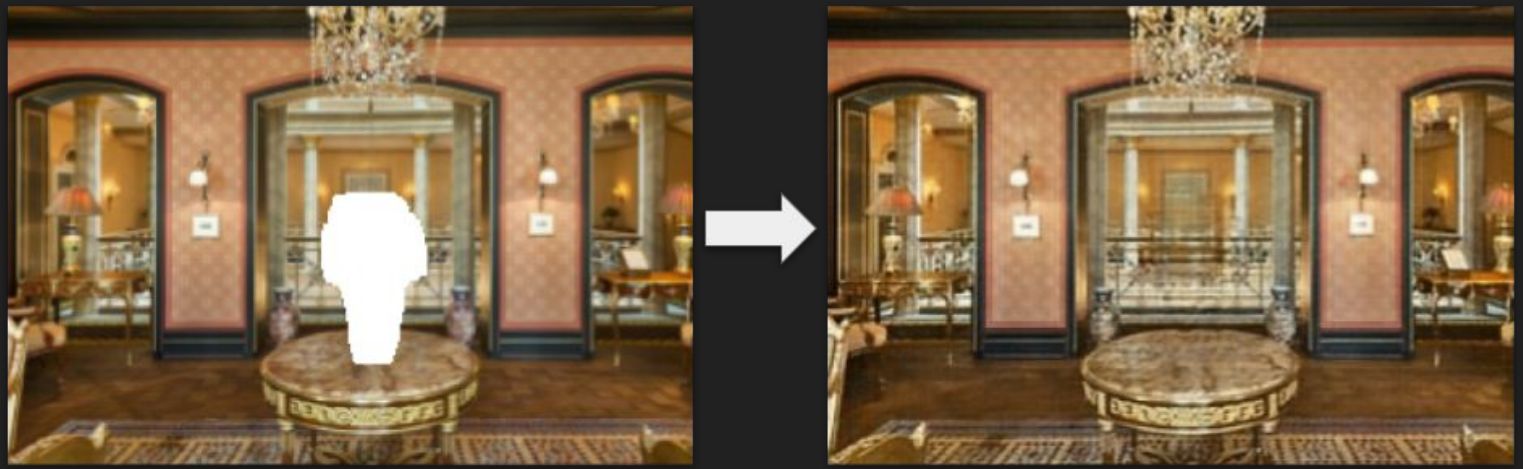
Exploring generative model prior for self-supervised learning

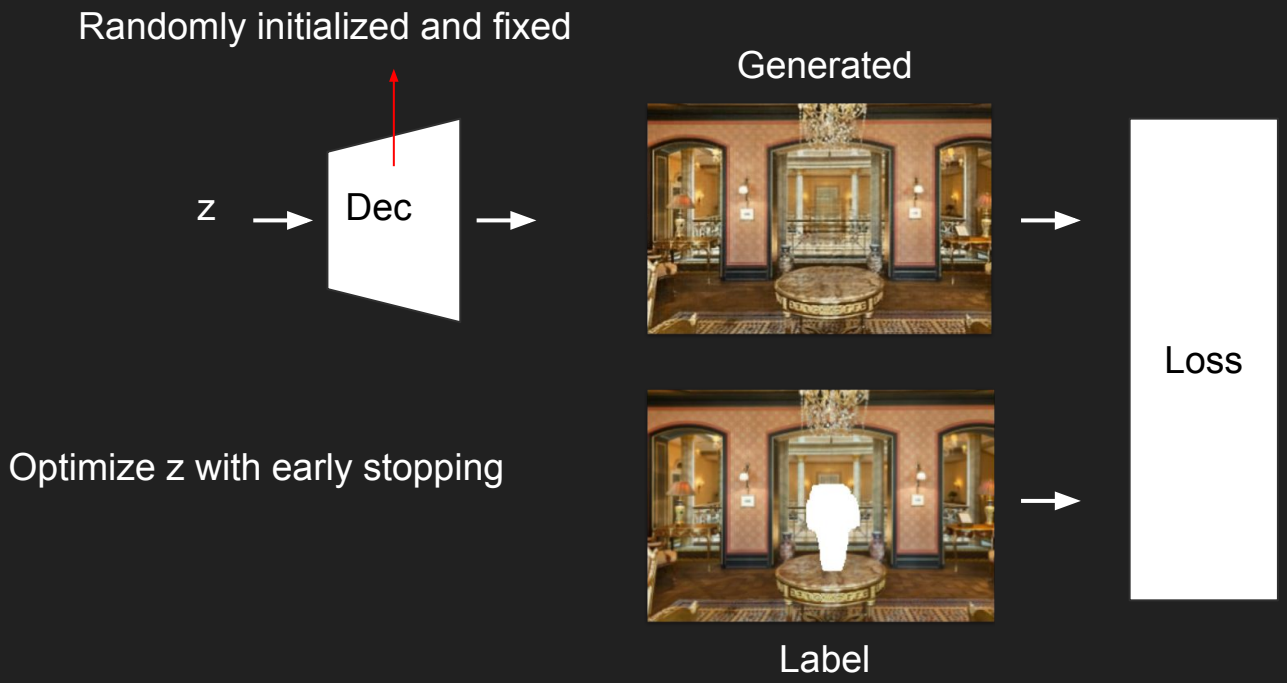
Zhen Liu, Yutong Yan

Motivation: Deep Image Prior[1]

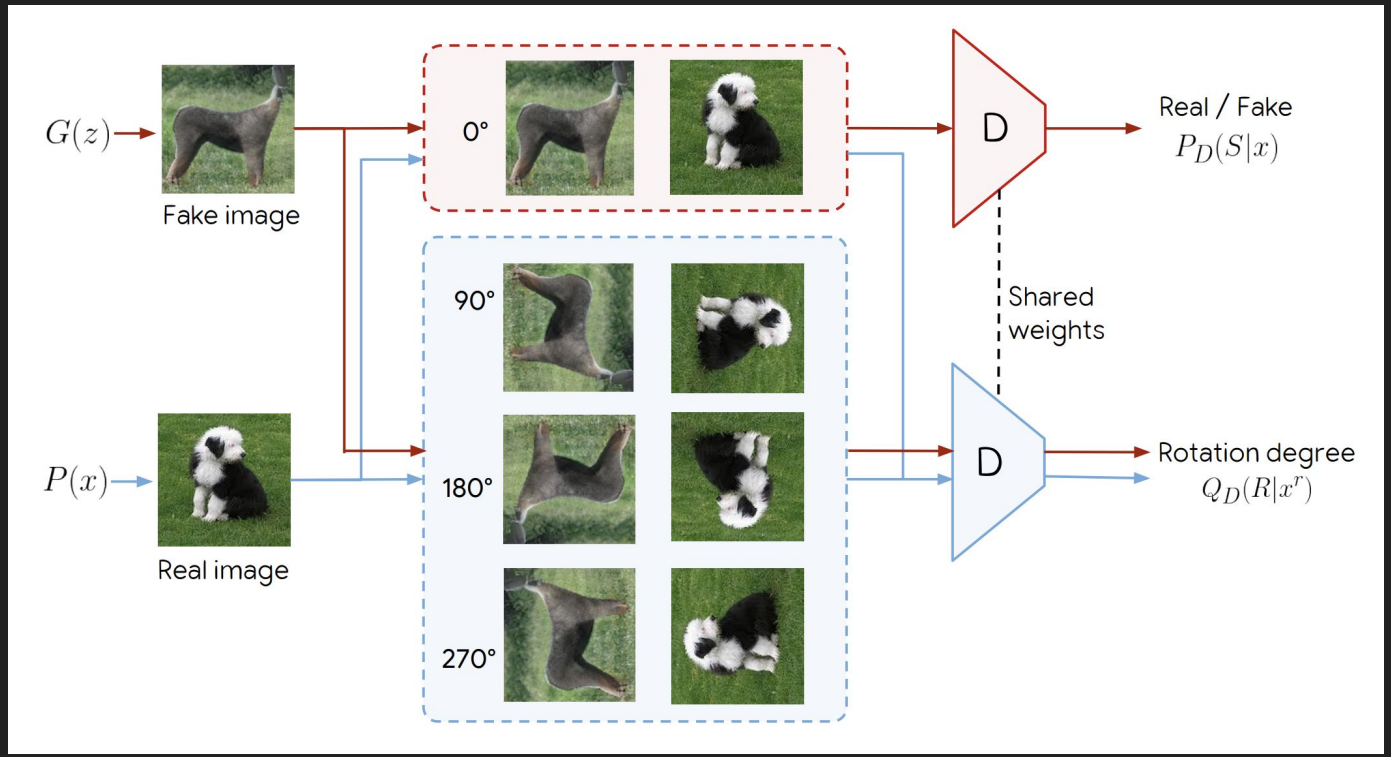


[1] Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. arXiv preprint arXiv:1711.10925 (2017)

Motivation: Deep Image Prior



SSL for GAN works



Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 12154–12163, 2019.

Can we improve self-supervised learning with generative models?

Idea #1: Generator as augmentation

Construct semantically similar images with different appearances

1. Simple augmentation - add augmented images to the dataset

2. Auxiliary contrastive loss from another view

Image generation via Langevin dynamics

Random Noise → Generated Image

$$I_{t+1} = I_t + \alpha \nabla f(I_t) + \omega, \quad \omega \sim \mathcal{N}(0, \sigma)$$

Energy-based model

Instead of running inverse Langevin dynamics, we generate noisy images by

$$I_0 = \beta I_{gt} + (1 - \beta)\epsilon, \quad \epsilon \sim U(0, 1)$$

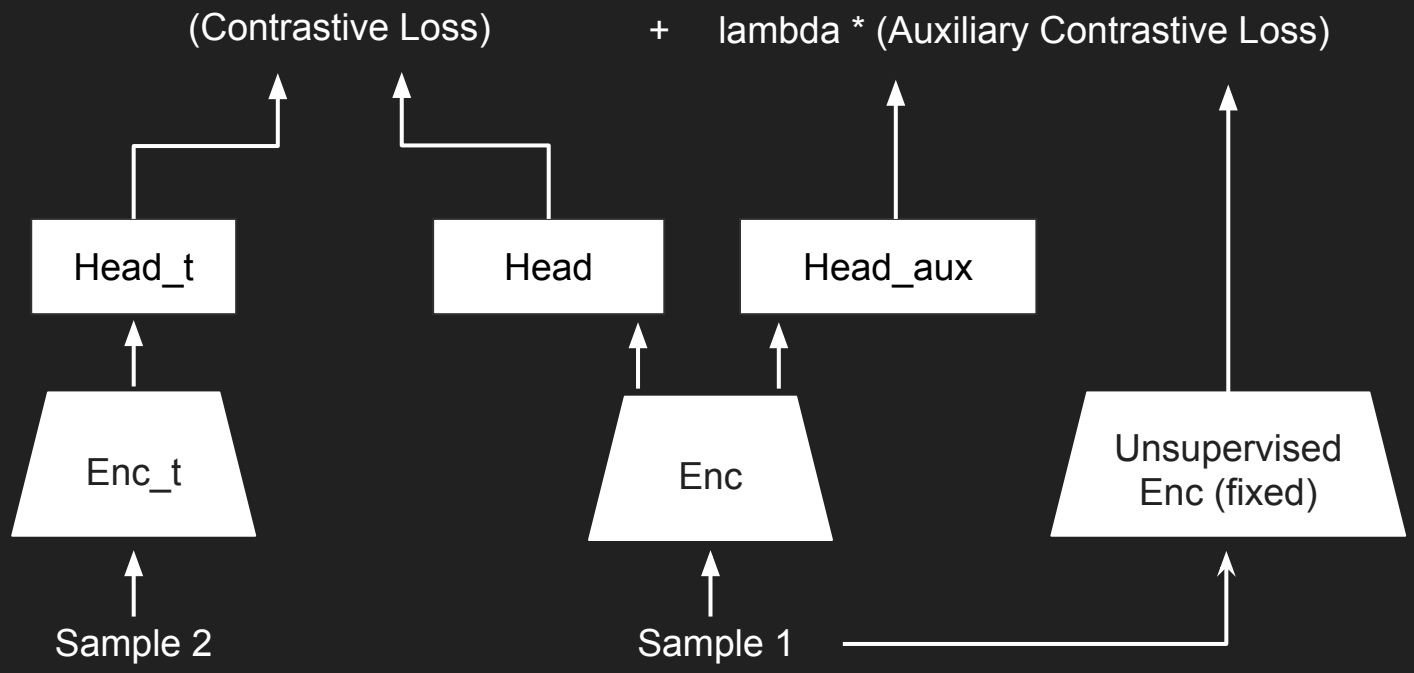
Y. Du and I. Mordatch. Implicit generation and generalization in energy-based models. arXiv preprint arXiv:1903.08689, 2019.
Bo Dai, Zhen Liu, Hanjun Dai, Niao He, Arthur Gretton, Le Song, and Dale Schuurmans. Exponential family estimation via adversarial dynamics embedding. arXiv preprint arXiv:1904.12083, 2019.

Idea #2: Discriminator as another view

As it is trained with generator (decoder), the discriminator (encoder) contains priors from the generator

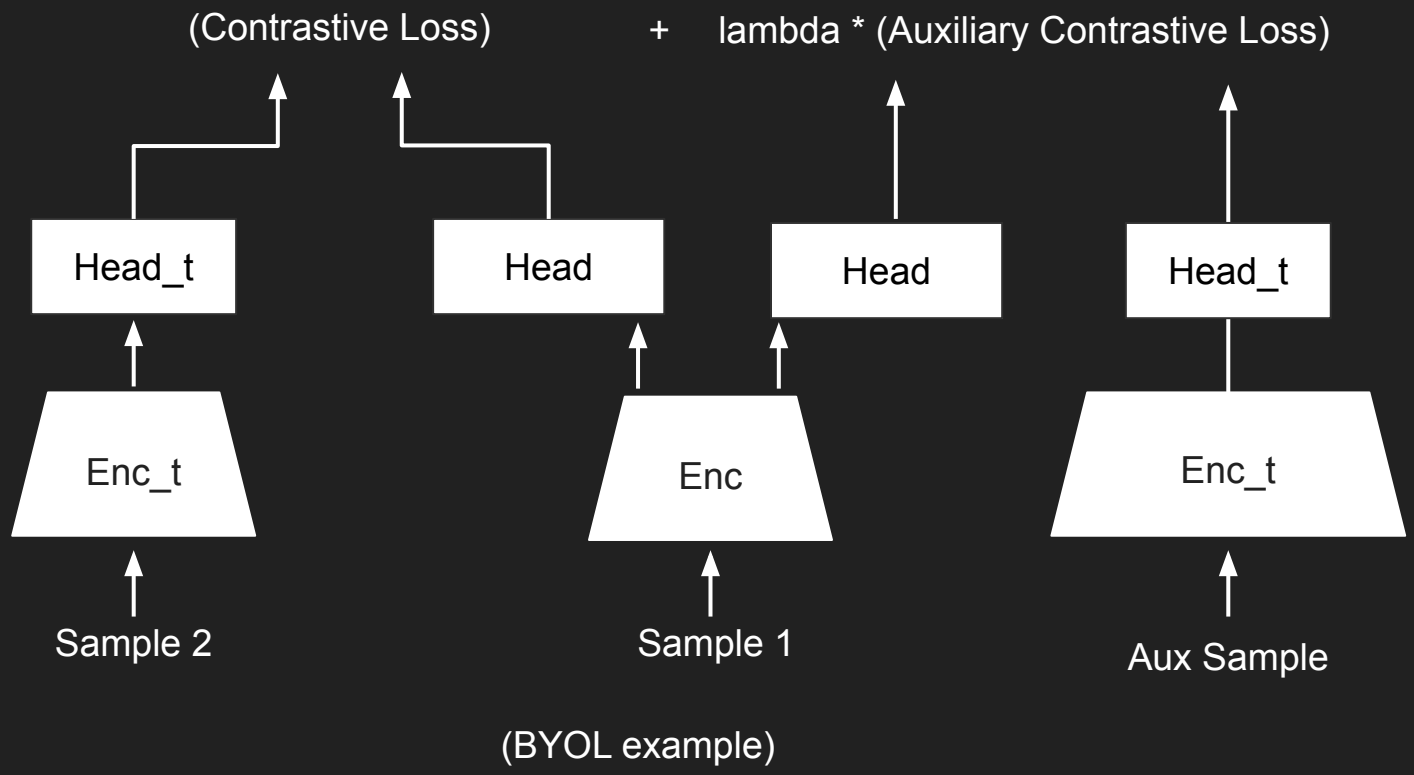
- Treat the output feature map of discriminator as a ‘view’

Idea #2: Discriminator as another view



(BYOL example)

3.1.2



(BYOL example)

Experiment settings

Train for 1000 epochs with base lr 3e-3 and embedding size 64

For idea #1, vary the following parameters: 1. Langevin dynamics step size, 2. # of steps, 3. std of noise, 4. lambda coeff in loss, 5. coeff for noisy image interpolation

Default setting - # Step=15, step size=10, noise std=1.0, lambda=0.01 and alpha=0.6

We use batch size 256 for BYOL and 512 for SimCLR

Ablation study - #1

	# Steps = 5	# Steps = 10	# Steps = 15	# Steps = 20	Baseline
SimCLR	86.75	87.62	87.82	88.11	91.81
BYOL	89.68	90.57	90.14	89.12	90.20

Ablation study - #1

	Alpha = 0.5	Alpha = 0.6	Alpha = 0.7	Baseline
SimCLR	87.88	87.82	88.22	91.81
BYOL	90.55	90.45	90.53	90.20

	std = 0.5	std = 1.0	std = 2.0	Baseline
SimCLR	88.53	87.82	87.73	91.81

Ablation study - #1

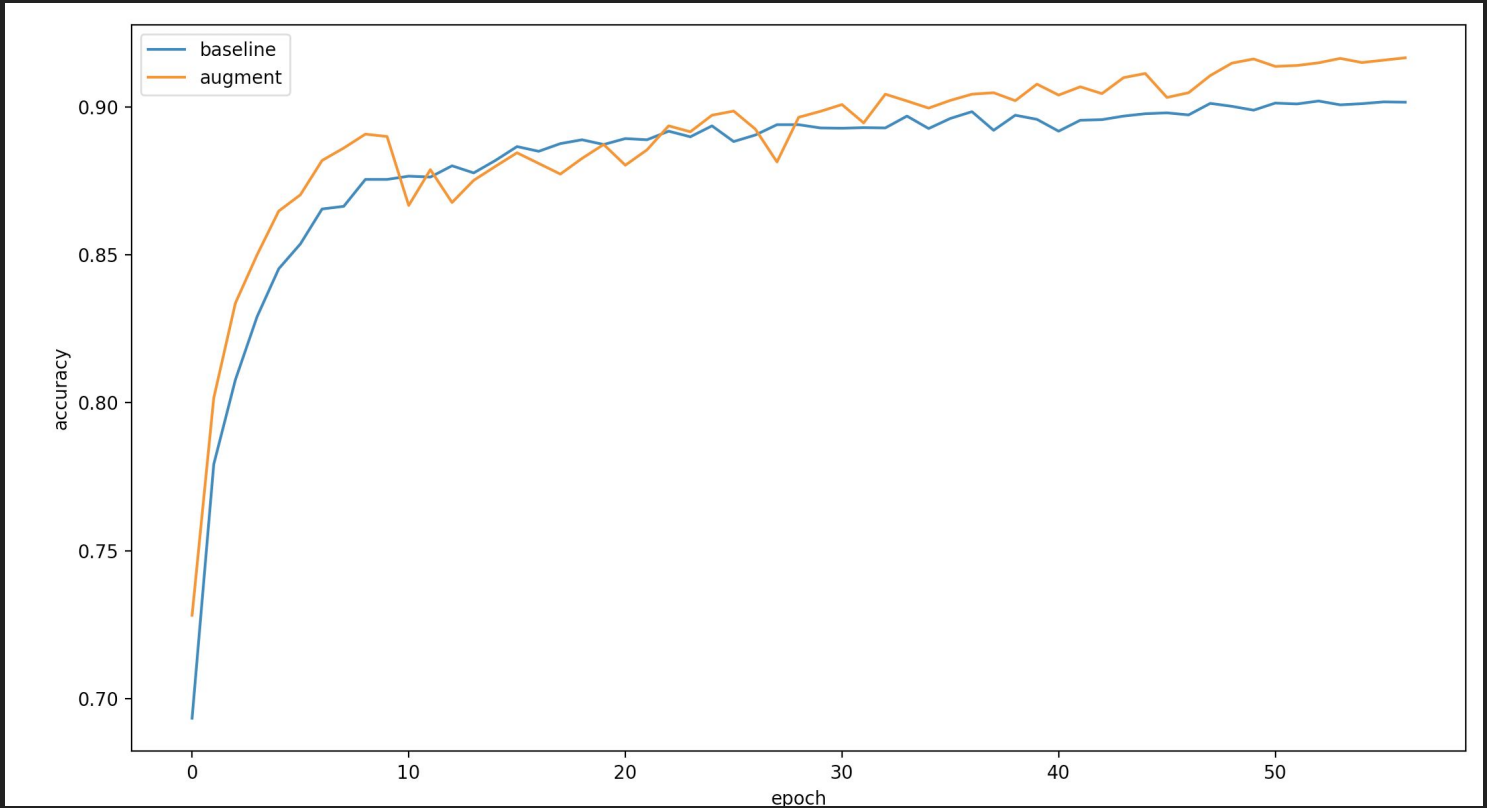
	Coeff = 0.001	Coeff = 0.01	Coeff = 0.1	Baseline
BYOL	89.12	90.57	89.21	90.20

	Step size = 5	Step size = 10	Step size = 20	Baseline
SimCLR	87.69	87.82	88.21	91.81

Augmentation with discriminator features

	Coeff = 0.001	Coeff = 0.01	Coeff = 0.1	Coeff = 1.0	Baseline
SimCLR	91.78	91.69	91.66	N/A	91.81
BYOL	N/A	91.66	92.13	91.50	90.20

Augmentation with discriminator features



Conclusion

- Direct augmentation with generated images does not seem to work easily
  - Image artifacts due to (class-)unconditional generative model
  - Hard to control semantics

- Additional view from pretrained discriminators seems to improve convergence for BYOL-like methods

Thanks!