NEURAL INFORMATION PROCESSING SYSTEMS | 2022

清華大學 Tsinghua University

THE UNIVERSITY of EDINBURGH

Distilling Representations from GAN Generator via Squeeze and Span

Yu Yang[1*], Xiaotian Cheng[1*], Chang Liu[1], Hakan Bilen[2], Xiangyang Ji[1]

[1] Tsinghua University [2] The University of Edinburgh

Yu Yang

Xiaotian Cheng

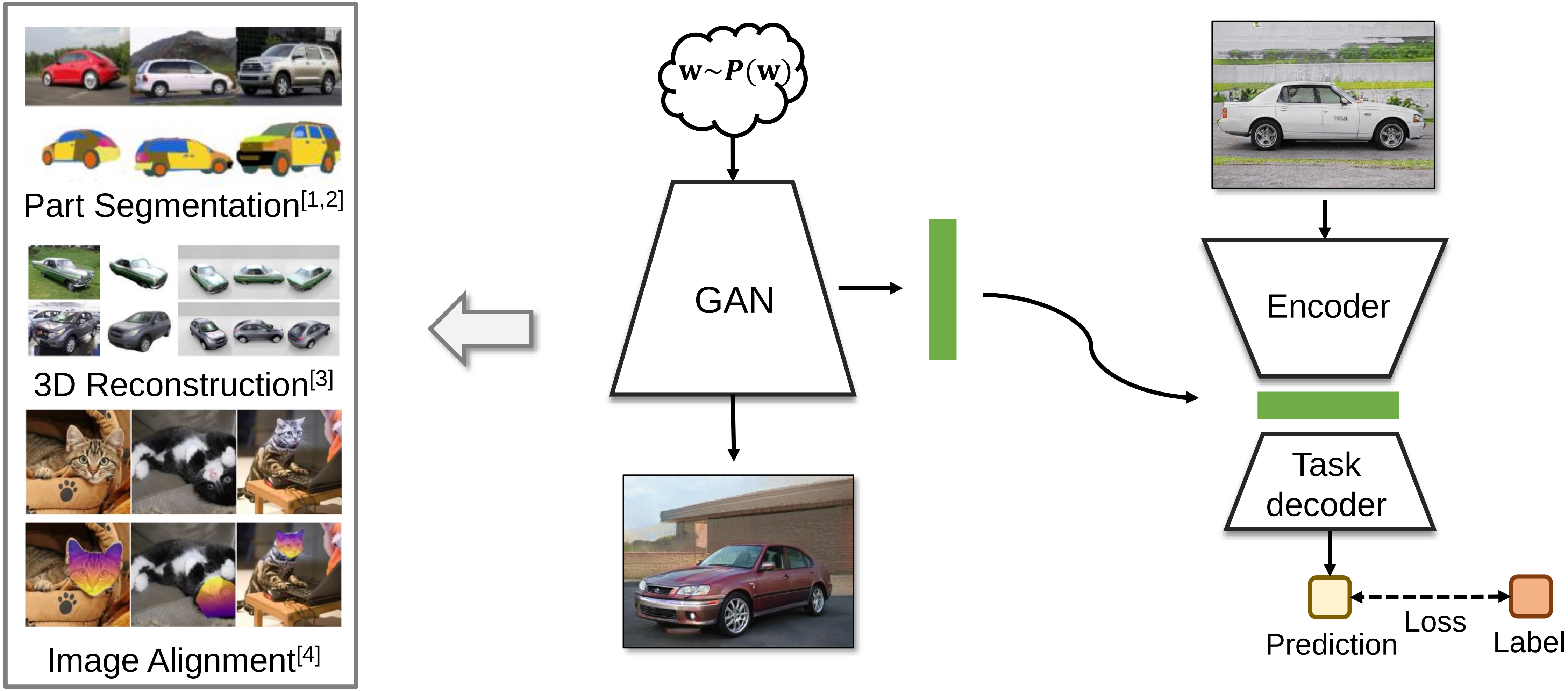
Chang Liu

Hakan Bilen

Xiangyang Ji

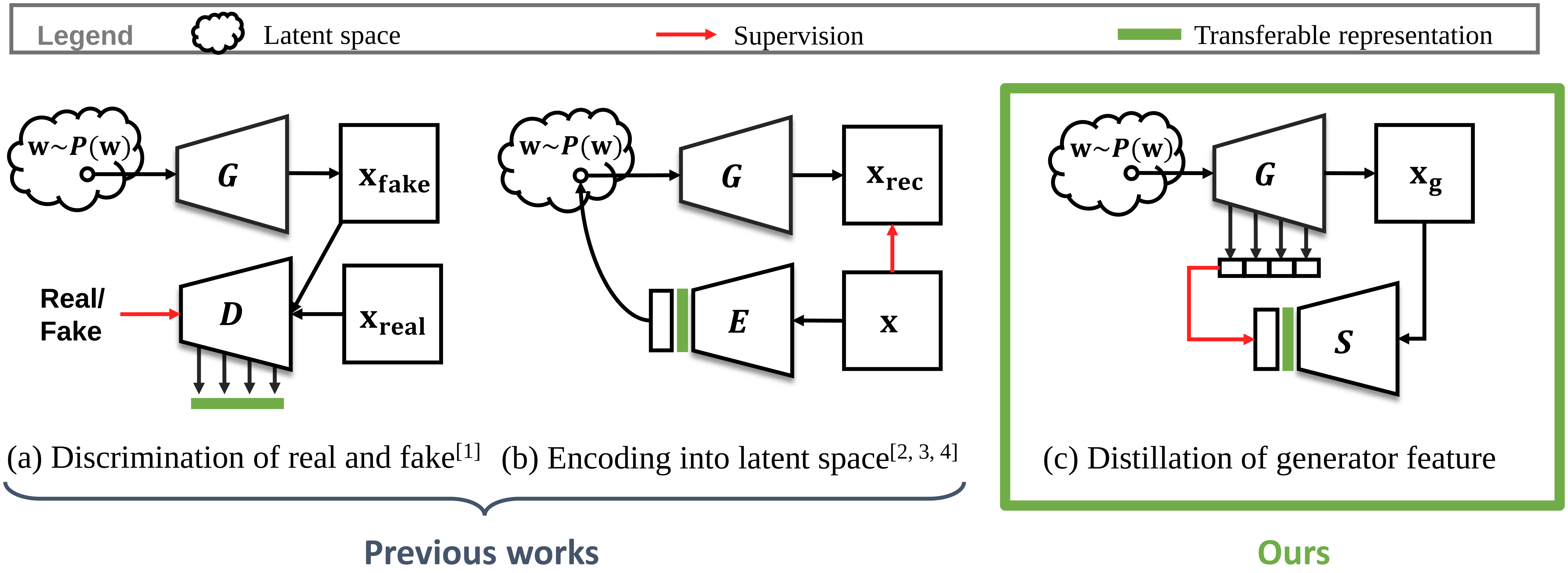
Introduction

GAN's application　　Image synthesis　　Representation learning



[1] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In CVPR, 2021.
[2] Yu Yang, Xiaotian Cheng, Hakan Bilen, and Xiangyang Ji. Learning to annotate part segmentation with gradient matching. In ICLR, 2022.
[3] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In ICLR, 2021.
[4] William Peebles, Jun-Yan Zhu, Richard Zhang, Antonio Torralba, Alexei Efros, and Eli Shechtman. Gan-supervised dense visual alignment. In CVPR, 2022.

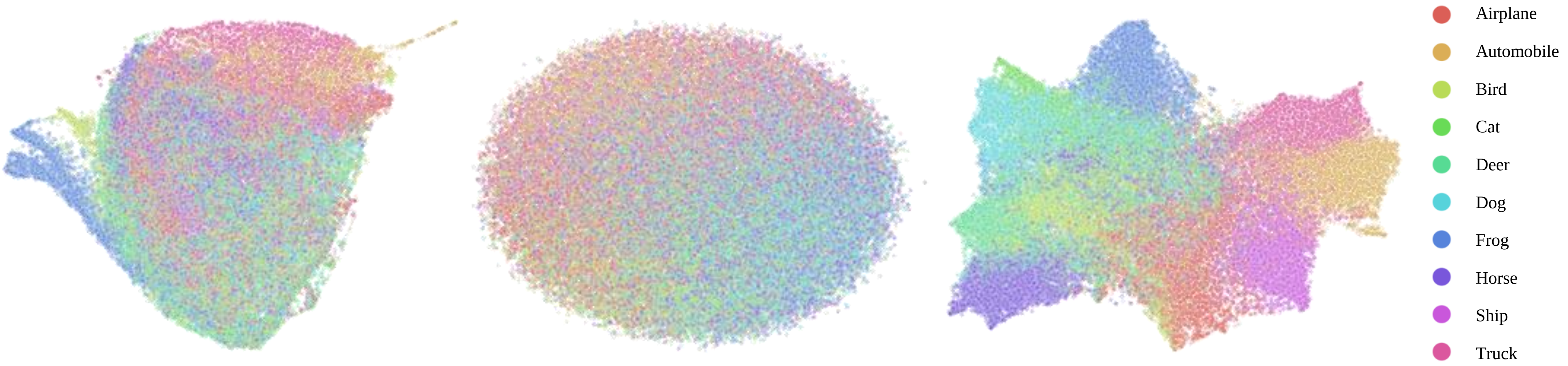
Method



[1] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016.
[2] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In ICLR, 2017.
[3] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In ICLR, 2017.
[4] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In NeurIPS, 2019.

Method

Visualization of three types of GAN representations



(a) Discriminator feature (b) Latent variable (c) Generator feature

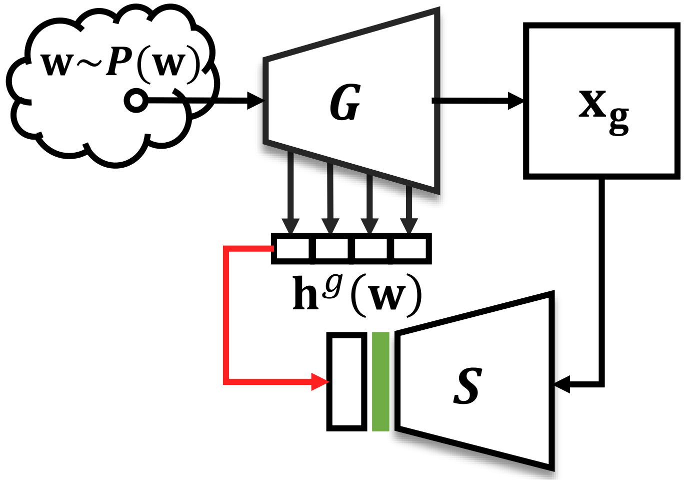
Method

- Squeeze-and-span representations from GAN generator
  - GAN distillation optimization

$$\min_{\theta} \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w})} \left\| S_\theta\left(G(\mathbf{w})\right) - \mathbf{h}^g(\mathbf{w}) \right\|_2^2$$

Problem:

Redundant representation

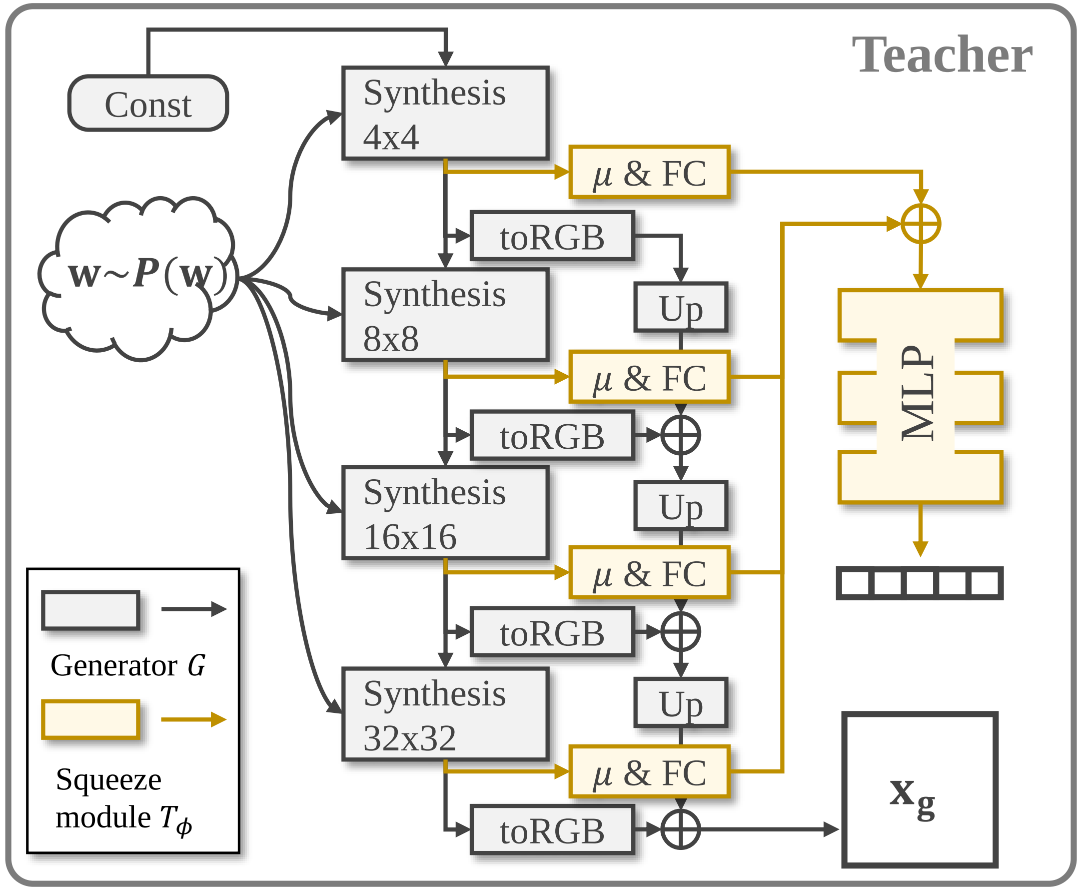


Method

- Squeeze-and-span representations from GAN generator
  - GAN distillation optimization

$$\min_{\theta} \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w})} \left\| S_\theta\left(G(\mathbf{w})\right) - \mathbf{h}^g(\mathbf{w}) \right\|_2^2$$

  - Squeezing informative representations
    - Use an MLP as squeeze module T_ϕ



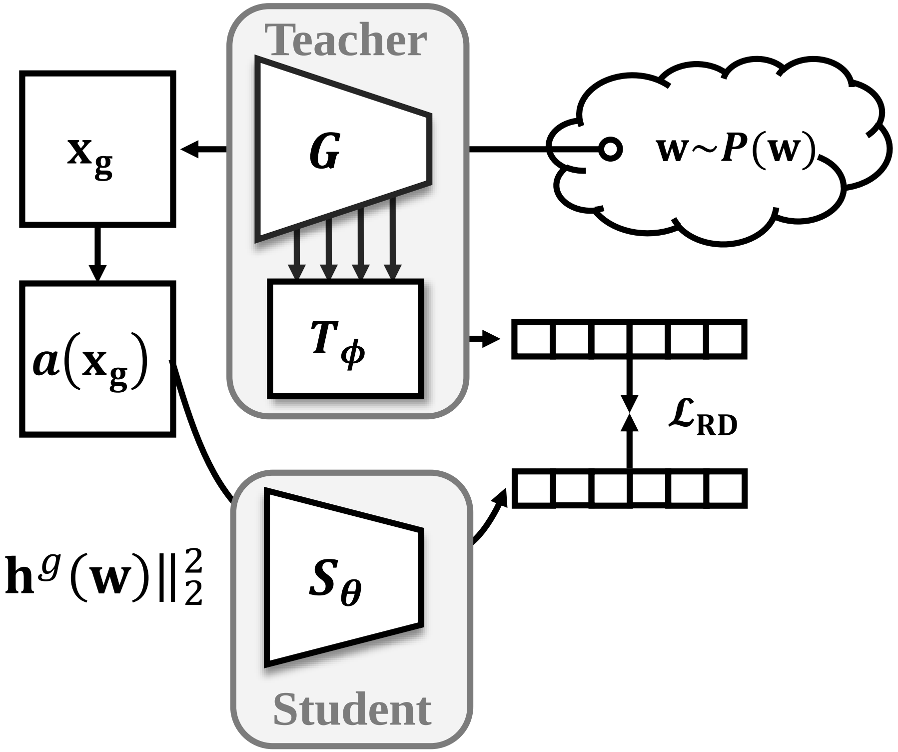
Method

- Squeeze-and-span representations from GAN generator
  - GAN distillation optimization

$$\min_{\theta} \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w})} \left\| S_\theta\left(G(\mathbf{w})\right) - \mathbf{h}^g(\mathbf{w}) \right\|_2^2$$

  - Squeezing informative representations
    - Use an MLP as squeeze module T_ϕ
    - Rewrite main objective as

$$\min_{\theta,\phi} \mathcal{L}_{\mathrm{RD}} = \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w}), a\sim\mathcal{A}} \left\| S_\theta(a[G(\mathbf{w})]) - \mathbf{h}^g(\mathbf{w}) \right\|_2^2$$



Problem:

The network may degenerate to output constant.

Method

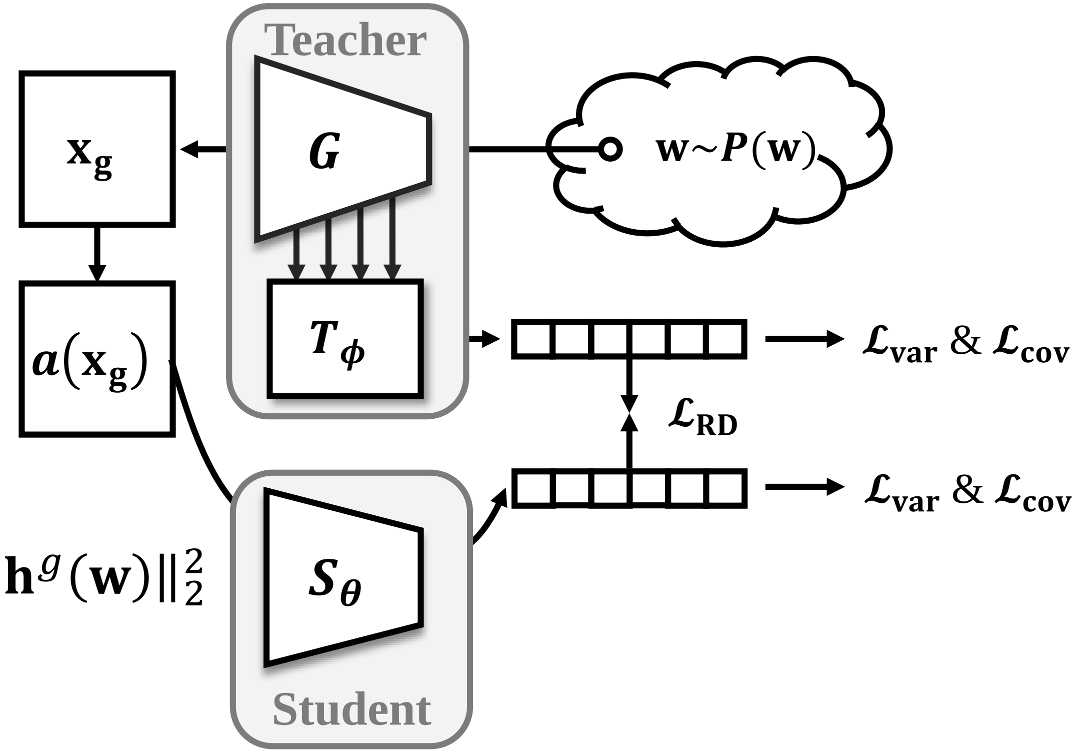
- Squeeze-and-span representations from GAN generator
  - GAN distillation optimization

$$\min_{\theta} \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w})} \left\| S_{\theta}\big(G(\mathbf{w})\big) - \mathbf{h}^{g}(\mathbf{w}) \right\|_2^2$$

  - Squeezing informative representations
    - Use an MLP as squeeze module T_{ϕ}
    - Rewrite main objective as

$$\min_{\theta,\phi} \mathcal{L}_{\mathrm{RD}} = \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w}),a\sim\mathcal{A}} \| S_{\theta}(a[G(\mathbf{w})]) - \mathbf{h}^{g}(\mathbf{w}) \|_2^2$$

    - Add variance and covariance loss[1]



$$\mathcal{L}_{\mathrm{var}}(Z) = \frac{1}{M}\sum_{j=1}^{M} \max\left(0, 1 - \sqrt{\mathrm{Var}(z^{j}) + \varepsilon}\right)$$

$$\mathcal{L}_{\mathrm{cov}}(Z) = \frac{1}{M}\textstyle\sum_{i\neq j}[C(Z)]_{ij}^{2}, \text{ where}$$

$$C(Z) = \frac{1}{N-1}\textstyle\sum_{i=1}^{N}(z_i - \bar{z})(z_i - \bar{z})^{T}, \bar{z} = \frac{1}{N}\sum_{i}^{N} z_i$$

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In ICLR, 2022.

Method

- Squeeze-and-span representations from GAN generator
  - GAN distillation optimization

$$\min_{\theta} \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w})} \left\| S_\theta\big(G(\mathbf{w})\big) - \mathbf{h}^g(\mathbf{w}) \right\|_2^2$$

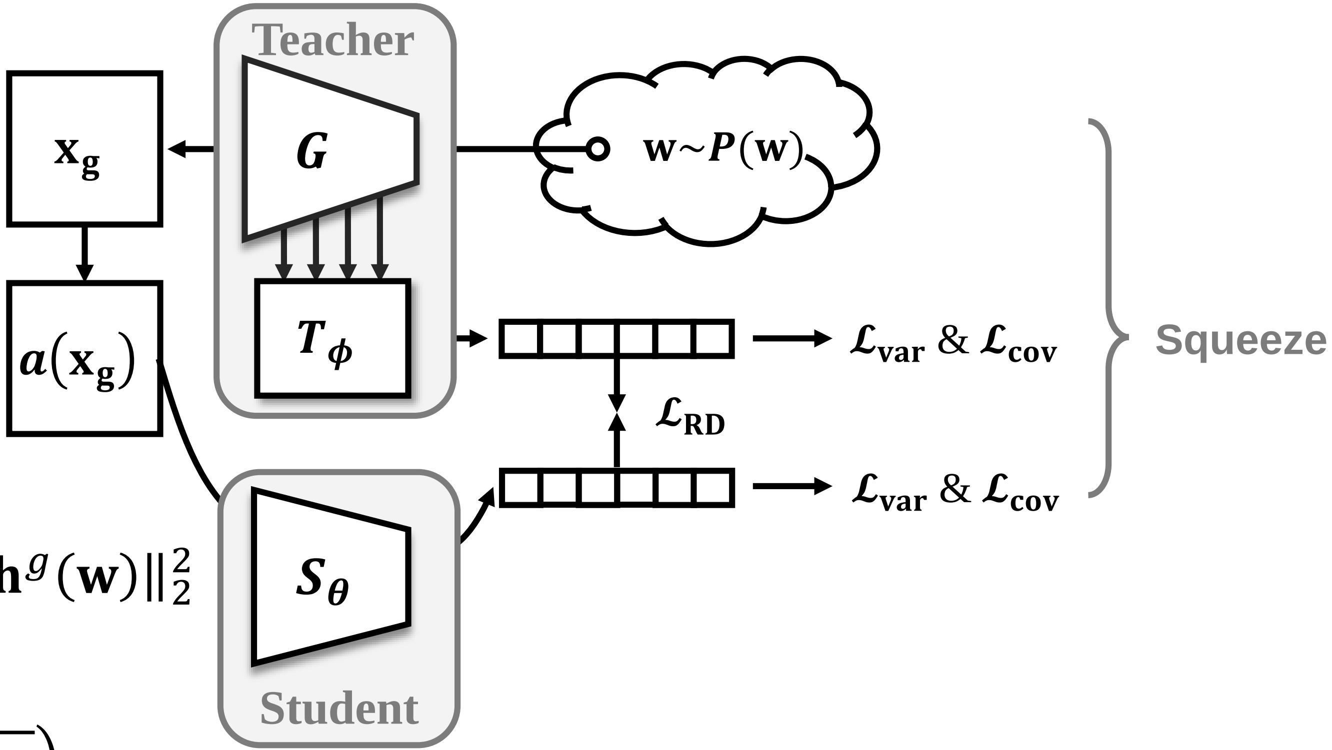
  - Squeezing informative representations
    - Use an MLP as squeeze module T_ϕ
    - Rewrite main objective as

$$\min_{\theta,\phi} \mathcal{L}_{\mathrm{RD}} = \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w}), a\sim\mathcal{A}} \| S_\theta(a[G(\mathbf{w})]) - \mathbf{h}^g(\mathbf{w}) \|_2^2$$

    - Add variance and covariance loss

$$\mathcal{L}_{\mathrm{var}}(Z) = \frac{1}{M} \sum_{j=1}^{M} \max\left(0, 1 - \sqrt{\mathrm{Var}(z^j) + \varepsilon}\right)$$

$$\mathcal{L}_{\mathrm{cov}}(Z) = \frac{1}{M} \sum_{i\neq j} [C(Z)]_{ij}^2$$



    - Squeezing loss: $\boldsymbol{\mathcal{L}_{\mathrm{squeeze}} = \lambda \mathcal{L}_{\mathrm{RD}} + \mu[\mathcal{L}_{\mathrm{var}}(Z_f) + \mathcal{L}_{\mathrm{var}}(Z_g)] + \nu[\mathcal{L}_{\mathrm{cov}}(Z_f) + \mathcal{L}_{\mathrm{cov}}(Z_g)]}$

Method

- Squeeze-and-span representations from GAN generator

  - GAN distillation optimization

$$\min_{\theta} \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w})} \left\| S_\theta\big(G(\mathbf{w})\big) - \mathbf{h}^g(\mathbf{w}) \right\|_2^2$$

  - Squeezing informative representations

    - Use an MLP as squeeze module T_ϕ

    - Rewrite main objective as

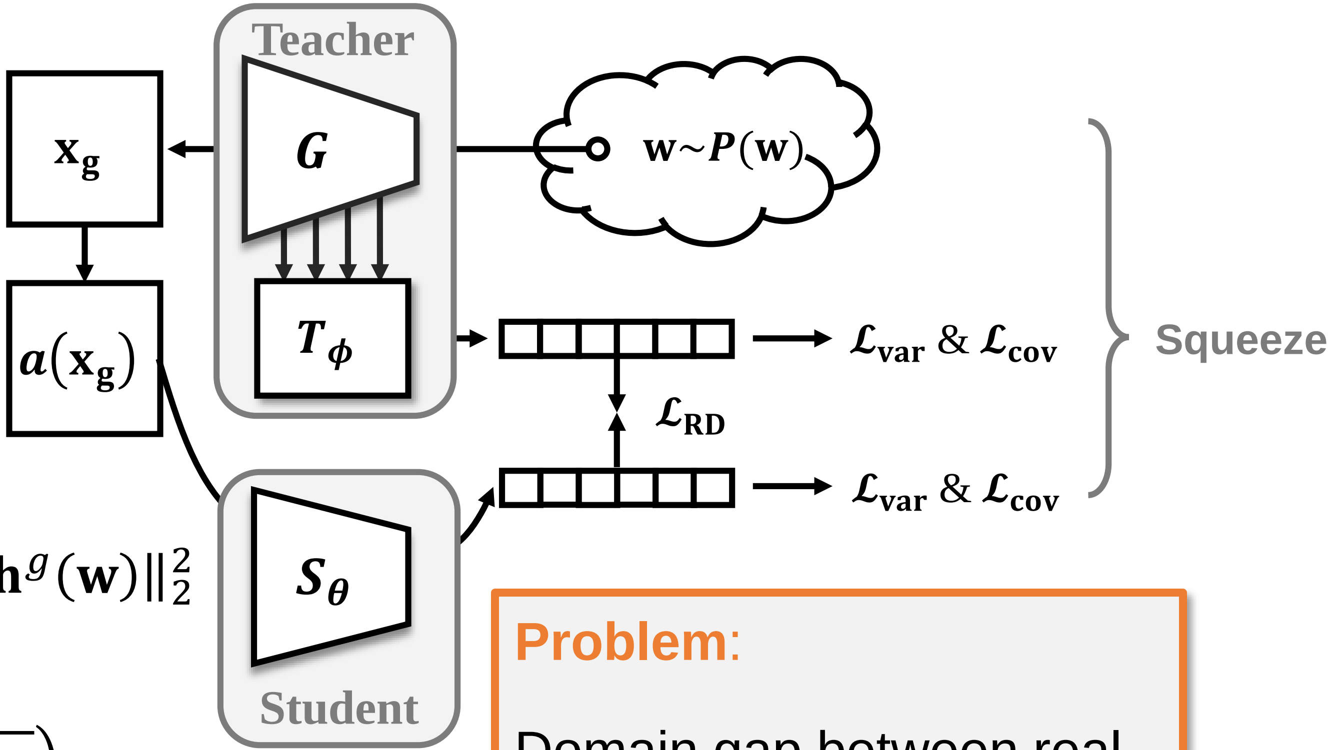
$$\min_{\theta,\phi} \mathcal{L}_{\mathrm{RD}} = \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w}),a\sim\mathcal{A}} \| S_\theta(a[G(\mathbf{w})]) - \mathbf{h}^g(\mathbf{w}) \|_2^2$$

    - Add variance and covariance loss

$$\mathcal{L}_{\mathrm{var}}(Z) = \frac{1}{M}\sum_{j=1}^{M} \max\left(0, 1 - \sqrt{\mathrm{Var}(z^j) + \varepsilon}\right)$$

$$\mathcal{L}_{\mathrm{cov}}(Z) = \frac{1}{M}\sum_{i\neq j}[C(Z)]_{ij}^2$$

    - Squeezing loss: $\mathcal{L}_{\mathbf{squeeze}} = \lambda\mathcal{L}_{\mathbf{RD}} + \mu[\mathcal{L}_{\mathbf{var}}(Z_f) + \mathcal{L}_{\mathbf{var}}(Z_g)] + \nu[\mathcal{L}_{\mathbf{cov}}(Z_f) + \mathcal{L}_{\mathbf{cov}}(Z_g)]$



Problem:

Domain gap between real and synthetic data

Method

- Squeeze-and-span representations from GAN generator
    - GAN distillation optimization

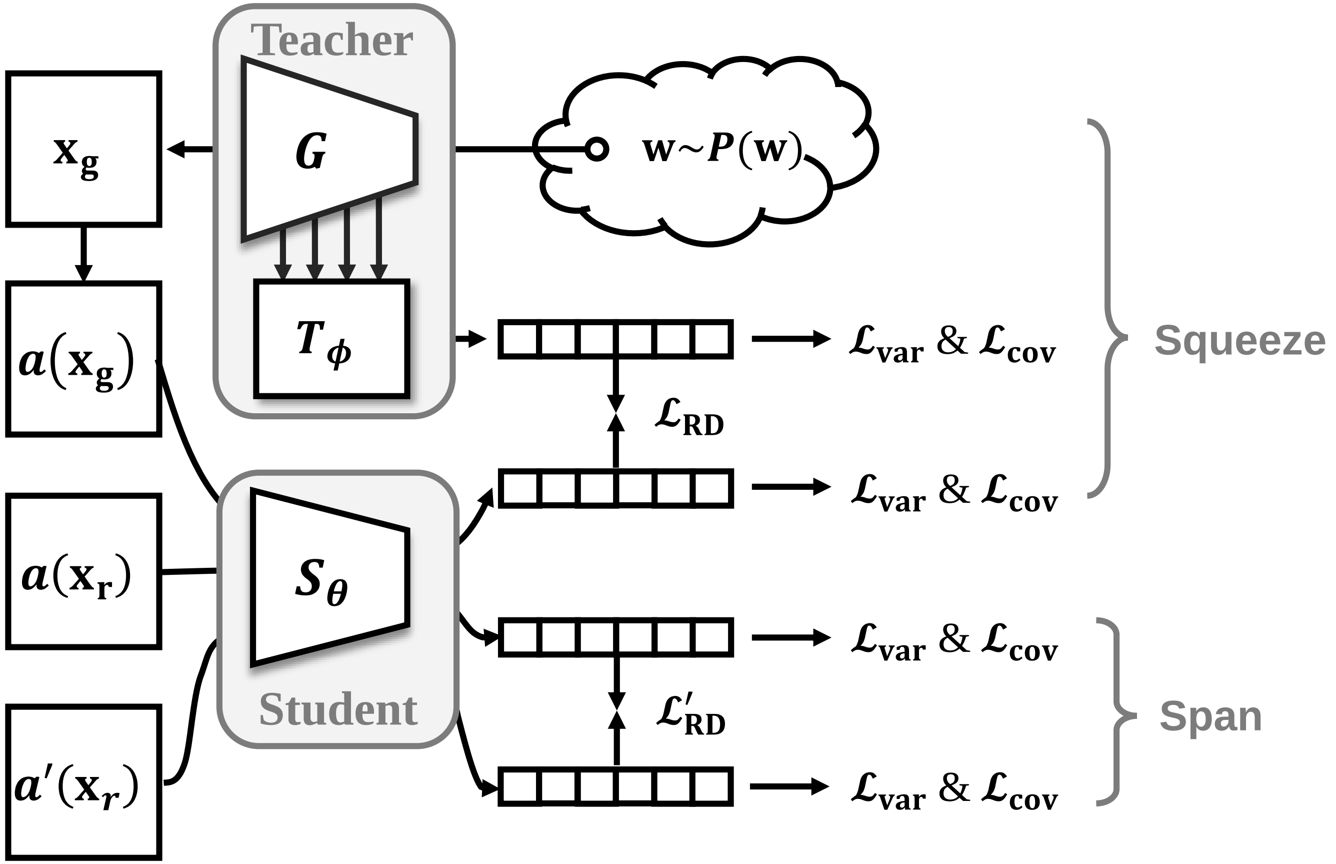
$$\min_{\theta} \mathbb{E}_{\mathbf{w}\sim P(\mathbf{w})} \left\| S_{\theta}\big(G(\mathbf{w})\big) - \mathbf{h}^{g}(\mathbf{w}) \right\|_2^2$$

    - Squeezing informative representations

$$\mathcal{L}_{\text{squeeze}}$$
$$= \lambda \mathcal{L}_{\text{RD}} + \mu\big[\mathcal{L}_{\text{var}}(Z_f) + \mathcal{L}_{\text{var}}(Z_g)\big]$$
$$+ \nu\big[\mathcal{L}_{\text{cov}}(Z_f) + \mathcal{L}_{\text{cov}}(Z_g)\big]$$

    - Spanning representations from synthetic to real domain

$$\mathcal{L}_{\text{span}}$$
$$= \lambda \mathcal{L}'_{\text{RD}} + \mu[\mathcal{L}_{\text{var}}(Z_r) + \mathcal{L}'_{\text{var}}(Z_r)]$$
$$+ \nu[\mathcal{L}_{\text{cov}}(Z_r) + \mathcal{L}'_{\text{cov}}(Z_r)]$$



$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{squeeze}} + (1 - \alpha)\mathcal{L}_{\text{span}}$$

Experiments

Representation distillation significantly outperforms *discriminator* and *encoding*

Knowledge Source	Transfer Method	Domain	CIFAR10	CIFAR100
Discriminator	Direct use (single feature)	Syn. & Real	63.81	30.11
	Direct use (multi-feature)	Syn. & Real	77.58	51.63
Latent variable	Encoding	Syn.	57.15	32.19
	Encoding	Syn. & Real	50.27	28.43
	Vanilla distillation (w/ aug)	Syn.	84.84	53.26
	Squeeze	Syn.	86.99	58.56
	Squeeze and span	Syn. & Real	90.95	66.17
Generator feature	Vanilla distillation (w/ aug)	Syn.	84.48	52.77
	Squeeze	Syn.	87.67	57.35
	Squeeze and span	Syn. & Real	92.54	67.87

Experiments

Squeeze and Span yields better results compared to SSL methods in linear classification

Pretrain Data	Methods	CIFAR10	CIFAR100	STL10	ImageNet100	ImageNet
Real	SimSiam [10]	<u>90.94</u>	62.44	71.30	–	–
	VICReg [3]	89.20	63.31	74.43	–	–
Syn	SimSiam [10]	85.11	47.89	73.38	–	–
	VICReg [3]	84.68	52.84	70.80	–	–
	Squeeze (Ours)	87.67	57.35	73.35	–	–
Real & Syn	SimSiam [10]	90.88	62.68	71.70	–	–
	VICReg [3]	90.46	<u>65.22</u>	<u>75.05</u>	<u>46.42</u>	<u>47.32</u>
	Sq & Sp (Ours)	**92.54**	**67.87**	**76.83**	**53.32**	**47.80**

Experiments

Better transferability in linear classification, object detection, surface normal estimation and semantic segmentation

Pre-training Data	Method	Aircraft	Caltech101	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food	Pets	SUN397	VOC2007	Avg.
ImageNet100	VICReg	**23.96**	60.29	15.27	80.28	57.11	45.95	60.26	33.40	38.41	29.53	49.07	44.86
(Syn.&Real)	Sq&Sp (Ours)	23.88	**63.59**	**15.30**	**84.37**	**61.28**	**49.36**	**63.41**	**37.80**	**43.28**	**33.04**	**55.64**	**48.26**
ImageNet	VICReg	32.39	79.49	22.37	90.09	70.67	58.88	79.97	50.55	59.47	45.76	**68.74**	59.85
(Syn.&Real)	Sq&Sp (Ours)	**33.85**	**80.65**	**25.71**	**90.14**	**70.81**	**61.75**	**80.09**	**50.84**	**61.01**	**46.34**	68.30	**60.86**

Pre-training Data	Method	VOC Detection			NYUv2 Surface Normal Estimation					ADE Semantic Segmentation	
		AP ↑	AP50 ↑	AP75 ↑	Mean ↓	Median ↓	11.25° ↑	22.5° ↑	30° ↑	Mean IoU ↑	Accuracy ↑
ImageNet100	VICReg	33.75	61.73	31.77	34.62	29.70	20.90	39.77	50.40	**0.3008**	73.19
(Syn.&Real)	Sq&Sp (Ours)	**41.10**	**69.07**	**42.54**	**33.09**	**27.47**	**22.90**	**42.71**	**53.47**	0.2993	**73.25**
ImageNet	VICReg	46.69	75.78	49.07	33.86	28.42	22.08	41.35	52.15	0.3263	74.85
(Syn.&Real)	Sq&Sp (Ours)	**48.98**	**77.50**	**52.85**	**33.39**	**28.30**	**22.27**	**41.65**	**52.26**	**0.3299**	**75.22**

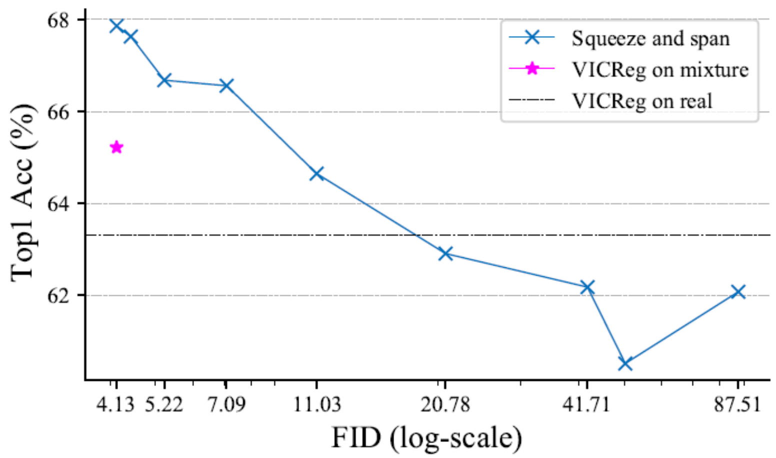
Experiments

For more results and discussion, please read our paper.

Discrepancy of representation

Methods	Pretrain Data	CIFAR10 ($\times 10^{-5}$)	CIFAR10- ($\times 10^{-5}$)	STL10 ($\times 10^{-3}$)
VICReg [3]	Syn	3.44	5.89	5.39
	Real & Syn	3.74	16.8	11.4
Squeeze (Ours)	Syn	4.79	1.24	9.82
Sq & Sp (Ours)	Real & Syn	0.45	0.25	3.71

Impact of generator quality



Ablation Study

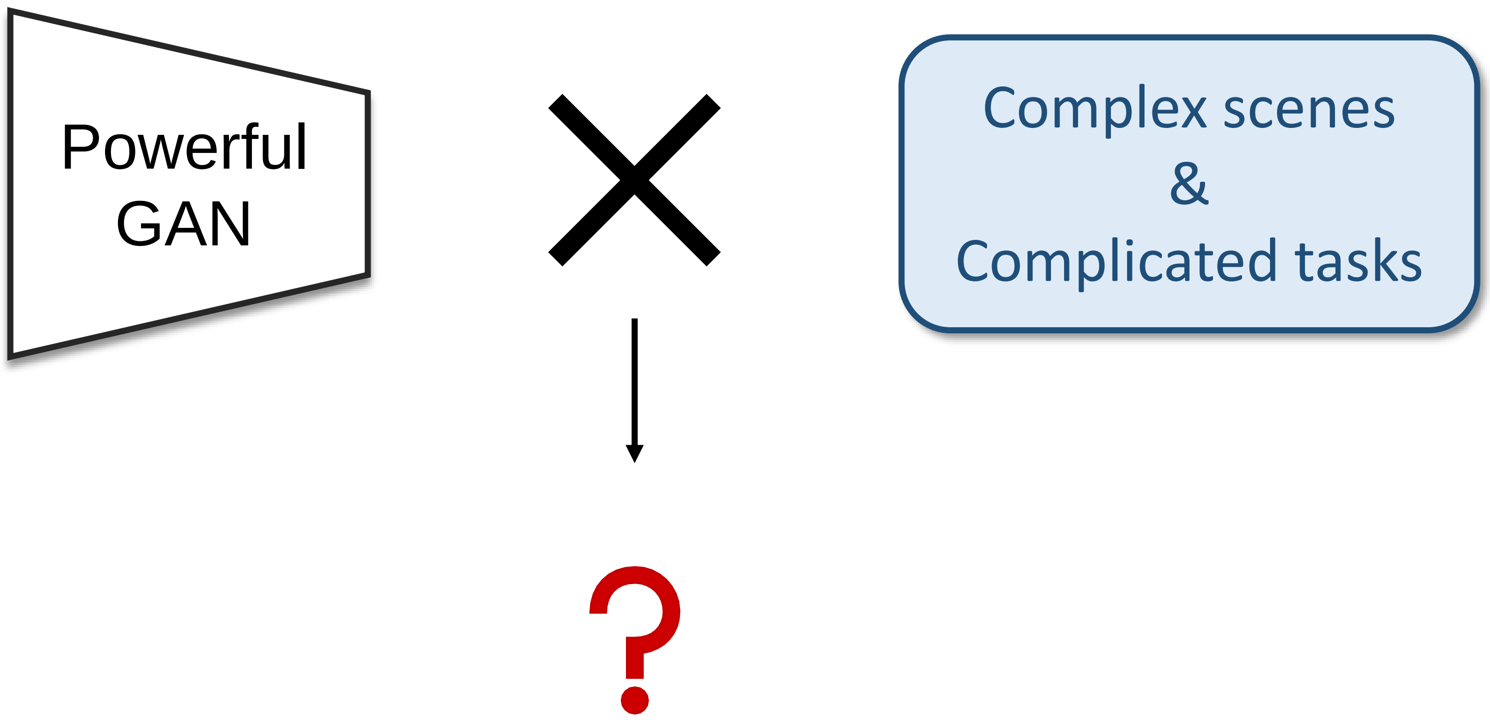
	$\mathcal{L}_{\mathrm{RD}}$	$\mathcal{A}$	T_ϕ	$\mathcal{L}_{\mathrm{var}}$	$\mathcal{L}_{\mathrm{cov}}$	Span	Top-1 Acc
a	✓						74.20
b	✓	✓					84.48
c	✓	✓	✓				10.00
d	✓	✓	✓	✓			79.10
e	✓	✓	✓	✓	✓		87.67
f	✓	✓	✓	✓	✓	✓	92.54

Impact of generator architecture

GAN architectures	Type	Sources	Generation FID	Squeeze Top-1 Acc
StyleGAN2-ADA	Unconditional	GitHub repo, model	2.92	87.67
AutoGAN	Unconditional	GitHub repo, model	12.42	76.28
StyleGAN2-ADA	Conditional	GitHub repo, model	2.42	**88.90**
StyleGAN-XL (StyleGAN3)	Conditional	GitHub repo, model	**1.85**	84.97
BigGAN-DiffAugment-cr	Conditional	GitHub repo, model	8.49	86.41

Future Work

Learning representations with stronger generalization and transferability with more powerful GANs.



Distilling Representations from GAN Generator via Squeeze and Span

NEURAL INFORMATION PROCESSING SYSTEMS

清華大學 Tsinghua University

THE UNIVERSITY of EDINBURGH

Code & models

https://github.com/yangyu12/squeeze-and-span

Yu Yang

Xiaotian Cheng

Chang Liu

Hakan Bilen

Xiangyang Ji