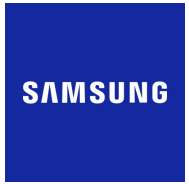
Exploring the gap between collapsed and whitened features in self-supervised learning

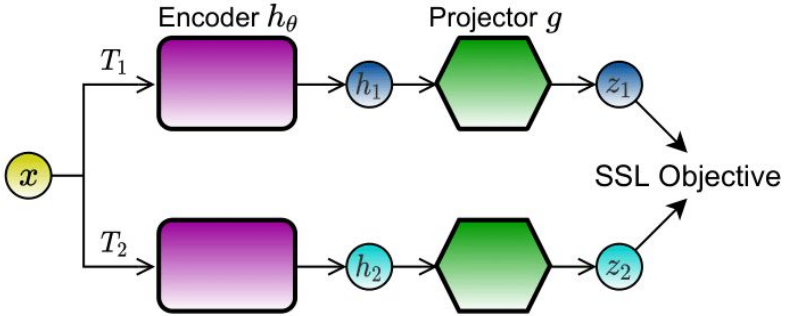
Bobby He and Mete Ozay

ICML 2022



Self-supervised learning (SSL)

- Obtaining labelled data can be expensive.
- **Self-supervised learning** (SSL) learns from unlabelled data $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^{N}$ producing useful encoder features, $h_{\boldsymbol{\theta}}(\boldsymbol{X}) \in \mathbb{R}^{N \times d}$ for downstream tasks.
- **Popular SSL strategy** (SimCLR, BYOL, SwAV, Barlow Twins + more): encourage feature invariance to some input transformations, T_1, T_2, ie. $h_{\boldsymbol{\theta}}(T_1(\boldsymbol{x})) \approx h_{\boldsymbol{\theta}}(T_2(\boldsymbol{x}))$



Feature collapse vs. whitening in SSL

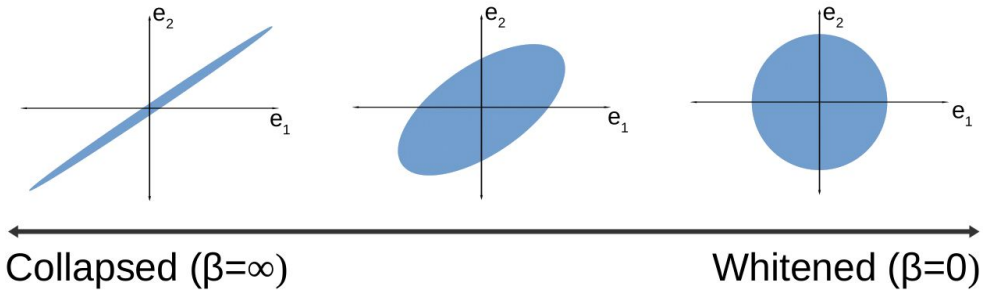
- **Common goal in SSL**: avoiding *collapsed* NN features $h_{\boldsymbol{\theta}}(\boldsymbol{x}') = \mathbf{c},\ \forall \boldsymbol{x}'$.
- Some approaches: contrastive learning (SimCLR) or clustering (SwAV).
- *Whitening* or *decorrelating* features has been proposed as a sufficient condition to avoid collapse.[1,2,3,4]
- **Barlow Twins**[2] adds a regulariser to the SSL loss function to encourage decorrelated features, whilst still achieving invariance to transformations:

Barlow Twins objective = Transformation Invariance loss + Feature Decorrelation reg.

[1] Ermolov et al, Whitening for Self-Supervised Representation Learning, 2020
[2] Zbontar et al, Barlow Twins: Self-Supervised Learning via Redundancy Reduction, 2021
[3] Hua et al, On Feature Decorrelation in Self-Supervised Learning, 2021
[4] Bardes et al, VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning, 2021

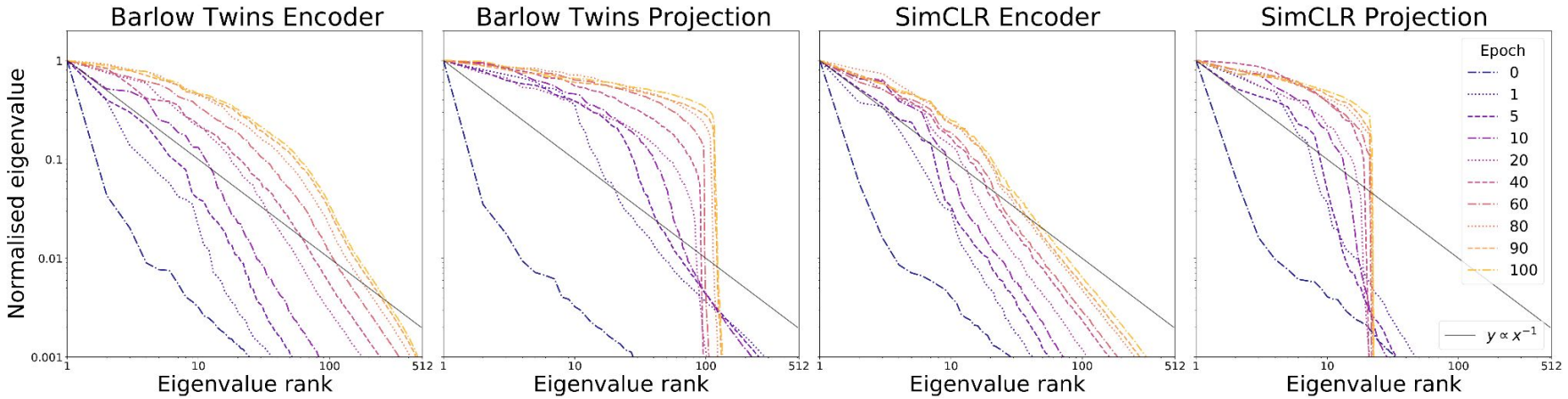
The spectrum of feature eigenspectra

- *Whitened* & *Collapsed* features represent two extremes of a gap:
  - Exactly whitened features have identity covariance: $h_{\boldsymbol{\theta}}(\boldsymbol{X})^T h_{\boldsymbol{\theta}}(\boldsymbol{X}) = I_d$
  - OTOH, collapsed features have (at most) a rank-one covariance.
- This gap can be characterised by rate of decay of feature eigenvalues $\{\lambda_i\}_{i=1}^d$:
  - If we have **power-law** decay, $\lambda_i = \frac{1}{i^\beta}, \quad \forall 1 \leq i \leq d$ with exponent $\beta \geq 0$ then:
    $\beta = 0$ denotes **whitened** features and $\beta = \infty$ denotes **collapsed** features.



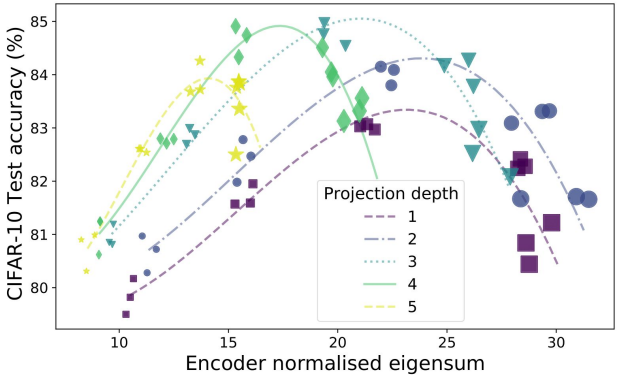
Factors affecting degree of collapse/whitening

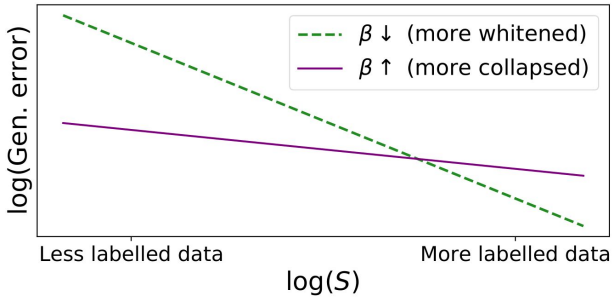
- Using our insights from power-law decay, we prove (for linear NNs) that:
  - (Commonly found) projection layers enable encoders to observe decaying eigenspectra.
  - Deeper projection layers lead to more collapsed encoder features.
  - Stronger Barlow Twins regularisation encourages more whitened features.



The effect of whitening on generalisation

- By varying projection depth and regularisation strength, we empirically show that more whitened features do not necessarily generalise better (above right).
- We prove that more collapsed features may generalise better in regimes of **low-labelled data**; a common setting in SSL.

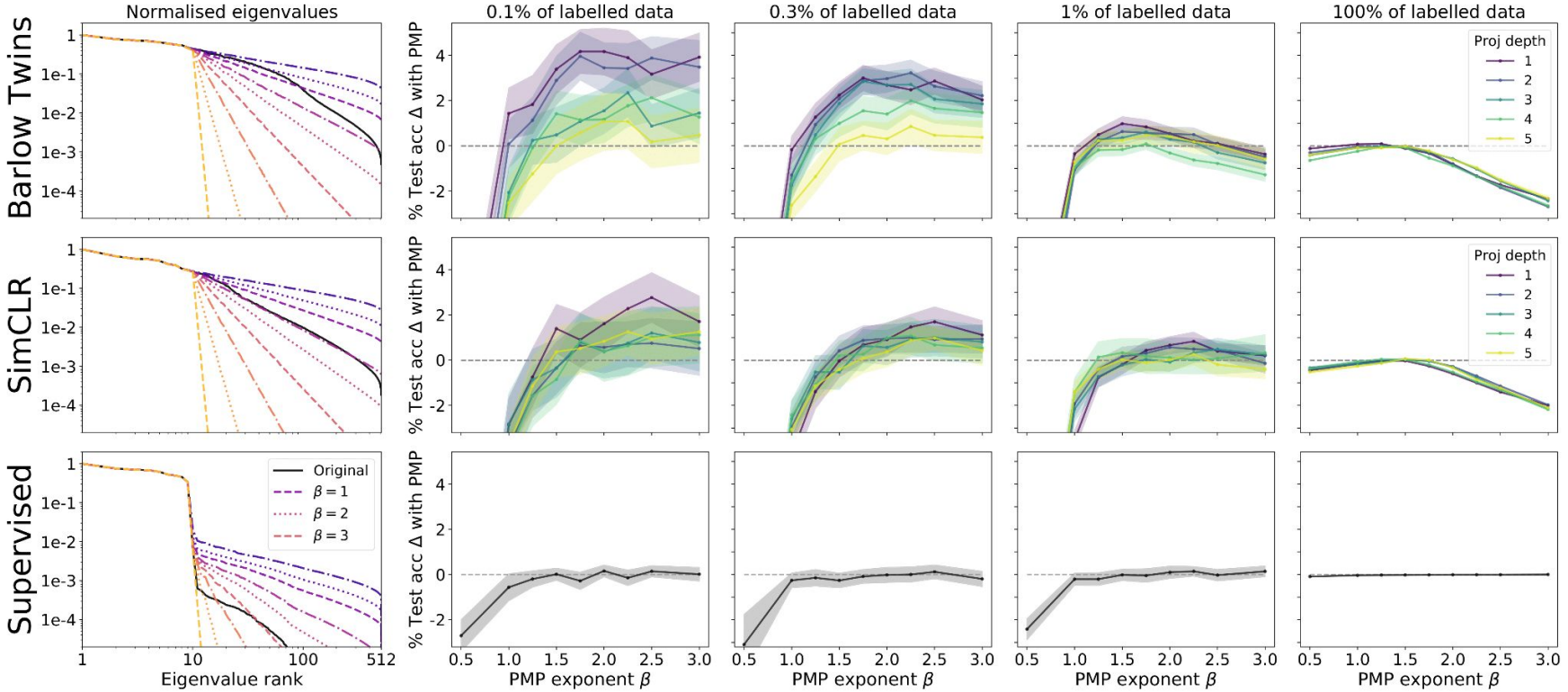




Method: **Post**-hoc **Man**ipulation of the **P**rinciple **A**xes and **T**race

- Propose **PostMan-Pat** (PMP) to control rate of feature eigenvalue decay, and deliver improved label efficiency in SSL.
  - Given a pretrained encoder $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}^d$ and power-law exponent $\beta \geq 0$ PMP computes an eigenspectrum-dependent rescaling matrix $W_{\mathrm{PMP}} \in \mathbb{R}^{d \times d}$
  - PMP defines a new encoder: $h_{\mathrm{PMP}}(\boldsymbol{x}) \leftarrow h_{\boldsymbol{\theta}}(\boldsymbol{x}) W_{\mathrm{PMP}}$ for linear evaluation.
  - $h_{\mathrm{PMP}}(\boldsymbol{x})$ is constructed to have power-law behaviour in eigenvalues, with exponent β acting as a hyperparameter that controls feature eigenvalue decay.

PMP Experiments: CIFAR-10



PMP Experiments: ImageNet-1K

1) Low-labelled data (w. ResNet-50)

Method		Top-1			Top-5		
Pretrain	Eval	0.3%	1%	10%	0.3%	1%	10%
SimCLR (69.3)	LP	$34.2_{\pm.2}$	48.1	61.0	$57.2_{\pm.3}$	73.8	84.3
	MLP	$31.8_{\pm.4}$	45.2	61.5	$54.1_{\pm.3}$	71.1	85.0
	PMP	$35.9_{\pm.2}$	50.9	62.5	$57.9_{\pm.2}$	76.6	85.2
	NFT	**39.8**$_{\pm.2}$	**52.5**	**67.5**	**65.5**$_{\pm.2}$	**78.9**	**88.7**
SwAV (74.7)	LP	$36.5_{\pm.3}$	53.8	68.2	$61.8_{\pm.1}$	78.8	88.8
	MLP	$34.6_{\pm.4}$	52.0	67.5	$59.3_{\pm.2}$	77.2	88.6
	PMP	**39.3**$_{\pm.3}$	**55.9**	68.5	**64.1**$_{\pm.2}$	**79.7**	89.1
	NFT	$32.4_{\pm.3}$	53.6	**70.8**	$57.8_{\pm.2}$	79.1	**90.5**
Barlow (73.5)	LP	$39.9_{\pm.1}$	55.0	63.2	$63.8_{\pm.2}$	79.0	83.4
	MLP	$37.6_{\pm.1}$	53.0	66.3	$61.5_{\pm.1}$	76.9	87.2
	PMP	**42.3**$_{\pm.1}$	**56.2**	67.3	**65.8**$_{\pm.1}$	**79.7**	88.6
	NFT	$40.7_{\pm.1}$	55.3	**70.0**	**65.8**$_{\pm.1}$	79.6	**89.9**
Supervised		–	25.4	56.4	–	48.4	80.4

2) Transfer learning

Method		Transfer Dataset		
Pretrain	Eval	C-100	Cars	Flowers
SimCLR	LP	65.26	46.88	84.65
	PMP	66.13	47.83	85.88
Barlow	LP	74.19	69.36	92.29
	PMP	75.10	69.67	92.54
SwAV	LP	75.24	63.39	90.47
	PMP	76.10	64.46	92.00

3) Different encoder architectures (ViT)

	Top-1			Top-5		
Eval	0.3%	1%	10%	0.3%	1%	10%
LP	$54.0_{\pm.3}$	64.5	72.3	$78.0_{\pm.3}$	86.6	91.2
PMP	$55.2_{\pm.2}$	65.1	72.4	$78.7_{\pm.2}$	86.8	91.2

Conclusion

- ➢ We highlight the spectrum that exists between collapsed and whitened features in SSL:
  - Parameterise by power-law behaviour
  - Influential hyperparameters: projection depth and regularisation strength.
  - Implications for generalisation, particularly in low-labelled data regimes
- ➢ Propose **Post**-hoc **Man**ipulation of the **P**rinciple **A**xes and **T**race (PostMan-Pat or PMP):
  - Controls rate of decay of feature eigenvalues.
  - Consistently outperforms linear probing, and often outperforms fine-tuning on low-labelled data.

Limitations/future work:

- ★ Devising (self-)supervised learning methods that consider feature eigenspectra directly.
- ★ Extending PMP to fine-tuning settings.
- ★ PMP doesn't consider the actual features themselves, just the importance of their relative weighting.
- ★ Performance depends on power-law β hyperparameter, can we automatically tune this?

Thanks for watching! 😊