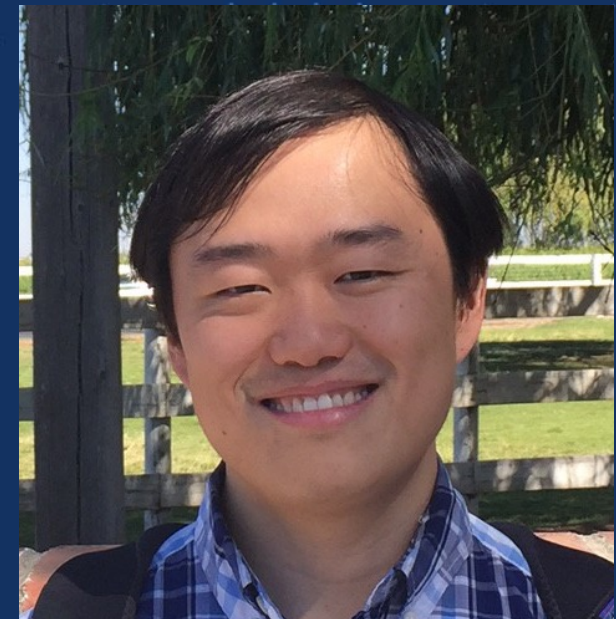


Toward Better Understanding of Deep Contrastive Learning

Yuandong Tian

Research Scientist and Manager

Meta AI (FAIR)

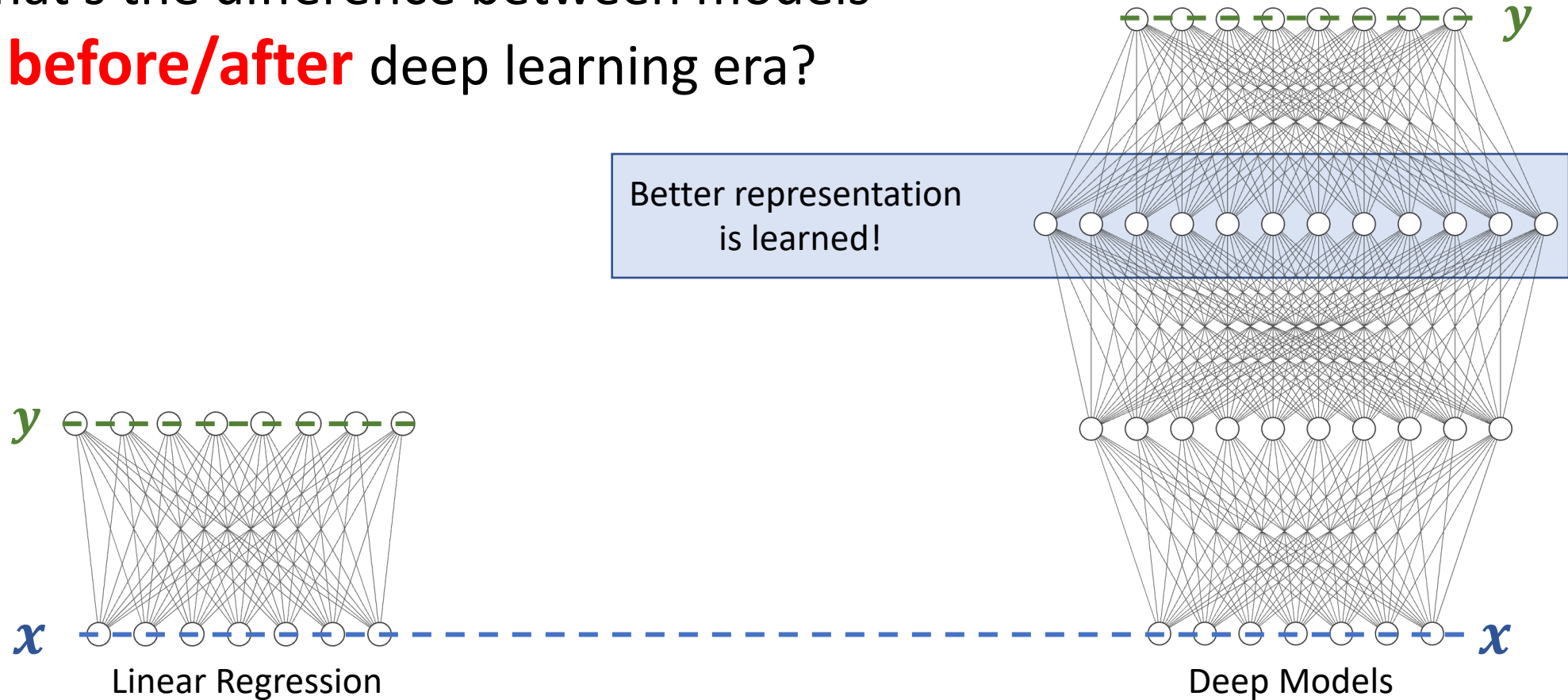


Great Empirical Success of Deep Models

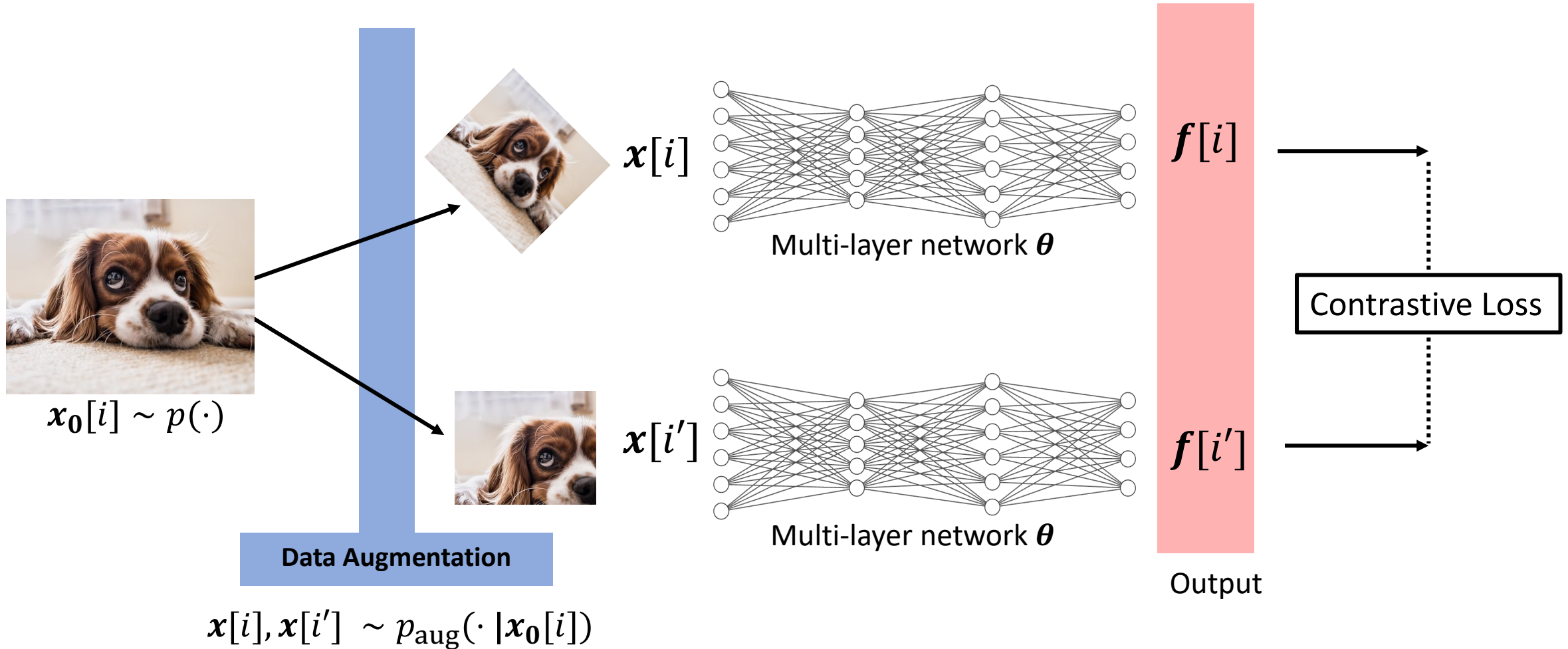


Representation Learning

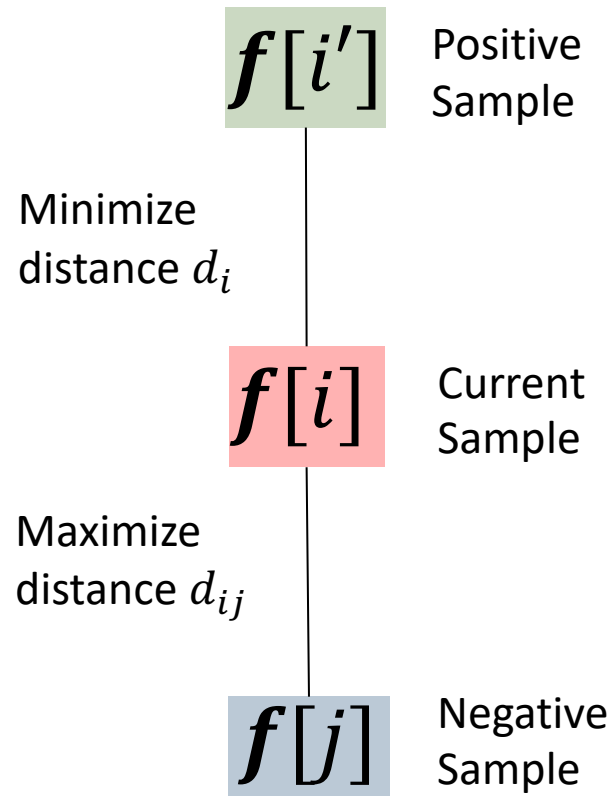
What's the difference between models
before/after deep learning era?



Contrastive Learning (CL) is Popular



Formulation of Contrastive Learning



InfoNCE loss:

$$\mathcal{L}_{nce} := -\tau \sum_{i=1}^N \log \frac{\exp(-d_i^2/\tau)}{\epsilon \exp(-d_i^2/\tau) + \sum_{j \neq i} \exp(-d_{ij}^2/\tau)}$$

$$\text{Intra-view distance } d_i^2 = \|f[i] - f[i']\|_2^2/2$$

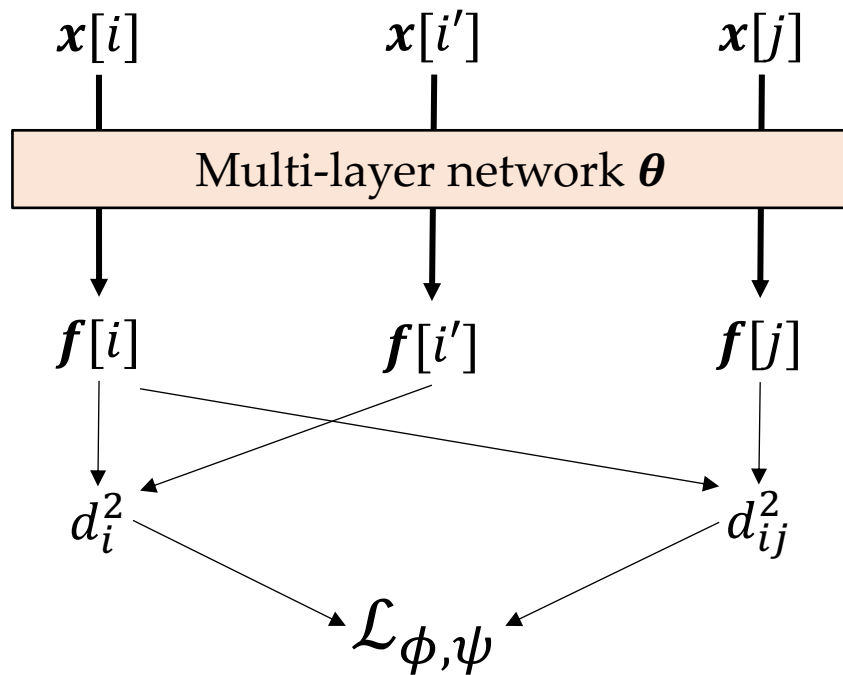
$$\text{Inter-view distance } d_{ij}^2 = \|f[i] - f[j]\|_2^2/2$$

Existing Understanding of contrastive learning

- Maximize the mutual information $MI(f, f')$
 - f and f' are two views of the same instance.
- Optimize Lower/Upper bound of mutual information $MI(f, f')$
 - InfoNCE loss (lower bound).
- Assume certain black-box function classes
 - No much understanding beyond the structure of loss function.

A family of contrastive losses

General Loss function we consider (ϕ, ψ are monotonous increasing functions)



$$\min_{\theta} \mathcal{L}_{\phi, \psi}(\theta) := \sum_{i=1}^N \phi \left(\sum_{j \neq i} \psi(d_i^2 - d_{ij}^2) \right)$$

Intra-view distance $d_i^2 = \|f[i] - f[i']\|_2^2 / 2$

Inter-view distance $d_{ij}^2 = \|f[i] - f[j]\|_2^2 / 2$

Example: InfoNCE

$$\mathcal{L}_{nce} := -\tau \sum_{i=1}^N \log \frac{\exp(-d_i^2 / \tau)}{\epsilon \exp(-d_i^2 / \tau) + \sum_{j \neq i} \exp(-d_{ij}^2 / \tau)}$$

$$= \tau \sum_{i=1}^N \log \left(\epsilon + \sum_{j \neq i} \exp \left(\frac{d_i^2 - d_{ij}^2}{\tau} \right) \right)$$

$$\phi(x) = \tau \log(\epsilon + x)$$

$$\psi(x) = \exp(x / \tau)$$

A general family

Contrastive Loss	$\phi(x)$	$\psi(x)$
InfoNCE (Oord et al., 2018)	$\tau \log(\epsilon + x)$	$e^{x/\tau}$
MINE (Belghazi et al., 2018)	$\log(x)$	e^x
Triplet (Schroff et al., 2015)	x	$[x + \epsilon]_+$
Soft Triplet (Tian et al., 2020c)	$\tau \log(1 + x)$	$e^{x/\tau + \epsilon}$
N+1 Tuplet (Sohn, 2016)	$\log(1 + x)$	e^x
Lifted Structured (Oh Song et al., 2016)	$[\log(x)]_+^2$	$e^{x+\epsilon}$
(Coria et al., 2020)	x	$\text{sigmoid}(cx)$
(Ji et al., 2021)	linear	linear

Coordinate-wise Optimization

Claim: Minimizing $\mathcal{L}_{\phi,\psi} \Leftrightarrow$ Coordinate-wise optimization:

$$\alpha_t := \arg \min_{\alpha \in \mathcal{A}} \mathcal{E}_{\alpha}(\theta_t) - \mathcal{R}(\alpha)$$

$$\theta_{t+1} := \theta_t + \eta \nabla_{\theta} \mathcal{E}_{\alpha_t}(\theta_t)$$

Max-player θ

Learns the representation to maximize contrastiveness.

Min-player α

Find distinct sample pairs that share similar representation (**hard negative pairs**)

The Energy Function $\mathcal{E}_\alpha(\boldsymbol{\theta})$

The energy \mathcal{E}_α is defined as the *trace* of **contrastive covariance** \mathbb{C}_α :

$$\mathcal{E}_\alpha(\boldsymbol{\theta}) := \text{tr } \mathbb{C}_\alpha[\mathbf{f}_\theta(\mathbf{x}), \mathbf{f}_\theta(\mathbf{x})]$$

The contrastive covariance $\mathbb{C}_\alpha[\mathbf{x}, \mathbf{y}] := \Sigma_0 - \Sigma_{\text{aug}}$

Inter-sample $\Sigma_0 := \sum_{i,j} \alpha_{ij} (\mathbf{x}[i] - \mathbf{x}[j])(\mathbf{y}[i] - \mathbf{y}[j])^T$

Intra-sample $\Sigma_{\text{aug}} := \sum_i \left(\sum_{j \neq i} \alpha_{ij} \right) (\mathbf{x}[i] - \mathbf{x}[i'])(\mathbf{y}[i] - \mathbf{y}[i'])^T$

Intuition of contrastive covariance $\mathbb{C}_\alpha[\mathbf{f}, \mathbf{f}]$

The contrastive covariance term $\mathbb{C}_\alpha[\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x})]$ has intuitions:

When α is uniform and batch size $N \rightarrow +\infty$, then

$$\mathbb{C}_\alpha[\mathbf{f}(\mathbf{x})] \rightarrow \mathbb{V}_{\mathbf{x}_0 \sim p(\cdot)} \mathbb{E}_{\mathbf{x} \sim p_{\text{aug}}(\cdot | \mathbf{x}_0)}[\mathbf{f}(\mathbf{x}) | \mathbf{x}_0]$$

Therefore, it becomes variance. Define $\mathbb{C}_\alpha[\mathbf{f}] := \mathbb{C}_\alpha[\mathbf{f}, \mathbf{f}]$

How min player α is determined?

If $\psi(x) = e^{x/\tau}$, then we have

$$\alpha(\boldsymbol{\theta}) := \arg \min_{\alpha \in \mathcal{A}} \mathcal{E}_\alpha(\boldsymbol{\theta}) - \mathcal{R}(\alpha)$$

where the feasible set $\mathcal{A} := \left\{ \alpha: \forall i, \sum_{j \neq i} \alpha_{ij} = \tau^{-1} \xi_i \phi'(\xi_i), \alpha_{ij} \geq 0 \right\}$

and entropy regularization term $\mathcal{R}(\alpha) := 2\tau \sum_{i=1}^N H(\alpha_i.)$

$$\xi_i := \sum_{j \neq i} \psi(d_i^2 - d_{ij}^2)$$

Different Losses, Same Energy Function

Contrastive Loss	$\phi(x)$	$\psi(x)$
InfoNCE (Oord et al., 2018)	$\tau \log(\epsilon + x)$	$e^{x/\tau}$
MINE (Belghazi et al., 2018)	$\log(x)$	e^x
Triplet (Schroff et al., 2015)	x	$[x + \epsilon]_+$
Soft Triplet (Tian et al., 2020c)	$\tau \log(1 + x)$	$e^{x/\tau + \epsilon}$
N+1 Tuplet (Sohn, 2016)	$\log(1 + x)$	e^x
Lifted Structured (Oh Song et al., 2016)	$[\log(x)]_+^2$	$e^{x + \epsilon}$
(Coria et al., 2020)	x	sigmoid(cx)
(Ji et al., 2021)	linear	linear

Different loss functions (ϕ, ψ) corresponds to the **same energy function \mathcal{E}**
How the min player α operates are different.

The intuition of pairwise importance α

For infoNCE with $\epsilon = 0$, solving the optimization problem yields:

$$\alpha_{ij}(\boldsymbol{\theta}) = \frac{\exp(-d_{ij}^2/\tau)}{\sum_{j \neq i} \exp(-d_{ij}^2/\tau)}$$

We put more weights on **small d_{ij}** , i.e., distinct samples with similar representations

Proposed: Pair-weighted CL (α -CL)

Optimize network parameter θ using gradient ascent of the energy function \mathcal{E}

$$\theta_{t+1} = \theta_t + \eta \nabla_{\theta} \mathcal{E}_{\text{sg}(\alpha_t)}(\theta_t)$$

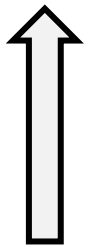
Pairwise importance $\alpha_t = \alpha(\theta_t)$

α can be either optimized by a separate loss function, or **directly** specified

Roadmap of α -CL

$$\mathcal{E}_\alpha(\boldsymbol{\theta}) := \text{tr } \mathbb{C}_\alpha[\mathbf{f}_\boldsymbol{\theta}(\mathbf{x})]$$

α -CL



$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\phi, \psi}(\boldsymbol{\theta})$$

Minimization of various CL losses



Understanding



Applications

- ✓ Dynamics of $\boldsymbol{\theta}$ with fixed α in the linear setting
- ✓? Dynamics of $\boldsymbol{\theta}$ with fixed α in the nonlinear setting
- ?? Dynamics of $\boldsymbol{\theta}$ with changing α
- ?? Hierarchical representation learning

- Finding the best $\alpha = \alpha(\boldsymbol{\theta})$ for performance gain
- Receptive field specific α
- More applications (e.g., CL in GNN)

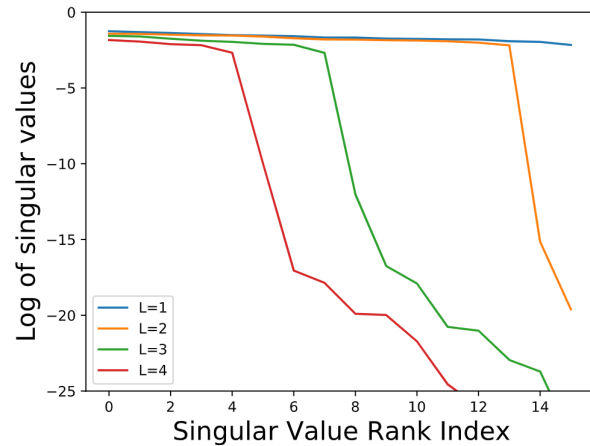
Initial Experimental Results

	CIFAR-10			STL-10		
	100 epochs	300 epochs	500 epochs	100 epochs	300 epochs	500 epochs
$\mathcal{L}_{quadratic}$	63.59 ± 2.53	73.02 ± 0.80	73.58 ± 0.82	55.59 ± 4.00	64.97 ± 1.45	67.28 ± 1.21
\mathcal{L}_{nce}	84.06 ± 0.30	87.63 ± 0.13	87.86 ± 0.12	78.46 ± 0.24	82.49 ± 0.26	83.70 ± 0.12
backprop $\alpha(\theta)$	83.42 ± 0.25	87.18 ± 0.19	87.48 ± 0.21	77.88 ± 0.17	81.86 ± 0.30	83.19 ± 0.16
α -CL- r_H	84.27 ± 0.24	87.75 ± 0.25	87.92 ± 0.24	78.53 ± 0.35	82.62 ± 0.15	83.74 ± 0.18
α -CL- r_γ	83.72 ± 0.19	87.51 ± 0.11	87.69 ± 0.09	78.22 ± 0.28	82.19 ± 0.52	83.47 ± 0.34
α -CL- r_s	84.72 ± 0.10	86.62 ± 0.17	86.74 ± 0.15	76.95 ± 1.06	80.64 ± 0.77	81.65 ± 0.59
α -CL-direct	85.09 ± 0.13	88.00 ± 0.12	88.16 ± 0.12	79.38 ± 0.16	82.99 ± 0.15	84.06 ± 0.24

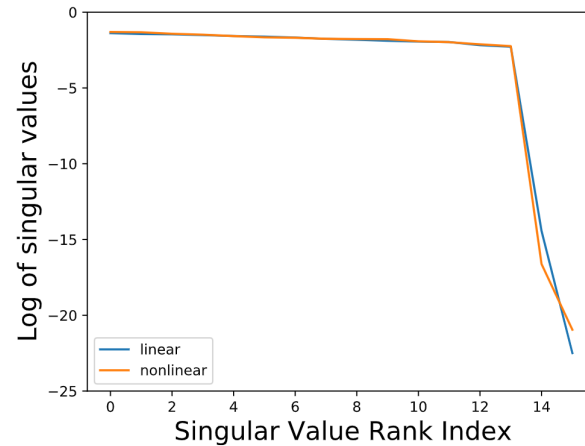
- (α -CL- r_H) Entropy regularizer $r_H(\alpha_{ij}) = -2\tau\alpha_{ij} \log \alpha_{ij}$;
- (α -CL- r_γ) Inverse regularizers $r_\gamma(\alpha_{ij}) = \frac{2\tau}{1-\gamma}\alpha_{ij}^{1-\gamma}$ ($\gamma > 1$).
- (α -CL- r_s) Square regularizer $r_s(\alpha_{ij}) = -\frac{\tau}{2}\alpha_{ij}^2$.
- (α -CL-direct) Directly setting α : $\alpha_{ij} = \exp(-d_{ij}^p/\tau)$ ($p > 1$).

Analysis of the Dynamics in CL with fixed α

Shouldn't contrastive SSL make full use of all dimensions? The answer is **No...**



(a) multiple layers



(b) nonlinear

Dimensional Collapsing

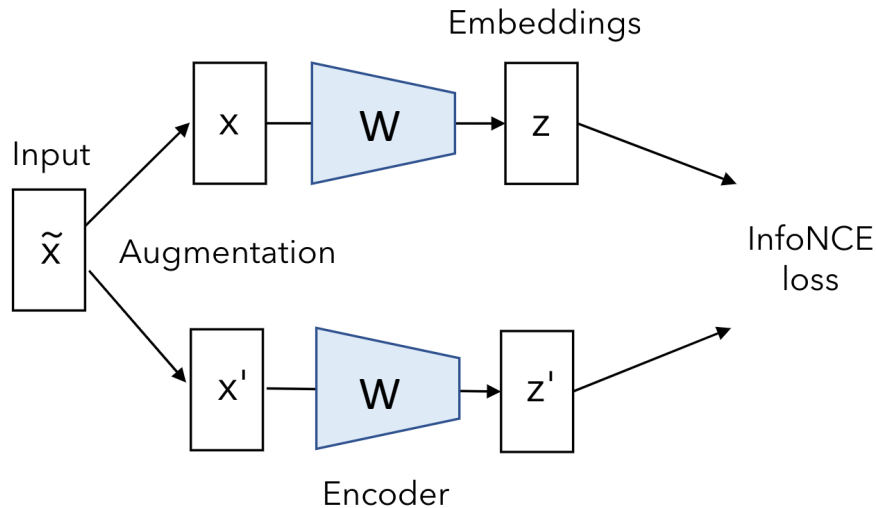
Two puzzling questions:

1. Why contrastive SSL still has collapsing issues?
2. Why $L = 1$ doesn't have collapsing, but $L \geq 2$ has the issue?



One-layer dynamics

Linear Model



InfoNCE loss

$$L := - \sum_{i=1}^N \log \frac{\exp(-d_i^2)}{\epsilon \exp(-d_i^2) + \sum_{j \neq i} \exp(-d_{ij}^2)}$$

The dynamics can be written down as follows:

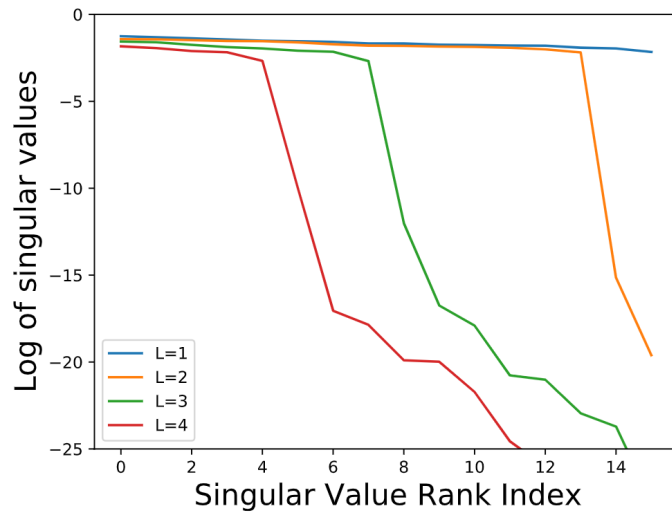
$$\frac{dW}{dt} = W \mathbb{C}_\alpha[x]$$

If $\mathbb{C}_\alpha[x]$ has negative eigenvalues,
then W will be low-rank

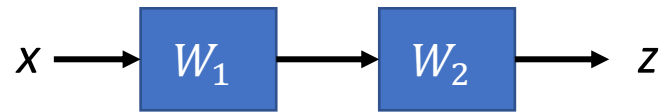
(will not happen in large batchsize + uniform α)

2-layer linear model still yields Dimensional Collapsing

- What if $\mathbb{C}_\alpha[x]$ is PSD?
- Still dimensional collapsing for deep models.



(a) multiple layers

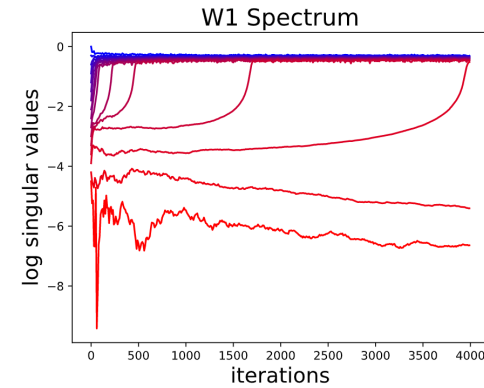


1. W_1 and W_2 will align with each other.
2. The dynamics of their singular values satisfy

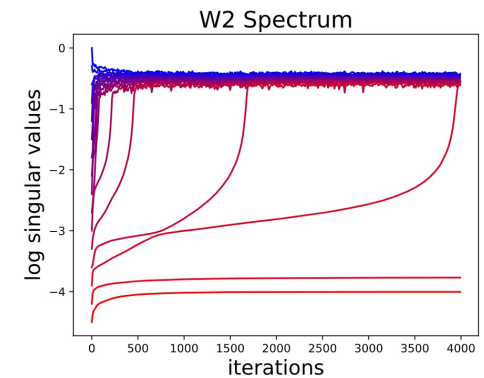
$$\dot{\sigma}_1^k = \sigma_1^k (\sigma_2^k)^2 (\mathbf{v}_1^k T X \mathbf{v}_1^k),$$

$$\dot{\sigma}_2^k = \sigma_2^k (\sigma_1^k)^2 (\mathbf{v}_1^k T X \mathbf{v}_1^k)$$

σ_1^k and σ_2^k grow much faster for k if $(\mathbf{v}_1^k)^T X \mathbf{v}_1^k$ is large.



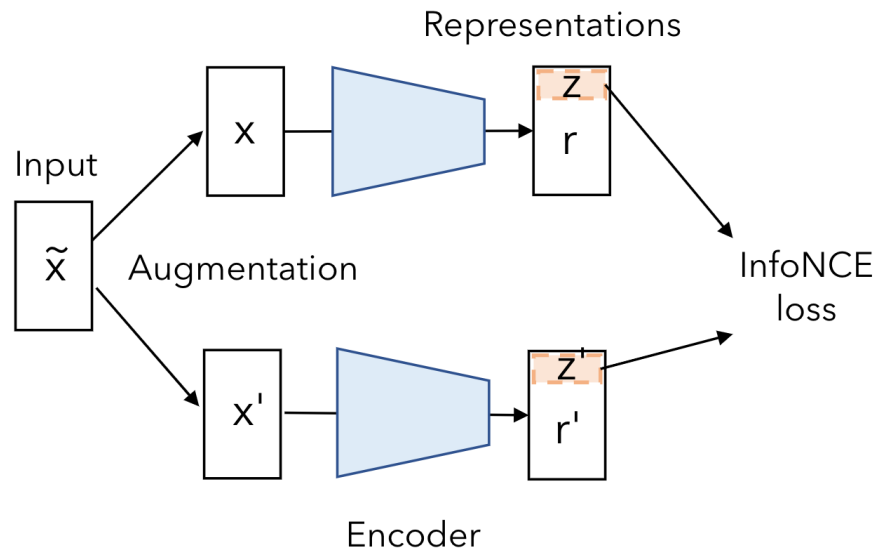
(a) W_1



(b) W_2

DirectCLR

- If things are aligned, why not let them align directly?



Loss function	Projector	Top-1 Accuracy
SimCLR	2-layer nonlinear projector	66.5
SimCLR	1-layer linear projector	61.1
SimCLR	no projector	51.5
DirectCLR	no projector	62.7



Deep linear case with fixed α

If $f_{\theta}(\mathbf{x}) = W(\boldsymbol{\theta})\mathbf{x}$, then it reduces to PCA objective

Corollary 2 (Representation learning in Deep Linear CL reparameterizes Principal Component Analysis (PCA)). *When $\mathbf{z} = W(\boldsymbol{\theta})\mathbf{x}$ with a constraint $WW^{\top} = I$, \mathcal{E}_{α} is the objective of Principal Component Analysis (PCA) with reparameterization $W = W(\boldsymbol{\theta})$:*

$$\max_{\boldsymbol{\theta}} \mathcal{E}_{\alpha}(\boldsymbol{\theta}) = \text{tr}(W(\boldsymbol{\theta})X_{\alpha}W^{\top}(\boldsymbol{\theta})) \quad \text{s.t. } WW^{\top} = I \quad (9)$$

here $X_{\alpha} := \mathbb{C}_{\alpha}[\mathbf{x}]$ is the contrastive covariance of input \mathbf{x} .

Deep linear case with fixed α

If $f_{\theta}(\mathbf{x}) = W_L W_{L-1} \dots W_1 \mathbf{x}$, then almost all local optima are global and it is PCA

Theorem 3 (Representation Learning with DeepLin is PCA). *If $\lambda_{\max}(X_{\alpha}) > 0$, then for any local maximum $\theta \in \Theta$ of Eqn. 11 whose $W_{>1}^{\top} W_{>1}$ has distinct maximal eigenvalue:*

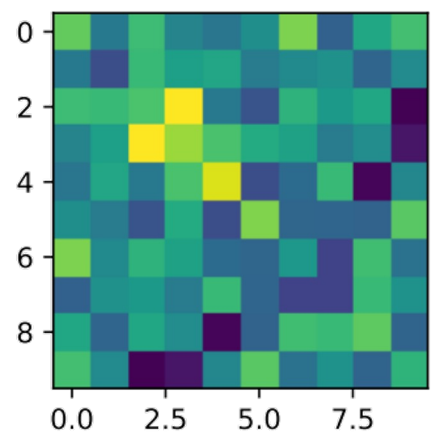
- *there exists a set of unit vectors $\{\mathbf{v}_l\}_{l=0}^L$ so that $W_l = \mathbf{v}_l \mathbf{v}_{l-1}^{\top}$ for $1 \leq l \leq L$, in particular, \mathbf{v}_0 is the unit eigenvector corresponding to $\lambda_{\max}(X_{\alpha})$, All W_l has rank-1 structure*
- *θ is global optimal with objective $\mathcal{E}^* = \lambda_{\max}(X_{\alpha})$.*

Corollary 3. *If we additionally use per-filter normalization (i.e., $\|\mathbf{w}_{lk}\|_2 = 1/\sqrt{n_l}$), then Thm. 3 holds and \mathbf{v}_l is more constrained: $[\mathbf{v}_l]_k = \pm 1/\sqrt{n_l}$ for $1 \leq l \leq L - 1$.*

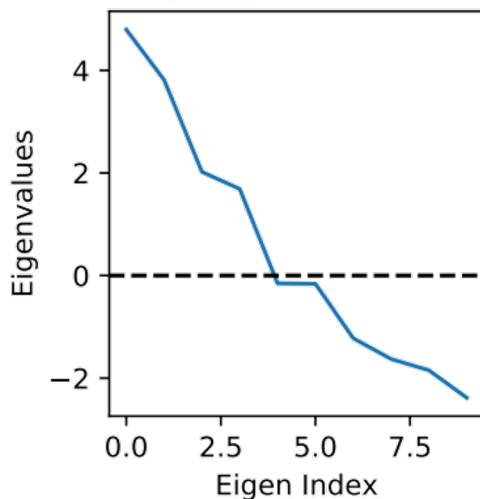
Summary

- If we fixed α and $\mathbf{f}_\theta(\mathbf{x})$ is a **linear** mapping, then
 - The max player $\max_{\theta} \mathcal{E}_\alpha(\theta) = \max_{\theta} \mathbb{C}_\alpha[\mathbf{f}_\theta(\mathbf{x})]$ becomes PCA
 - If $\mathbf{f}_\theta(\mathbf{x}) = W_L W_{L-1} \dots W_1 \mathbf{x}$, then all W_l becomes rank-1

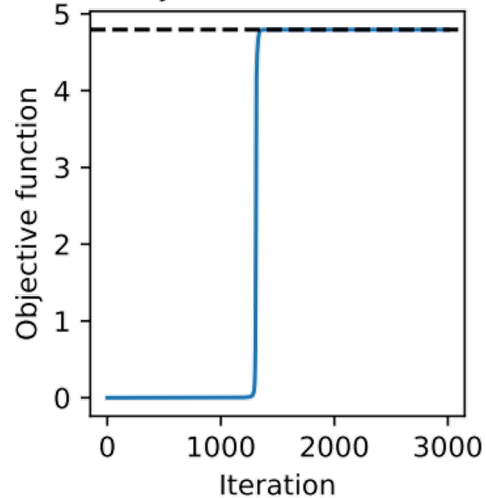
Contrastive Covariance X



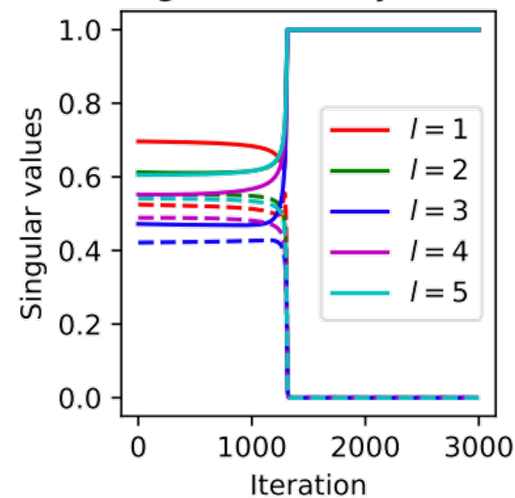
Eigenspectrum of X



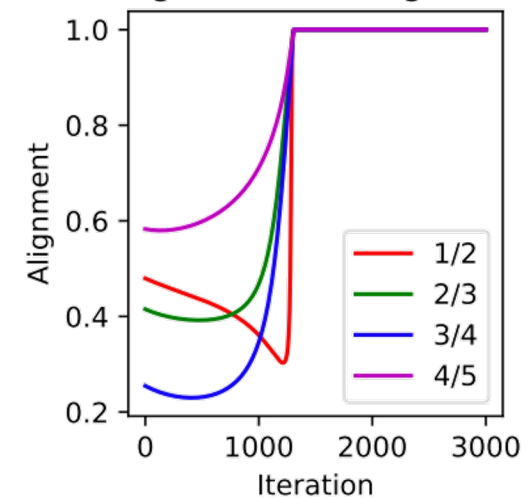
Objective over time



Singular value dynamics



Singular vector alignment



Nonlinear Setting

CL with linear model connects with classic approaches.

Where does the magic of deep models come from?

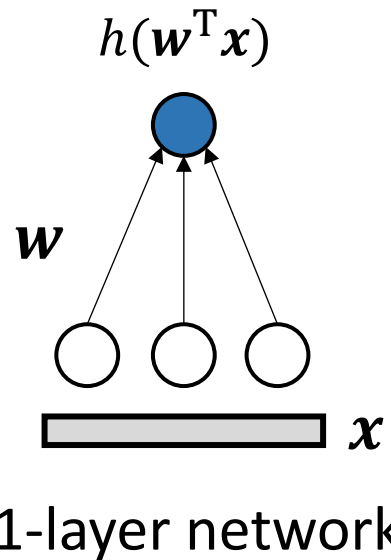
Nonlinearity!

Overview of Nonlinear Analysis

- One and Two-layer nonlinear networks
- Homogenous activations: $h(x) = h'(x)x$
 - Linear, ReLU, leaky ReLU and monomial activations $h(x) = x^p$ (with additional constant)
- Training Dynamics / Critical Point Analysis
 - Statistics of local optima.
 - Dynamics of weights during training

Nonlinear Setting

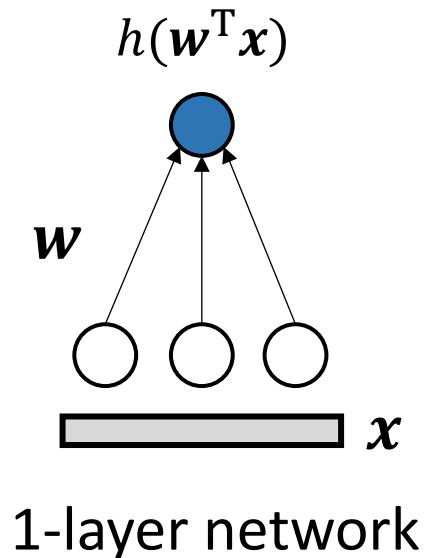
One-layer nonlinear network: $f_{\theta}(\mathbf{x}) = h(\mathbf{w}^T \mathbf{x})$



$$\max_{\|\mathbf{w}\|_2=1} \mathbb{C}_{\alpha}[f_{\theta}] = \mathbb{C}_{\alpha}[h(\mathbf{w}^T \mathbf{x})]$$

Nonlinear Setting

One-layer nonlinear network: $f_{\theta}(\mathbf{x}) = h(\mathbf{w}^T \mathbf{x})$



$$\max_{\|\mathbf{w}\|_2=1} \mathbb{C}_{\alpha}[f_{\theta}] = \mathbb{C}_{\alpha}[h(\mathbf{w}^T \mathbf{x})]$$

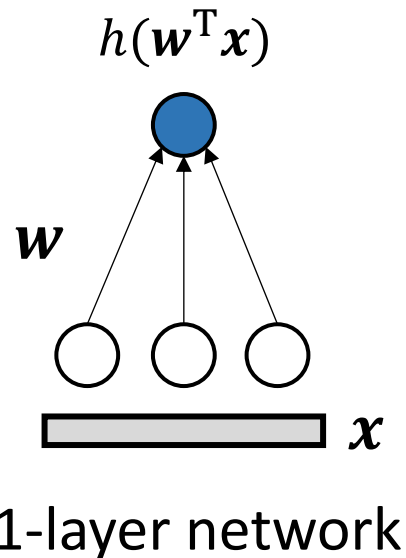
$$\text{Homogeneity: } \mathbb{C}_{\alpha}[h(\mathbf{w}^T \mathbf{x})] = \mathbf{w}^T \underbrace{\mathbb{C}_{\alpha}[\tilde{\mathbf{x}}^w]}_{\downarrow} \mathbf{w}$$

$\tilde{\mathbf{x}}^w := \mathbf{x} \cdot h'(\mathbf{w}^T \mathbf{x})$ is the **gated** data point

Similar to covariance matrix in PCA,
but now the matrix is not constant.

Nonlinear Setting

One-layer nonlinear network: $f_{\theta}(\mathbf{x}) = h(\mathbf{w}^T \mathbf{x})$



$$\max_{\|\mathbf{w}\|_2=1} \mathbb{C}_{\alpha}[f_{\theta}] = \mathbb{C}_{\alpha}[h(\mathbf{w}^T \mathbf{x})]$$

Homogeneity: $\mathbb{C}_{\alpha}[h(\mathbf{w}^T \mathbf{x})] = \mathbf{w}^T A(\mathbf{w}) \mathbf{w}$

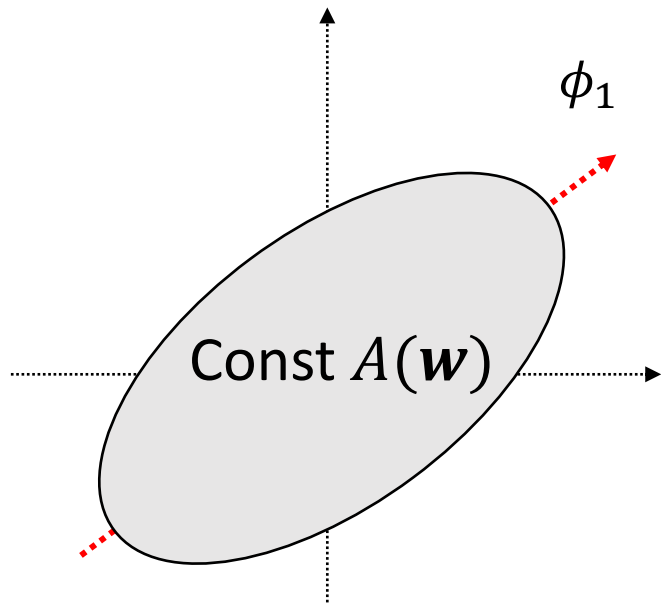
$\tilde{\mathbf{x}}^{\mathbf{w}} := \mathbf{x} \cdot h'(\mathbf{w}^T \mathbf{x})$ is the **gated** data point

$$\max_{\|\mathbf{w}\|_2=1} \mathbf{w}^T A(\mathbf{w}) \mathbf{w}$$

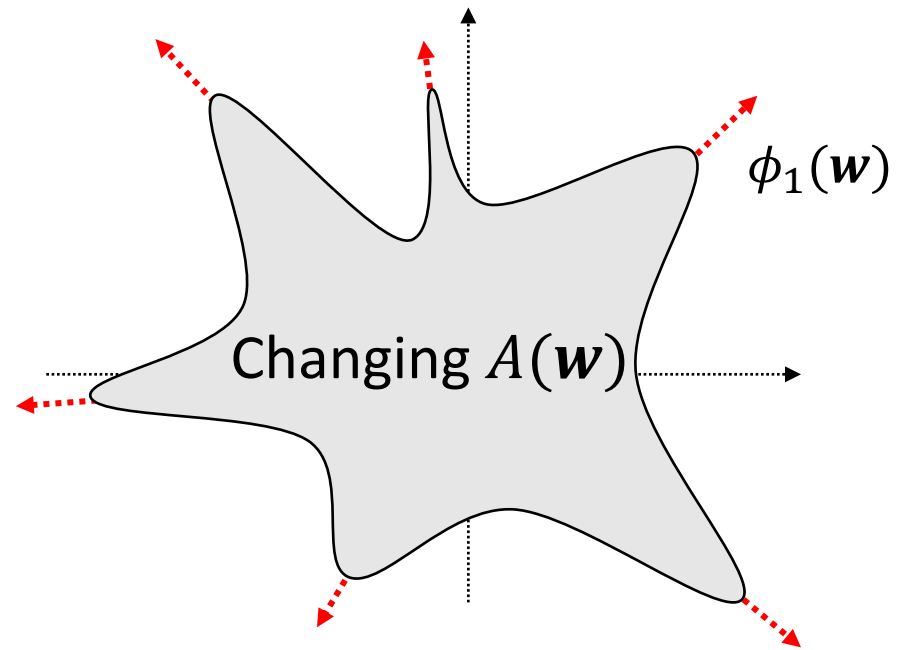
1-layer 1-node nonlinear network

$$\max_{\|\mathbf{w}\|_2=1} \mathbf{w}^T A(\mathbf{w}) \mathbf{w}$$

Linear



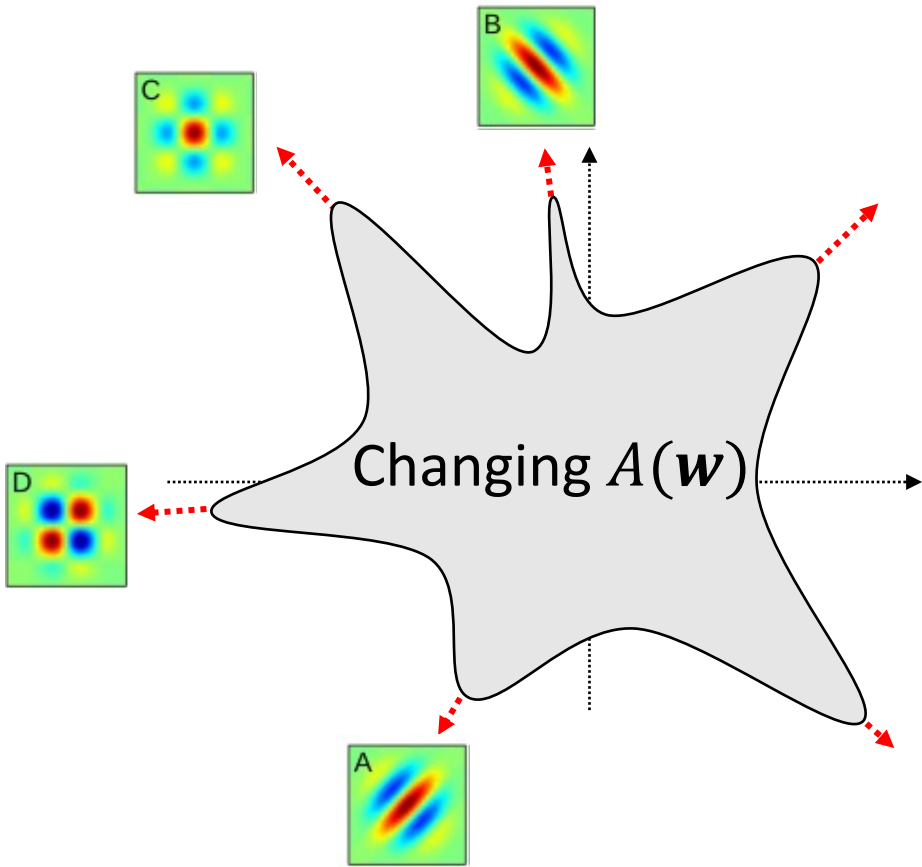
Non-linear



$\phi_1(\mathbf{w})$: Largest eigenvector of $A(\mathbf{w}) = \mathbb{C}_\alpha[\tilde{\mathbf{x}}^{\mathbf{w}}]$

Multiple largest eigenvectors!

1-layer 1-node nonlinear network



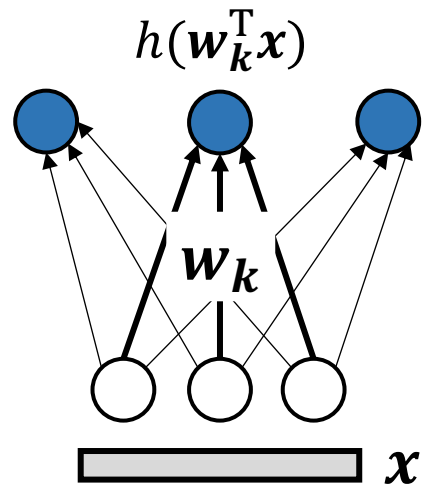
if $A(\mathbf{w}_*)\mathbf{w}_* = \lambda_*\mathbf{w}_*$, and $\lambda_{\text{gap}}(\mathbf{w}_*) > L$.

A pattern \mathbf{w}_* is dynamically stable (L is Lipschitz constant)

There exists multiple patterns in the data

1. Linear model cannot capture (only one PCA dimension)
2. With the nonlinearity, the model can capture them.

1-layer multiple node nonlinear network



1-layer network

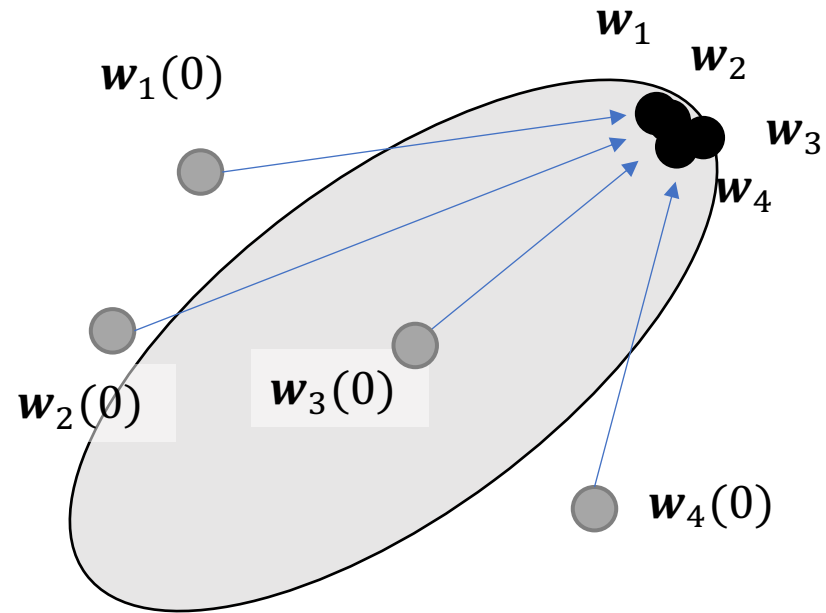
$$\text{Network } \mathbf{f}_\theta(\mathbf{x}) = h(W\mathbf{x})$$

$$W = \begin{bmatrix} \mathbf{w}_1^T \\ \dots \\ \mathbf{w}_k^T \\ \dots \\ \mathbf{w}_K^T \end{bmatrix} \quad k\text{-th filter}$$

$$\max_{\substack{\|\mathbf{w}_k\|_2=1 \\ 1 \leq k \leq K}} \text{tr } \mathbb{C}_\alpha[\mathbf{f}_\theta] = \sum_{k=1}^K \max_{\|\mathbf{w}_k\|_2=1} \mathbf{w}_k^T A(\mathbf{w}_k) \mathbf{w}_k$$

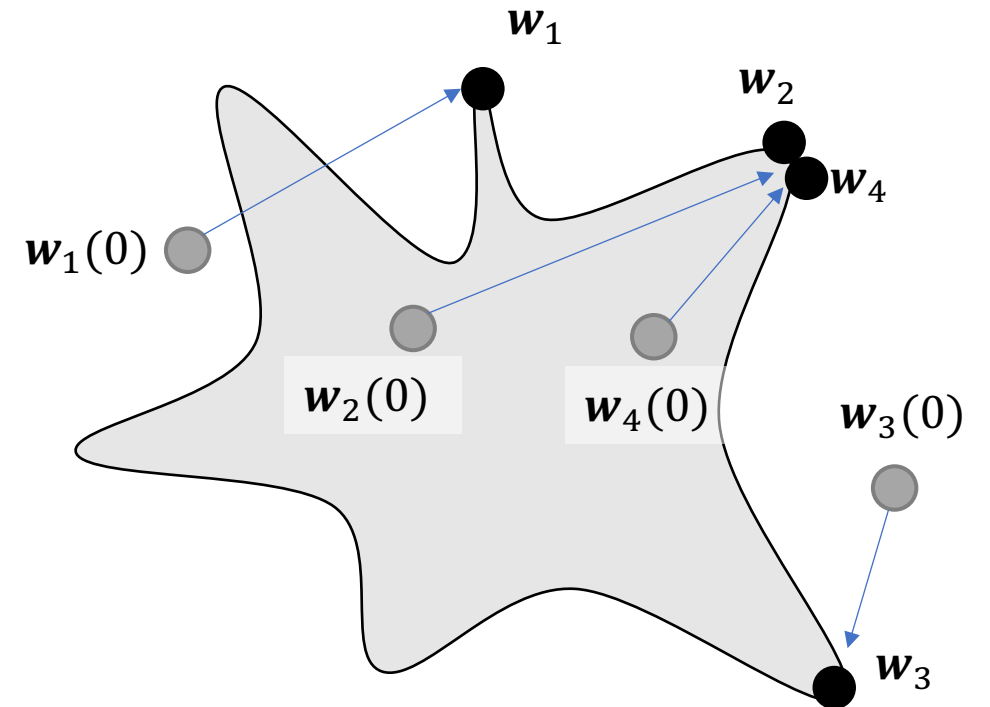
Independent one node objective

1-layer multiple node nonlinear network



Linear model

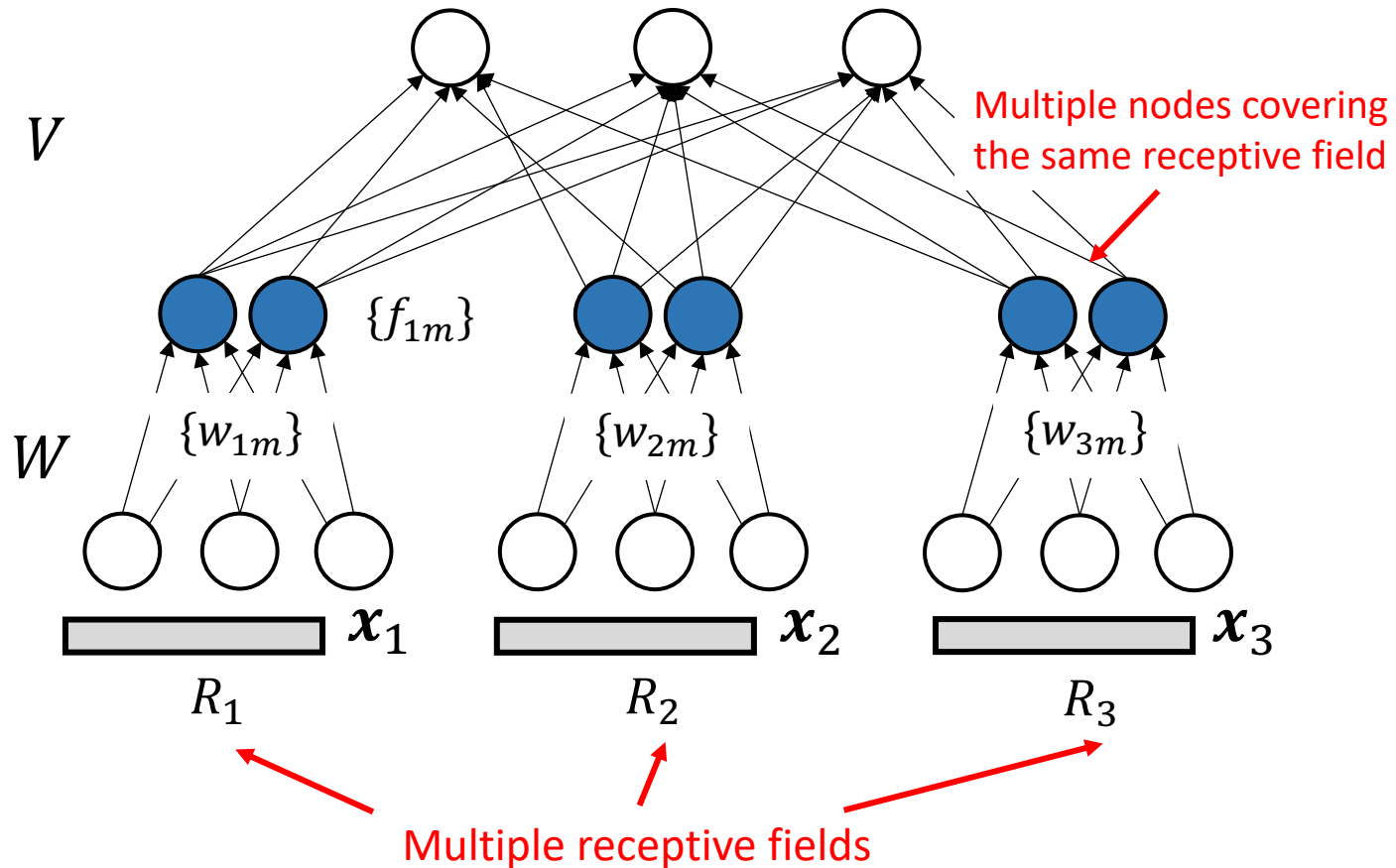
1. Every w_k converges to the global maximal eigenvector
2. More nodes do NOT help.



Nonlinear model

1. Each w_k can converge to different patterns
2. More nodes with diverse initialization learn more patterns!

2-layer nonlinear networks



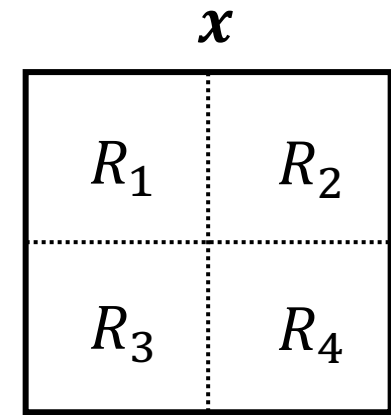
Notation

w_{km} \rightarrow m -th weight at
receptive field R_k
($1 \leq m \leq M$)
 \downarrow
Receptive
field R_k

Assumptions

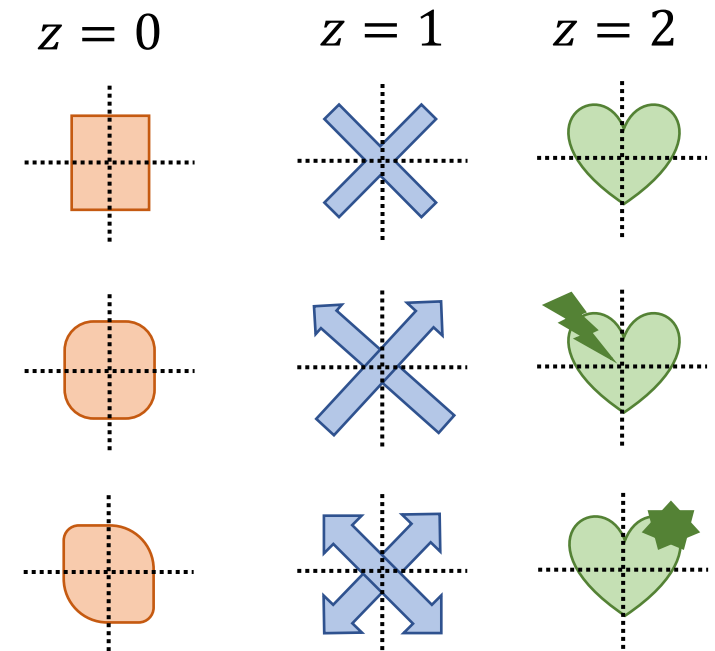
- 1. Uniform α , $N \rightarrow +\infty$, no augmentation $\rightarrow \mathbb{C}_\alpha[f] = \mathbb{V}[f]$
 - Technical condition to make the analysis more concise.
- 2. Fast training on the top layer $\rightarrow S := V^T V$ is rank-1
- 3. Conditional Independence $\rightarrow \mathbb{P}[\mathbf{x}|z] = \prod_{k=1}^K \mathbb{P}[\mathbf{x}_k|z]$

Conditional Independence



There **exists** categorical variable z (of cardinality C) so that

$$\mathbb{P}[\mathbf{x}|z] = \prod_{k=1}^K \mathbb{P}[\mathbf{x}_k | z]$$



What linear network cannot do

With linear activation, we have

$$\frac{d\mathbf{w}_{km}}{dt} = s_{km} \mathbf{b}_k(W, V)$$

For a receptive field R_k , the gradients of all its weights are ***co-linear***.

→ No diverse feature can be learned.

Nonlinear network will not have such constraints.

Global modulation

$C = 2, M = 1$ case (binary z , one filter at one receptive field)

$$\frac{d\mathbf{w}_k}{dt} = \left(s_k^2 L_k + \frac{1}{Z^2 (\lambda - d_k)} \Delta_k \Delta_k^T \right) \mathbf{w}_k$$

$L_k := \mathbb{E}_z \mathbb{V}[\tilde{\mathbf{x}}_k | z]$, lower-layer $A(\mathbf{w})$ matrix, $d_k = \mathbf{w}_k^T L_k \mathbf{w}_k$

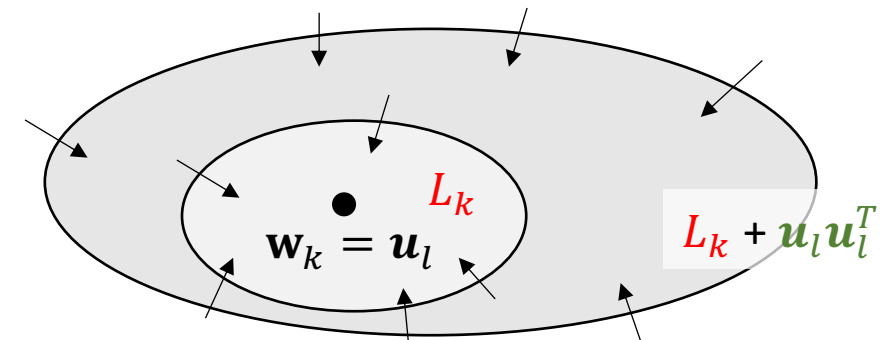
$\Delta_k := \mathbb{E}[\tilde{\mathbf{x}}_k | z = 1] - \mathbb{E}[\tilde{\mathbf{x}}_k | z = 0]$, *global modulation*

→ Features that help discriminates the variable z are encouraged.

“Feature Emergence”

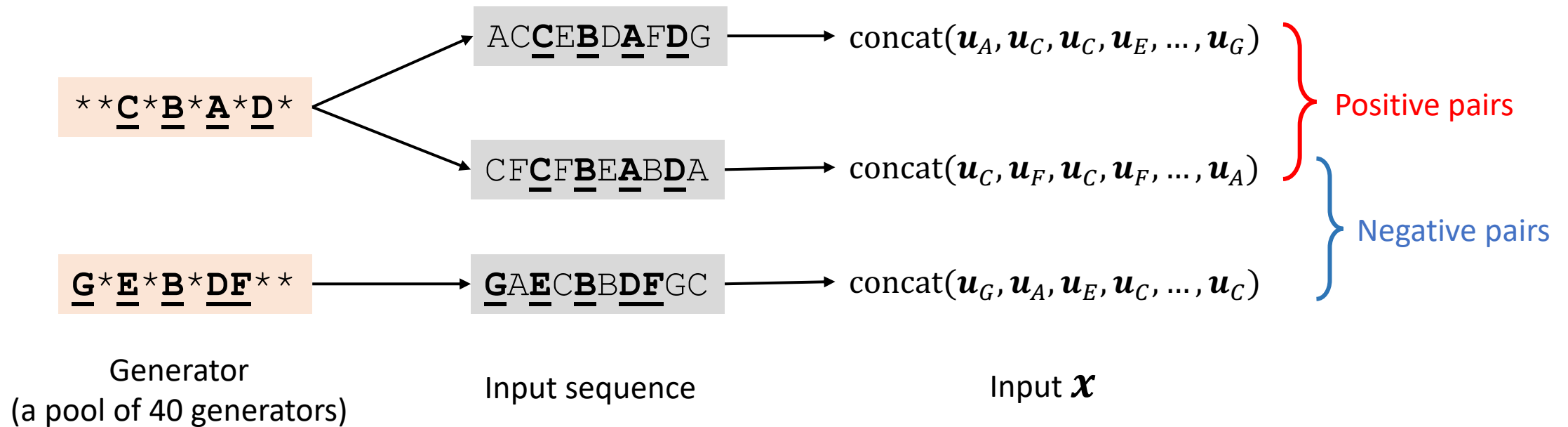
- During CL training, **no specification** of the hidden variable z .
- However, CL automatically prioritizes features that are sensitive to z .
 - $L_k + \Delta_k \Delta_k^T$, where $\Delta_k := \mathbb{E}[\tilde{\mathbf{x}}_k | z = 1] - \mathbb{E}[\tilde{\mathbf{x}}_k | z = 0]$

Theorem 4 (Global modulation of attractive basin). *If the structural assumption holds: $L_k(\mathbf{w}_k) = \sum_l g(\mathbf{u}_l^T \mathbf{w}_k) \mathbf{u}_l \mathbf{u}_l^T$ with $g(\cdot) > 0$ a linear increasing function and $\{\mathbf{u}_l\}$ orthonormal bases, then for $L_k + c \mathbf{u}_l \mathbf{u}_l^T$, its attractive basin of $\mathbf{w}_k = \mathbf{u}_l$ is larger than L_k 's for $c > 0$.*



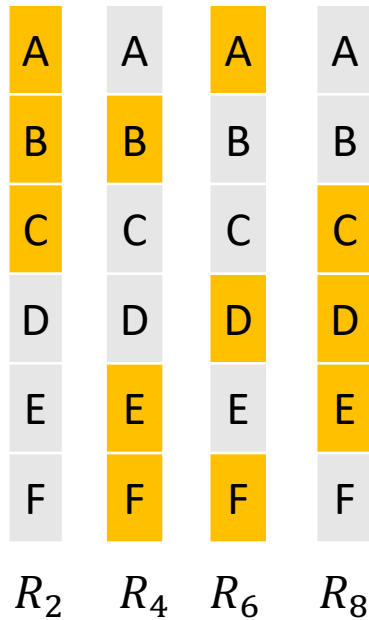
Experiment Setting

Synthetic Dataset




Embedding $\{\mathbf{u}_a\}$ are orthogonal to each other


Experiment Setting



$$R_2^g = \{A, B, C\}$$

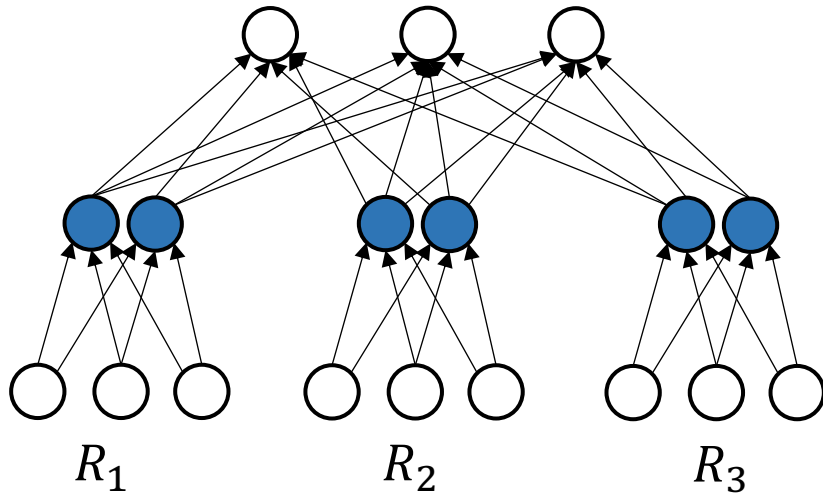
$$R_4^g = \{B, E, F\}$$

 R_k^g : **relevant** symbols to appear in generators, $|R_k^g| = P$

 **irrelevant** symbols that is only generated by wildcard *

Model Architecture & Evaluation Metric

Network Architecture



The over-parameterization factor $\beta := M/P$
(i.e., #filters / #patterns at each receptive field)

Evaluation Metric

Check whether the learned weights \mathbf{w}_{km} is matched with a token embedding \mathbf{u}_a

$$\chi_+(R_k) = \frac{1}{P} \sum_{a \in R_k^g} \max_m \frac{\mathbf{w}_{km}^T \mathbf{u}_a}{\|\mathbf{w}_{km}\|_2 \|\mathbf{u}_a\|_2}$$

$$\bar{\chi}_+ = \frac{1}{K} \sum_{k=1}^K \chi_+(R_k)$$

Experimental Results

β	$P = 1$		$P = 3$		$P = 5$		$P = 10$	
	Linear	ReLU	Linear	ReLU	Linear	ReLU	Linear	ReLU
1	0.95 ± 0.00	0.31 ± 0.23	0.79 ± 0.02	0.75 ± 0.11	0.67 ± 0.03	0.70 ± 0.06	0.60 ± 0.01	0.68 ± 0.05
2	0.95 ± 0.00	0.61 ± 0.13	0.81 ± 0.01	0.90 ± 0.09	0.70 ± 0.00	0.88 ± 0.04	0.63 ± 0.01	0.93 ± 0.02
5	0.96 ± 0.00	0.93 ± 0.06	0.85 ± 0.01	0.99 ± 0.02	0.73 ± 0.00	1.00 ± 0.00	0.66 ± 0.01	0.99 ± 0.01
10	0.96 ± 0.00	1.00 ± 0.00	0.86 ± 0.01	1.00 ± 0.00	0.77 ± 0.00	1.00 ± 0.00	0.68 ± 0.01	1.00 ± 0.00

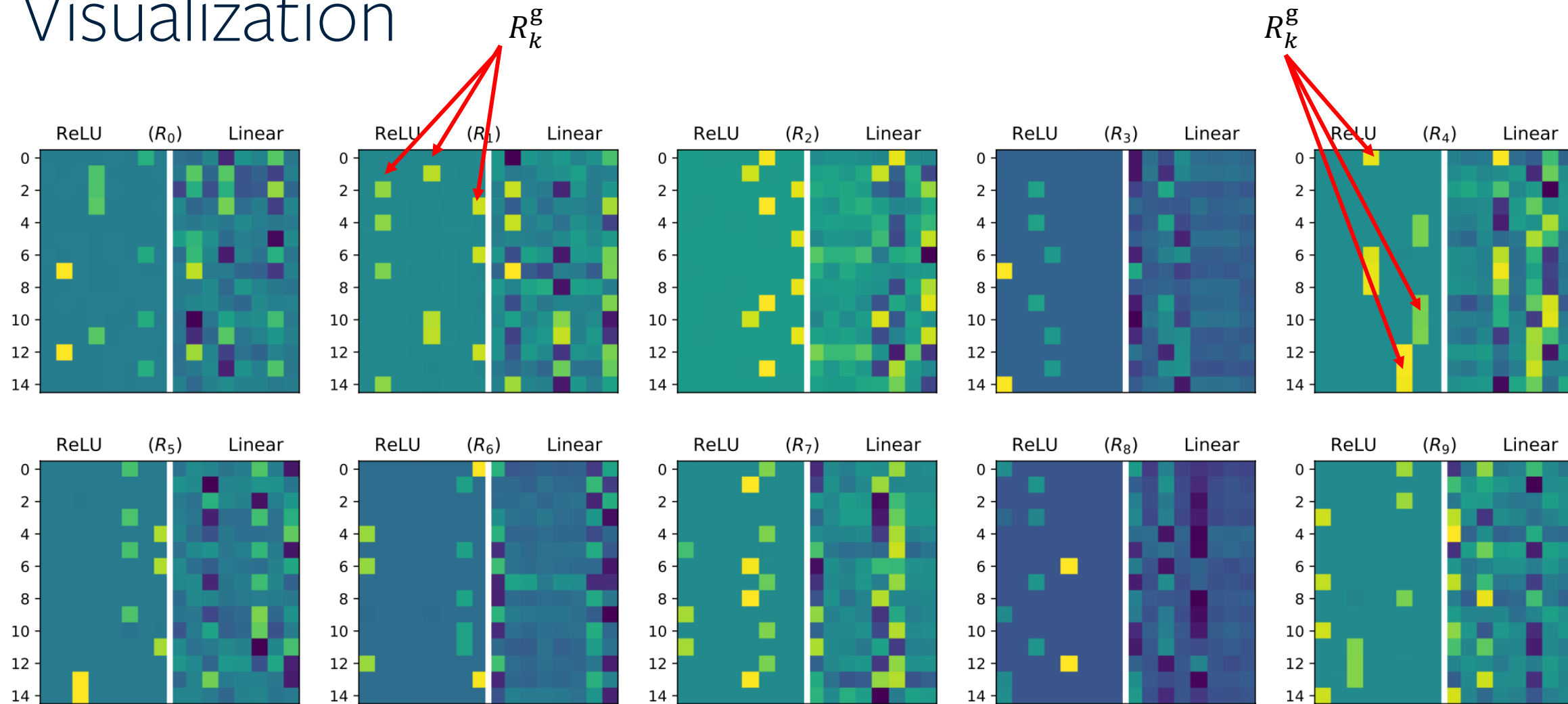
When there is no diverse patterns ($P = 1$),
Linear is better than ReLU

Also, over-parameterization (large β) doesn't help in linear case

ReLU gives stronger performance (than linear) when

1. There are diverse relevant patterns ($P > 1$)
2. There are strong over-parameterization ($\beta > 1$)

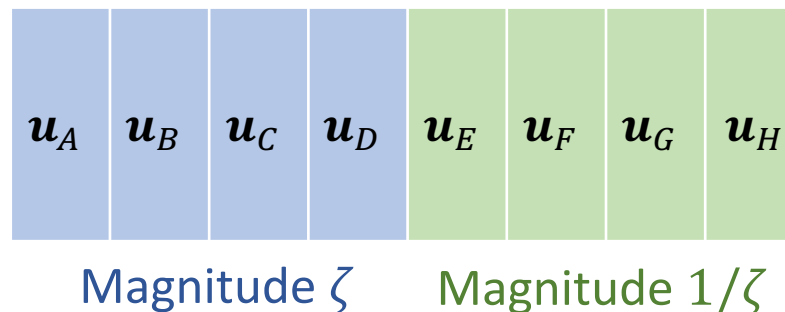
Visualization



Effect of BatchNorm

Non-uniformity ζ :

Token embedding matrix



$\zeta = 10$

β	$P = 1$		$P = 3$		$P = 5$		$P = 10$	
	NoBN	BN	NoBN	BN	NoBN	BN	NoBN	BN
1	0.35 ± 0.17	0.23 ± 0.06	0.51 ± 0.02	0.24 ± 0.11	0.54 ± 0.04	0.38 ± 0.04	0.54 ± 0.03	0.56 ± 0.04
2	0.39 ± 0.18	0.33 ± 0.12	0.59 ± 0.05	0.41 ± 0.07	0.58 ± 0.07	0.56 ± 0.02	0.56 ± 0.01	0.71 ± 0.05
5	0.53 ± 0.06	0.46 ± 0.06	0.66 ± 0.06	0.60 ± 0.08	0.65 ± 0.09	0.74 ± 0.02	0.60 ± 0.01	0.90 ± 0.03
10	0.56 ± 0.16	0.46 ± 0.15	0.78 ± 0.02	0.80 ± 0.03	0.70 ± 0.04	0.92 ± 0.01	0.63 ± 0.01	0.97 ± 0.02

BatchNorm works in the region of multiple patterns ($P > 1$) and over-parameterization ($\beta > 1$)

Quadratic Loss versus InfoNCE

InfoNCE loss

β	$P = 1$		$P = 3$		$P = 5$		$P = 10$	
	Linear	ReLU	Linear	ReLU	Linear	ReLU	Linear	ReLU
1	0.95 ± 0.00	0.31 ± 0.23	0.79 ± 0.02	0.75 ± 0.11	0.67 ± 0.03	0.70 ± 0.06	0.60 ± 0.01	0.68 ± 0.05
2	0.95 ± 0.00	0.61 ± 0.13	0.81 ± 0.01	0.90 ± 0.09	0.70 ± 0.00	0.88 ± 0.04	0.63 ± 0.01	0.93 ± 0.02
5	0.96 ± 0.00	0.93 ± 0.06	0.85 ± 0.01	0.99 ± 0.02	0.73 ± 0.00	1.00 ± 0.00	0.66 ± 0.01	0.99 ± 0.01
10	0.96 ± 0.00	1.00 ± 0.00	0.86 ± 0.01	1.00 ± 0.00	0.77 ± 0.00	1.00 ± 0.00	0.68 ± 0.01	1.00 ± 0.00

Quadratic loss

β	$P = 1$		$P = 3$		$P = 5$		$P = 10$	
	Linear	ReLU	Linear	ReLU	Linear	ReLU	Linear	ReLU
1	0.92 ± 0.07	-0.02 ± 0.19	0.60 ± 0.04	0.31 ± 0.22	0.45 ± 0.03	0.42 ± 0.07	0.38 ± 0.01	0.51 ± 0.06
2	0.96 ± 0.01	0.37 ± 0.29	0.64 ± 0.07	0.61 ± 0.05	0.48 ± 0.02	0.63 ± 0.11	0.43 ± 0.02	0.64 ± 0.02
5	0.95 ± 0.02	0.67 ± 0.29	0.68 ± 0.04	0.65 ± 0.12	0.52 ± 0.04	0.80 ± 0.05	0.48 ± 0.01	0.69 ± 0.04
10	0.96 ± 0.01	0.86 ± 0.06	0.68 ± 0.08	0.80 ± 0.16	0.52 ± 0.01	0.82 ± 0.04	0.50 ± 0.01	0.74 ± 0.02

Same trend, but worse performance

Future Works

- Dynamics of θ under changing pairwise importance α .
 - With changing α , it is possible that we might create **exponential** locally maximal eigenvectors in \mathbb{C}_α
 - Capture richer patterns.

- What if the top-layer also has ReLU activations?
 - Capture local hidden variable rather than a global one.
 - Advantage over traditional Graphical Model (GM)
 - GM operates on human-defined random variables.
 - NN **discovers** novel random variables.

$$\text{Global: } \mathbb{P}[\mathbf{x}|z] = \prod_{k=1}^K \mathbb{P}[\mathbf{x}_k|z]$$



$$\text{Local: } \mathbb{P}[\mathbf{x}|z] = \prod_{k=1}^K \mathbb{P}[\mathbf{x}_k|z] \text{ for } \mathbf{x} \in \Omega$$

- Towards understanding multi-layer networks / hierarchical representation
- Other architectures: e.g., Transformers.

Thanks!