Non-negative Contrastive Learning

Yifei Wang, *Postdoc at CSAIL*

Based on joint work with Qi Zhang, Yaoyu Guo, Yisen Wang

2024.04

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

ICLR

Takeaway: an one-line trick (to try on your own task!)

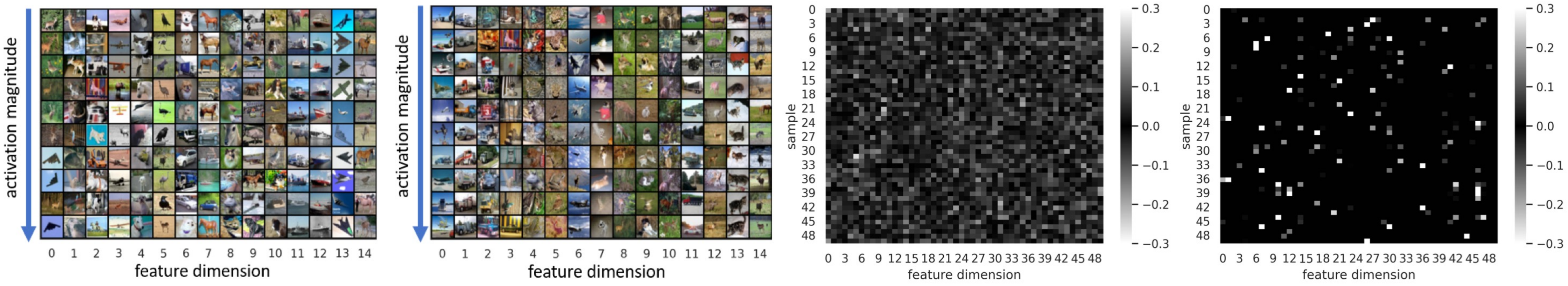
Contrastive learning (CL) obtains feature vectors, eg $[0.3, -0.2, 0.01, -0.5]$

that are **non-interpretable, non-sparse, and entangled**

Our fix: convert it to Non-negative Contrastive Learning (NCL) by adding one line of code at the last layer output

```
z = torch.nn.functional.relu(z)
```

✅ more disentangled ✅ sparser



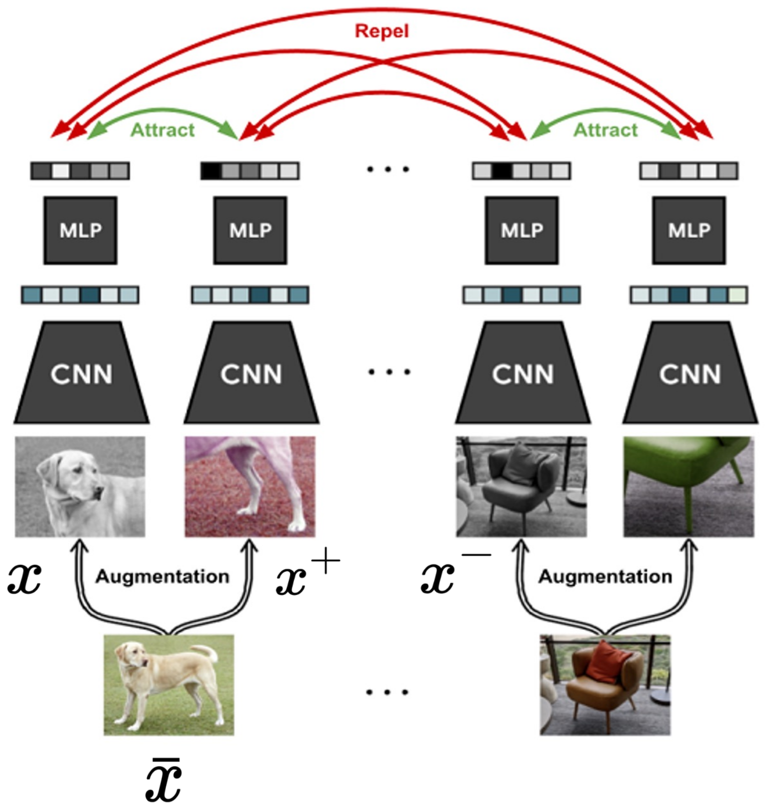
(a) CL (b) NCL (c) CL (d) NCL

This Talk

- How feature non-interpretability happens in CL
- Revisiting Non-negative Matrix Factorization (NMF) as a cue
- Non-negative Contrastive Learning as a modern variant of NMF
- Benefits of NCL in real-world applications

Computer Science & Artificial Intelligence Laboratory

Contrastive Learning: It Takes Two to Tango



One of the SOTA methods for **vision SSL (SimCLR, DINO),** vision-language learning (CLIP), NLP (sentence embedding)

Positive pairs (x, x^+): augmented from the same sample

Negative pairs (x, x^-) : augmented from different samples

Most popular contrastive loss: **InfoNCE** (Oord et al., 2018)

$$\mathcal{L}_{\mathrm{NCE}}(f) = -\mathbb{E}_{x,x_+,\{x_i^-\}_{i=1}^M} \log \frac{\exp(f(x)^\top f(x_+))}{\exp(f(x)^\top f(x_+)) + \sum_{i=1}^M \exp(f(x)^\top f(x_i^-))},$$

cross-entropy loss with sample features replacing class centers

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

Contrastive Learning "is" Matrix Factorization

The augmentation $A(\cdot|\bar{x})$ induces a joint probability in the sample space $\mathcal{X}$ (assume finite size N)

$$\mathcal{P}(x, x') = \mathbb{E}_{\bar{x}} \mathcal{A}(x \mid \bar{x}) \mathcal{A}(x' \mid \bar{x}), \forall\, x, x' \in \mathcal{X}$$

$P \in \mathbb{R}^{N \times N}$ is the co-occurrence matrix under aug. Let $\bar{A} = D^{-1/2} P D^{-1/2}$ denote the normalized P.

Haochen et al. (2021): (spectral) contrastive loss = matrix factorization loss

spectral contrastive loss: $\mathcal{L}_{\mathrm{sp}}(f) = -2\mathbb{E}_{x,x_+} f(x)^\top f(x_+) + \mathbb{E}_{x,x^-}(f(x)^\top f(x_i^-))^2.$

a slight different loss on negative samples

matrix factorization: $\mathcal{L}_{\mathrm{MF}}(F) = \left\| \bar{A} - FF^\top \right\|^2 \quad F \in \mathbb{R}^{n \times d}$

equivalent under $F_{x,:} = \sqrt{\mathcal{P}(x)} f(x)$

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

An MF perspective of CL's non-interpretability

Assume that features are unconstrained (UFM)

The optimal solution F* is characterized by the eigendecomposition of $\bar{A} = U\Sigma U^\top$

$F^* = U\Sigma^{1/2}R$, where R can be any rotation matrix

even if there is a good disentangled (axis-aligned) features F, FR is also optimal

because of this ambiguity, CL cannot find the disentangled one

conclusion: rotation symmetry *hurts* feature interpretability & disentanglement

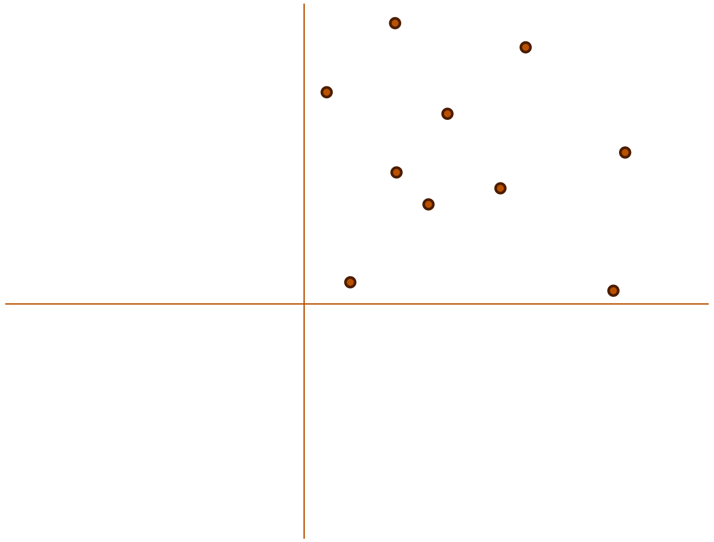
Computer Science & Artificial Intelligence Laboratory

Breaking the rotation symmetry

Tools from the classic literature: Non-negative Matrix Factorization (NMF) (90s – now)

$$\mathcal{L}_{\mathrm{NMF}}(F) = \|\bar{A} - F_+ F_+^\top\|^2, \text{ where } F_+ \geq 0.$$

Simple intuition: enforcing features within the positive plane, so features cannot be **arbitrarily** rotated



Uniqueness: under further assumptions/regularizations (*extensively studied in NMF*),

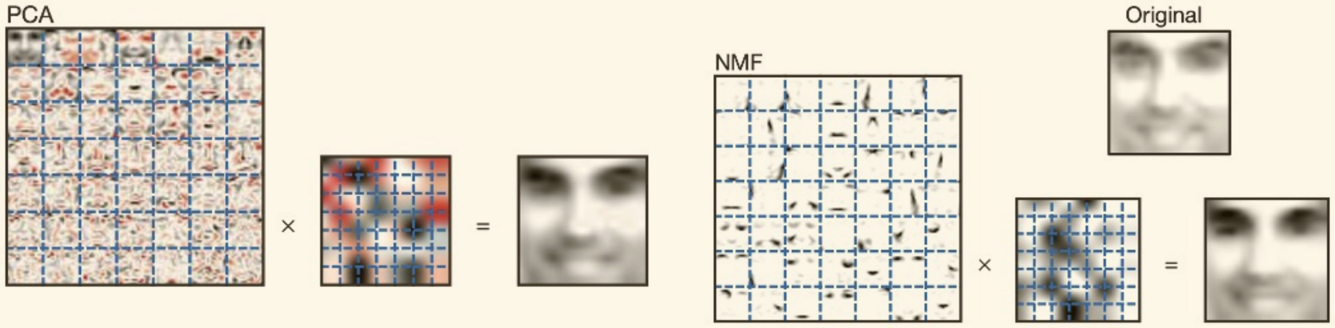
NMF solutions are unique up to axis permutations (which do not break disentanglement!)

Computer Science & Artificial Intelligence Laboratory

NMF yields sparse and disentangled features

Even without uniqueness guarantees, NFM still works pretty well in practice

The seminal work Lee & Seung (Nature, 97)



❌ PCA (MF) gives non-interpretable filter banks

✅ NMF features are local, sparse, and interpretable

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

NMF was a SOTA ML method

Non-negative Contrastive Learning

$$\mathcal{L}_{\rm NMF}(F) = \|\bar{A} - F_+F_+^\top\|^2, \text{ where } F_+ \geq 0.$$

The co-occurrence matrix A is also non-negative. Let us do NMF for SSL then!

Two key problems:
- A is ***unknown*** (we only have samples from the underlying distribution)
- A is **exponentially large** (NxN, N is #samples) – any matrix operator is prohibitive

equivalent!

Our solution: convert NMF back to a **sampling-based objective**

Non-negative Contrastive Learning (NCL)

$$\mathcal{L}_{\rm NCL} = -2\mathbb{E}_{x,x_+} f_+(x)^\top f_+(x_+) + \mathbb{E}_{x,x^-} \left(f_+(x)^\top f_+(x^-)\right)^2,$$
$$\textit{such that } f_+(x) \geq 0, \forall x \in \mathcal{X}.$$

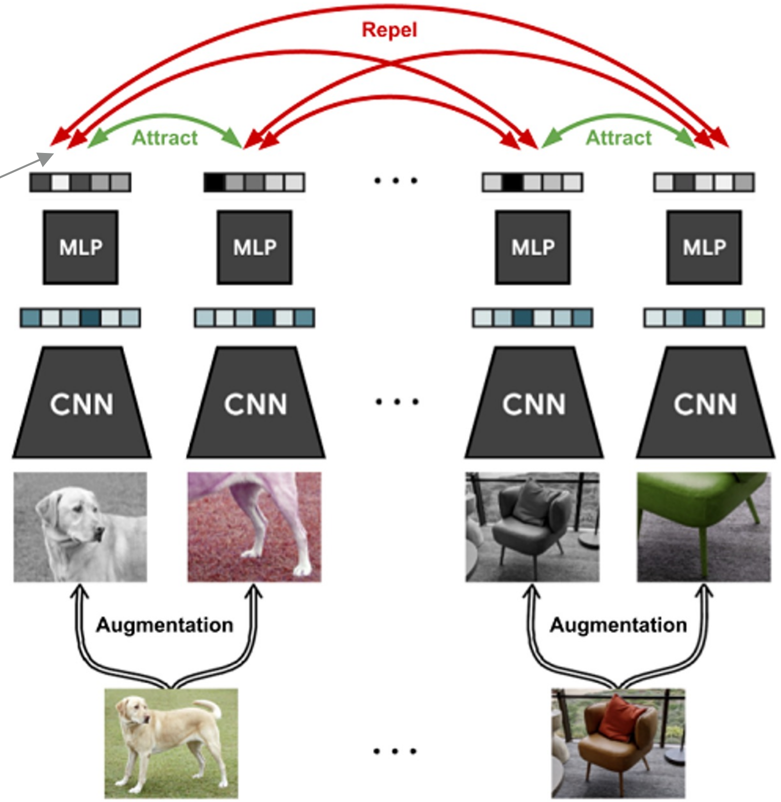
MIT CSAIL Computer Science & Artificial Intelligence Laboratory

Non-negative reparameterization

Solving a constrained problem with NN is hard

a simple reparameterization trick: just use a conventional NN, and apply a **non-negative transformation σ_+** **at last**

$$f_+(x) = \sigma_+(f(x)).$$



We've tried sigmoid, softplus, relu, exp; even leaky relu, gelu

- non-negativity is critical (leakly relu and gelu are way worse)
- relu is better, since it induces better sparsity

Computer Science & Artificial Intelligence Laboratory

Theoretical Justifications (a glimpse)

subclasses in "cars"

- As in Arora et al. (2019), we assume that positives are sampled from the same *latent class c*

Assumption 1 (Positive Generation). $\forall x, x' \in \mathcal{X}, \mathcal{P}(x, x') = \mathbb{E}_c \mathcal{P}(x|c)\mathcal{P}(x'|c)$.

The optimal representation of NCL:

$$\phi(x) = \left[\frac{1}{\sqrt{\mathcal{P}(\pi_1)}} \mathcal{P}(\pi_1|x), \dots, \frac{1}{\sqrt{\mathcal{P}(\pi_m)}} \mathcal{P}(\pi_m|x) \right] \in \mathbb{R}_+^m, \forall\, x \in \mathcal{X},$$

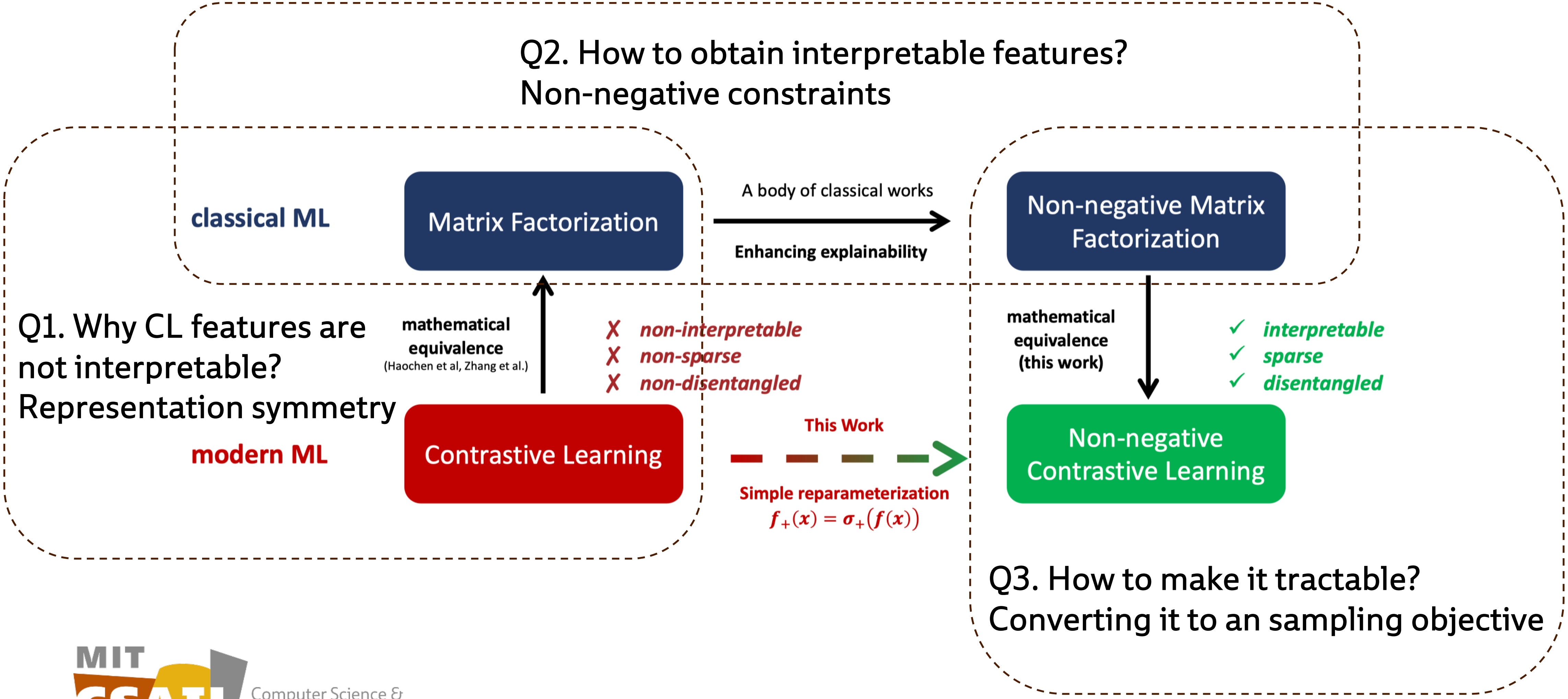
$[\pi_1, \dots, \pi_m]$ **is a random permutation of latent classes** $[c_1, \dots, c_m]$.

That is, the feature values directly represent the posterior distribution on latent classes

Based on this nice property, we further establish guarantees on its sparsity, disentanglement, and downstream classification error (more in paper)

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

Wrap up



Computer Science & Artificial Intelligence Laboratory

Comparing NMF and NCL

$$\mathcal{L}_{\mathrm{NMF}}(F) = \|\bar{A} - F_+F_+^\top\|^2, \text{ where } F_+ \geq 0.$$

$$\mathcal{L}_{\mathrm{NCL}} = -2\mathbb{E}_{x,x_+} f_+(x)^\top f_+(x_+) + \mathbb{E}_{x,x^-}\left(f_+(x)^\top f_+(x^-)\right)^2$$

If they are equivalent, why NCL is better than conventional NMF?

NCL performs NMF implicitly with benefits in many ways:

Method	A (data)	F (features)	Solver
NMF	Explicit similarity based on distance (eg, L2, kernels)	Explicit Matrix	Multiplicative update, Projected GD, etc
Limitations	Not working for high-dim data	Not scalable; transductive	Explicit matrix operations & constrained opt
NCL	**Implicit similarity based on sampling**	**Amortized via NNs**	**Reparameterized with NN + ReLU; SGD training**
Benefits	Inject domain knowledges via augmentation design	Expressive, scalable, inductive (generalize to new data)	Scalable, unconstrained, fully differentiable

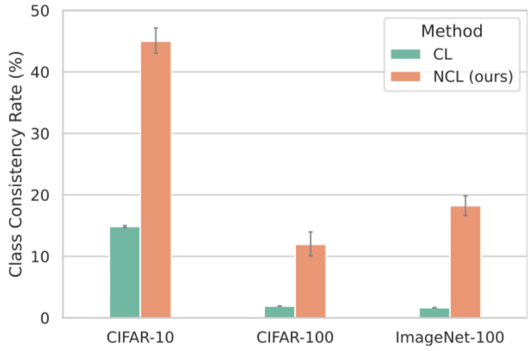
MIT CSAIL Computer Science & Artificial Intelligence Laboratory

NCL makes NMF great again by merging it with modern SSL innovations

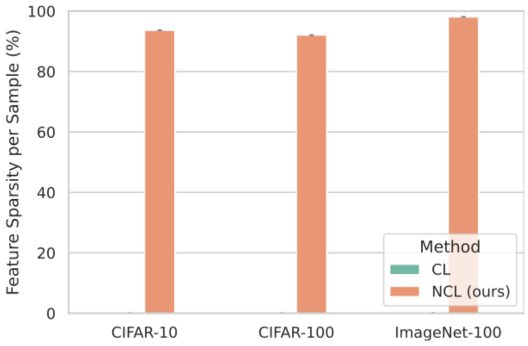
Real-world Experiments

Quantitative comparison on feature properties

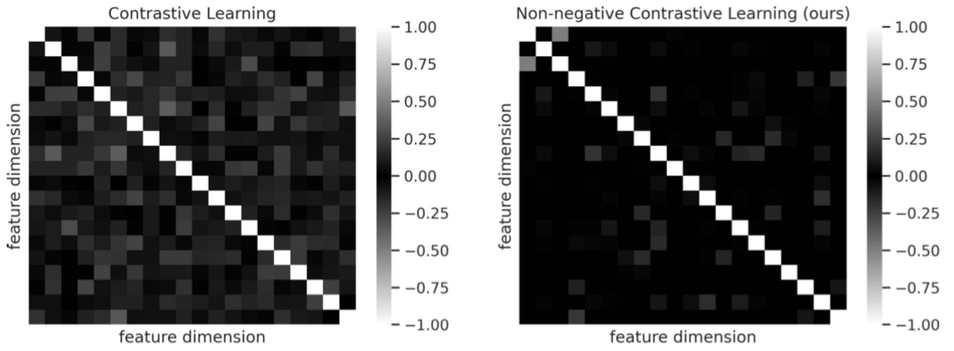
- **semantic consistency**: ratio of activated samples from the same class along each dimension
- **sparsity**: ratio of zero elements of each sample – more than 90% are zeros in NCL
- **correlation**: correlation among different feature dimensions



(a) Semantic Consistency



(b) Feature Sparsity



(c) Feature Correlation

Computer Science & Artificial Intelligence Laboratory

Transfer learning (SSL-> downstream classification)

Two common evaluation protocols:

- LP: linear probing (train a linear classifier on top of frozen learned features)
- FT: full finetuning the entire model with learned initialization

ImageNet-100

(a) in-distribution evaluation

Method	CIFAR-100		CIFAR-10		ImageNet-100	
	LP	FT	LP	FT	LP	FT
CL	58.6 ± 0.2	72.6 ± 0.1	87.6 ± 0.2	92.3 ± 0.1	68.7 ± 0.3	77.3 ± 0.5
NCL	$\mathbf{59.7 \pm 0.4}$	$\mathbf{73.0 \pm 0.2}$	$\mathbf{87.8 \pm 0.2}$	$\mathbf{92.6 \pm 0.1}$	$\mathbf{69.4 \pm 0.3}$	$\mathbf{79.2 \pm 0.4}$

(b) out-of-distribution transferability

Method	Stylized	Corruption	Sketch
CL	19.6 ± 0.4	34.5 ± 0.2	27.1 ± 0.1
NCL	$\mathbf{21.2 \pm 0.2}$	$\mathbf{36.1 \pm 0.3}$	$\mathbf{28.0 \pm 0.2}$

Consider that we only add a ReLU to the output, the improvement is quite favorable

Computer Science & Artificial Intelligence Laboratory

Feature Disentanglement

- Score: SEPIN@k (k: number of features, Do & Tran, 2020)
- Significant improvement on disentanglement

	SEPIN@1	SEPIN@10	SEPIN@100	SEPIN@all
CL	0.88 ± 0.08	0.79 ± 0.02	0.69 ± 0.01	0.47 ± 0.01
NCL	$\mathbf{7.43 \pm 0.15}$	$\mathbf{5.93 \pm 0.12}$	$\mathbf{3.87 \pm 0.04}$	$\mathbf{0.48 \pm 0.01}$

ImageNet-100

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

Feature Selection

Goal: select 512 features out of 2048 features and maintain its performance

Branded as *"shortening embedding"* in OpenAI API recently for faster inference

NCL admits a natural way to select important features based on their average activation

hypothesis: more frequently activated features are more common / important

ImageNet-100

Selection	Linear Probing		Image Retrieval		Transfer Learning	
	CL	**NCL**	CL	**NCL**	CL	**NCL**
All (2048)	66.8 ± 0.2	**68.9 ± 0.1**	10.9 ± 0.2	**14.2 ± 0.2**	17.2 ± 0.1	**19.9 ± 0.1**
Random (512)	66.2 ± 0.1 (-0.6)	64.3 ± 0.2 (-5.6)	10 ± 0.1 (-0.9)	8.2 ± 0.1 (-6.0)	16.6 ± 0.2 (-0.6)	16.7 ± 0.1 (-3.2)
EA (512, w/o ReLU)	66.3 ± 0.2 (-0.5)	66.7 ± 0.1 (-2.2)	9.9 ± 0.21 (-0.9)	11.1 ± 0.2 (-3.1)	16.5 ± 0.3 (-0.7)	17.7 ± 0.2 (-2.2)
EA (512, w/ ReLU) (ours)	66.5 ± 0.1 (-0.3)	**68.9 ± 0.3** (-0.0)	10.2 ± 0.2 (-0.7)	**14.2 ± 0.2** (-0.0)	16.6 ± 0.3 (-0.6)	**19.8 ± 0.1** (-0.1)

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

1. NCL is better using all features
2. NCL also has less or no drop with 512/2048 features

Extension to Broader Scenarios

- Contrastive objectives have broad applications
  - graph, text, multi-modal learning, supervised learning
  - NCL can be applied too
- Supervised learning with Non-negative Cross Entropy (NCE)
  - based on the essential view that CE loss is a special CL loss

$$\mathcal{L}_{\mathrm{CE}}(f) = -\mathbb{E}_{x,y} \log \frac{\exp\big(f(x)^\top w_y\big)}{\sum_{c=1}^{C} \exp\big(f(x)^\top w_c\big)}$$

- Imagenet-100 experiments: ~2x faster training at early stage & 3% higher final performance

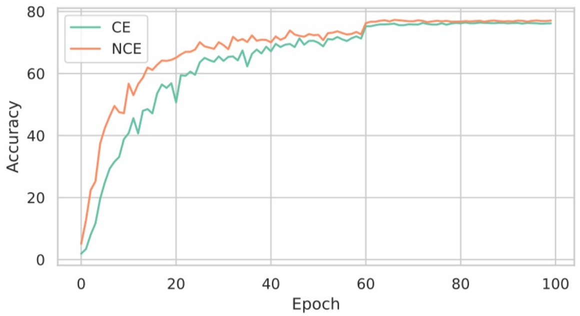


Figure 4: Training from scratch with CE and NCE (w/o projector) on ImageNet-100.

Loss	From Scratch	Finetune
CE	76.1	78.6
NCE	78.6	80.2
CE + MLP projector	78.4	81.1
NCE + MLP projector	**79.2**	**82.0**

Table 4: Test accuracy (%) of CE and NCE losses for supervised learning on ImageNet-100.

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

Summary

- CL features suffer from non-interpretability due to representation symmetry
- Symmetry breaking with NMF
- Non-negative Contrastive Learning as implicit NMF
- NCL attains comparable and even better performance than CL

more benefits are yet to be discovered!

Computer Science & Artificial Intelligence Laboratory

Thank you!

MIT CSAIL
Computer Science &
Artificial Intelligence
Laboratory