0.42
	ours	**85.30+/-0.24**	**88.95+/-0.08**	**75.49+/-0.36**
MoCo-v2	CE	87.31+/-0.31	90.26+/-0.12	78.42+/-0.28
	ours	**88.76+/-0.34**	**90.75+/-0.04**	**79.62+/-0.12**
SimCLR-v2	CE	87.37+/-0.48	90.27+/-0.12	78.16+/-0.19
	ours	**87.95+/-0.20**	**90.71+/-0.13**	**79.15+/-0.33**
InfoMin	CE	87.17+/-0.20	89.84+/-0.09	77.84+/-0.24
	ours	**88.92+/-0.36**	**90.65+/-0.09**	**79.48+/-0.30**

Effectiveness on cross-domain generalization

- Enforcing contrastive regularizer improves generalization performance
- Core-tuning further improves the generalization performance

Pre-training	Fine-tuning	PACS					VLCS				
		A	C	P	S	Avg.	C	L	V	S	Avg.
Supervised	CE	83.65	79.21	96.11	81.46	85.11	98.41	63.81	68.55	75.45	76.56
MoCo-v2	CE	78.71	76.92	90.87	75.67	80.54	94.96	66.87	68.96	64.98	73.94
	CE-Con	85.11	81.77	95.58	80.12	85.65	95.94	67.76	69.31	73.57	77.67
	ours	**87.31**	**84.06**	**97.53**	**83.43**	**88.08**	**98.50**	**68.19**	**73.15**	**81.53**	**80.34**

Robustnesss to adversarial samples

- We generate adversarial samples via projected gradient descent (PGD)[9]
- Core-tuning is beneficial to model robustness on various image datasets
- We hope that Core-tuning can motivate people to rethink the accuracy-robustness trade-off in adversarial training

Method	ℓ_2-attack						ℓ_∞-attack					
	ϵ=0.5		ϵ=1.5		ϵ=2.5		ϵ= 2/255		ϵ= 4/255		ϵ= 8/255	
	Robust	Clean	Robust	Clean	Robust	Clean	Robust	Clean	Robust	Clean	Robust	Clean
CE	50.25	94.70	48.29	94.70	46.82	94.70	25.13	94.70	12.28	94.70	4.57	94.70
AT-CE	86.59	92.00	89.60	94.28	89.16	94.15	83.20	93.05	75.82	91.99	69.27	92.79
AT-CE-Con	90.74	94.71	90.29	94.80	89.70	94.27	85.07	94.56	79.75	93.79	70.70	93.38
AT-ours	**92.97**	**96.82**	**92.32**	**96.90**	**92.05**	**96.87**	**86.92**	**96.29**	**82.01**	**95.95**	**74.83**	**95.90**

[9] Madry, A., et al. Towards deep learning models resistant to adversarial attacks. In ICLR, 2018

Contents

Summary

- This work studied how to fine-tune contrastive self-supervised visual models
- There are three main contributions:
  1. We propose a simple yet effective contrast-regularized tuning method
  2. We analyze the benefits of supervised contrastive loss to fine-tuning
  3. We empirically show that Core-tuning effectively improve the fine-tuning performance of contrastive self-supervised models
- We call for more attention to the fine-tuning of contrastive self-supervised models on understanding its underlying theories and better approaches
- Source code is available: `https://github.com/Vanint/Core-tuning`.

Thanks!