Contrastive Learning Can Find An Optimal Basis For Approximately View-Invariant Functions

Daniel D. Johnson, Ayoub El Hanchi, Chris J. Maddison

UNIVERSITY OF TORONTO

Google Research

Contrastive learning models are **kernels**

Common contrastive learning losses can be decomposed into:

- a kernel-based loss function
- and a learned positive-definite kernel

The minima of these decomposed losses are *the same*!

(up to constant scaling)

NT-XEnt (Chen et al., 2020a; Van den Oord et al., 2018)	Loss	$\mathbb{E}\left[-\log \frac{\widehat{K}_\theta(a_1^+,a_2^+)}{\widehat{K}_\theta(a_1^+,a_2^+)+\sum_{a_i^-}\widehat{K}_\theta(a_1^+,a_i^-)}\right]$
	Kernel	$\widehat{K}_\theta(a_1,a_2) = \exp(h_\theta(a_1)^\top h_\theta(a_2)/\tau)$
	Minimum	$\widehat{K}_*(a_1,a_2) = \frac{p_+(a_1,a_2)}{p(a_1)p(a_2)} \cdot C_{[a_1]}$
NT-Logistic (Chen et al., 2020a; Tosh et al., 2021)	Loss	$\mathbb{E}\left[-\log\sigma(\log \widehat{K}_\theta(a_1^+,a_2^+))\right] + \mathbb{E}\left[-\log\sigma(-\log \widehat{K}_\theta(a_1^-,a_2^-))\right]$
	Kernel	$\widehat{K}_\theta(a_1,a_2) = \exp(h_\theta(a_1)^\top h_\theta(a_2)/\tau)$
	Minimum	$\widehat{K}_*(a_1,a_2) = \frac{p_+(a_1,a_2)}{p(a_1)p(a_2)}$
Spectral (HaoChen et al., 2021)	Loss	$\mathbb{E}\left[-2\widehat{K}_\theta(a_1^+,a_2^+)\right] + E\left[(\widehat{K}_\theta(a_1^-,a_2^-))^2\right]$
	Kernel	$\widehat{K}_\theta(a_1,a_2) = h_\theta(a_1)^\top h_\theta(a_2)$
	Minimum	$\widehat{K}_*(a_1,a_2) = \frac{p_+(a_1,a_2)}{p(a_1)p(a_2)}$

Contrastive learning models are **kernels**

This minimum is the **positive-pair kernel:**

$$K_+(a_1, a_2) = \frac{p_+(a_1, a_2)}{p(a_1)p(a_2)}$$

defined in terms of

- a distribution $p(z)$ of latent examples
- a distribution $p(a|z)$ of augmentations

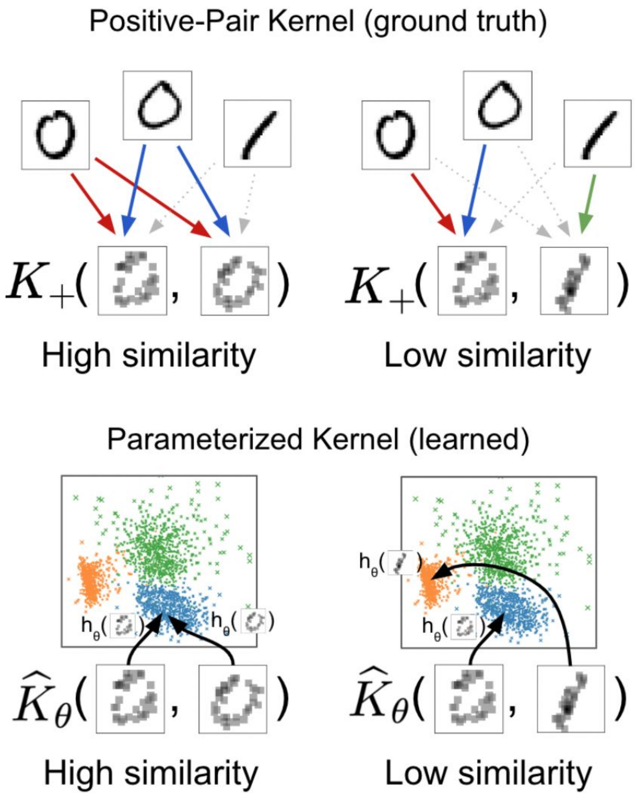
with $p_+(a_1, a_2) = \sum_z p(a_1|z)p(a_2|z)p(z)$

Contrastive learning models are **kernels**

The positive pair kernel assigns high similarity to likely positive pairs.

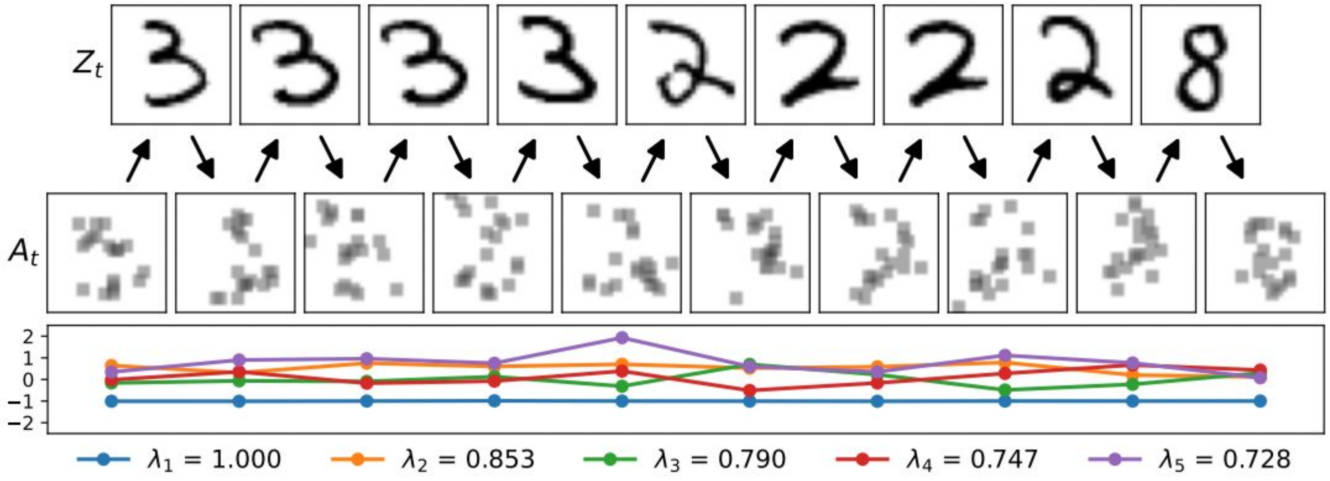
$$K_+(a_1, a_2) = \frac{p_+(a_1, a_2)}{p(a_1)p(a_2)}$$

Contrastive learning models can be seen as *parameterized approximations* of this kernel.



Kernel principal components are eigenfunctions

Performing Kernel PCA yields a sequence of projection functions, which are the *eigenfunctions of a Markov chain* over augmentations.



These eigenfunctions are an **optimal basis**

Assumption: The downstream task involves learning an *approximately view-invariant function* $g : \mathcal{A} \to \mathbb{R}$:

$$\mathbb{E}_{p_+(a_1,a_2)}\left[\left(g(a_1) - g(a_2)\right)^2\right] \leq \varepsilon$$

These eigenfunctions are an **optimal basis**

We prove that the eigenfunction representation *minimizes worst-case L_2 approximation error* of linear predictors under this assumption.

Theorem 4.1. *Let $\mathcal{F}_r = \{a \mapsto \beta^\top r(a) : \beta \in \mathbb{R}^d\}$ be the subspace of linear predictors from representation r, and S_ε be the set of functions satisfying Assumption 1.1. Let $r_*^d(a) = [f_1(a), f_2(a), \ldots, f_d(a)]$ be the representation consisting of the d eigenfunctions of the positive pair Markov chain with the largest eigenvalues. Then $\mathcal{F}_{r_*^d}$ maximizes the view invariance of the least-invariant unit-norm predictor in $\mathcal{F}_{r_*^d}$:*

$$\mathcal{F}_{r_*^d} = \operatorname*{argmin}_{\dim(\mathcal{F})=d} \max_{\hat{g}\in\mathcal{F},\ \mathbb{E}[\hat{g}(a)^2]=1} \mathbb{E}_{p_+}\left[\left(\hat{g}(a_1) - \hat{g}(a_2)\right)^2\right]. \tag{5}$$

Simultaneously, $\mathcal{F}_{r_*^d}$ minimizes the (quadratic) approximation error for the worst-case target function satisfying Assumption 1.1 for any fixed ε:

$$\mathcal{F}_{r_*^d} = \operatorname*{argmin}_{\dim(\mathcal{F})=d} \max_{g\in S_\varepsilon} \min_{\hat{g}\in\mathcal{F}} \mathbb{E}_{p(a)}\left[\left(g(a) - \hat{g}(a)\right)^2\right]. \tag{6}$$

These eigenfunctions are an **optimal basis**

In fact, decomposing a function as a sum of weighted eigenfunctions exactly determines its view-invariance!

Specifically, if $g(a) = \sum_i c_i f_i(a)$, then

$$\mathbb{E}_{p_+(a_1,a_2)}\left[\left(g(a_1) - g(a_2)\right)^2\right] = \sum_i (2 - 2\lambda_i)c_i^2.$$

How to build eigenfunction representations with contrastive learning

Strategy 1:

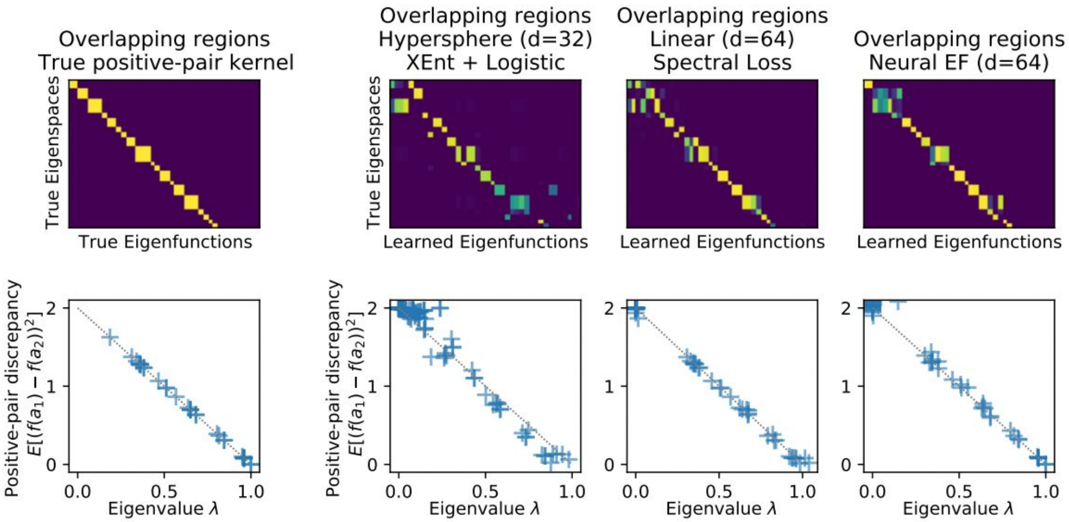
- Train a contrastive learning model using cross entropy, logistic, or spectral loss
- Use Kernel PCA to extract a representation from the learned kernel

Strategy 2:

- Directly estimate the principal eigenfunctions of the positive-pair kernel

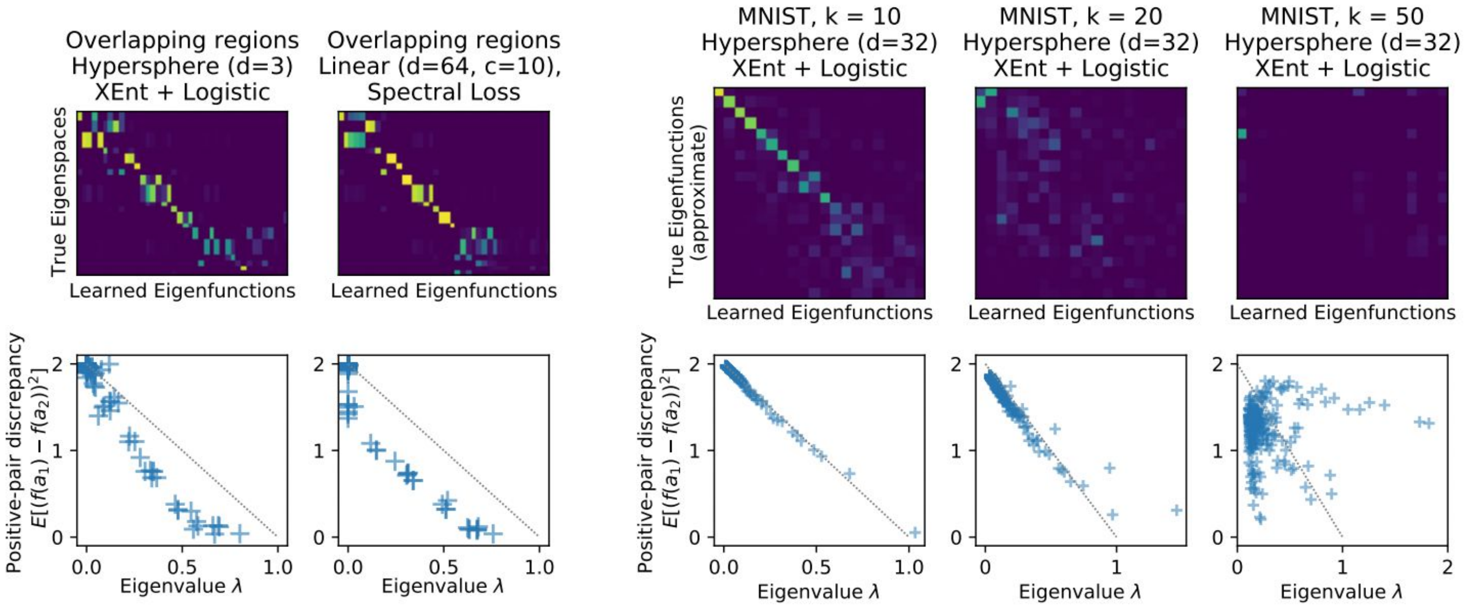
Do contrastive learning methods find eigenfunctions?

At high augmentation strengths, we can extract the same eigenfunction representation across multiple model parameterizations, with the same learned eigenvalues.



Do contrastive learning methods find eigenfunctions?

But *constrained kernel parameterizations* and *weak augmentations* both degrade approximation quality.



Conclusion

- Our work highlights the surprising connections between contrastive learning, view-invariance, Markov chains, and kernel methods

- Future directions:
  - Building new self-supervised learning methods using the positive-pair kernel
  - Using the kernel perspective to understand the effects of inductive biases