CHAIN: Enhancing Generalization in Data-Efficient GANs via lipsCHitz continuity constrAIned Normalization

Australian National University

Yao Ni[†], Piotr Koniusz[§,†]

[†]The Australian National University [§]Data61♥CSIRO

firstname.lastname@anu.edu.au

CSIRO

Summary

Background: Limited data in GANs causes discriminator overfitting and unstable training. Batch Normalization (BN) boosts generalization and training stability but is rarely used in data-efficient GAN discriminators as it impairs performance.

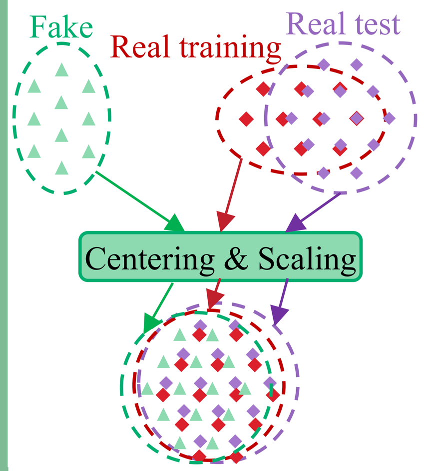
Goal: Integrate BN into data-efficient GAN discriminators.

Contributions:

- We derive a new error bound emphasizing reduced discriminator weight gradients to enhanced GAN generalization.
- We find that BN in the discriminator tends to cause gradient explosion due to its centering and scaling steps.
- We design CHAIN by replacing BN's centering and scaling steps with 0-mean regularization and adaptive RMS normalization. We show that CHAIN improves stability and generalization of GANs by reducing feature and weight gradients.

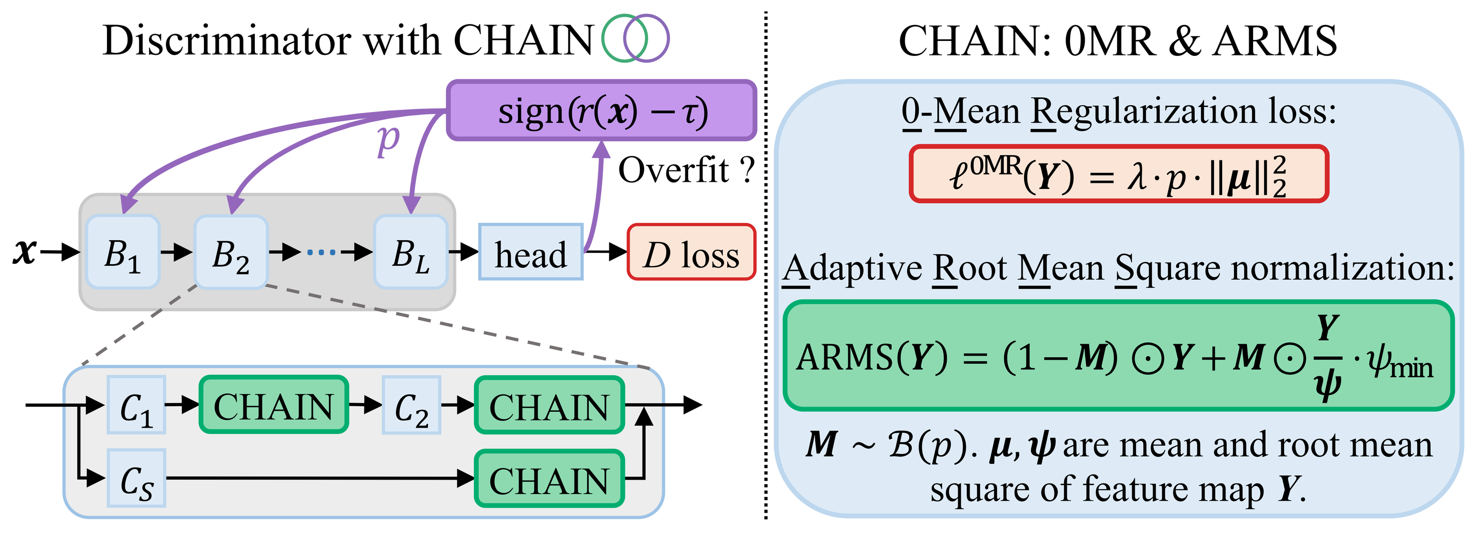
Pipeline

Motivation of BN



Pytorch-style pseudo code for $\text{CHAIN}_{\text{batch}}$

```
# Y: B×d×H×W size; lbd:hyperparameter λ
def CHAIN_batch(Y, p, lbd, eps=1e-5):
  reg = Y.mean([0,2,3]).square().sum() * (p * lbd)
  M = (torch.rand(*Y.shape[:2], 1, 1) < p) * 1.0
  psi_s = Y.square().mean([0,2,3], keepdim = True)
  psi = (psi_s + eps).sqrt()
  psi_min = psi.min().detach()
  Y_arms = (1 - M) * Y + M * (Y / psi * psi_min)
  return Y_arms, reg
```



$p \in [0,1]$: Bernoulli probability and $\ell^{0\text{MR}}$ strength, updated with $r(\boldsymbol{x}) = \mathbb{E}[\text{sign}(D(\boldsymbol{x}))]$, $p_{t+1} = p_t + \Delta_p \cdot \text{sign}(r(\boldsymbol{x}) - \tau)$.

CHAIN is applied **separately** to real and fake data batches.

$\boldsymbol{x}$: Real image. B_l: l-th block. C_S: Conv. in skip branch. D: Discriminator. τ: A predefined threshold. Δ_p: A small value. $\mathcal{B}$: Bernoulli noise. λ: Hyperparameter.

Method

Lowering real/fake discrepancy aids generalization:

GAN generalization error: $\epsilon_{\text{gan}} \leqslant 2d_{\mathcal{H}}(\mu, \hat{\mu}_n) + 2d_{\mathcal{H}}(\nu_n^*, \hat{\nu}_n)$

$d_{\mathcal{H}}$: discrepancy of D. $\mu, \hat{\mu}_n$: unseen/seen real data. $\nu_n^*, \hat{\nu}_n$: ideal/seen fake. $\nu_n^* \approx \hat{\mu}_n$ implies lowering real/fake gap reduces ϵ_{gan}. μ inaccessible, thus we derive $\epsilon_{\text{gan}}^{\text{nn}}$.

Lowering weight gradient norms of D aids generalization:

$$\epsilon_{\text{gan}}^{\text{nn}} \leqslant 2\omega\big(\|\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}\|_2 + \|\widetilde{\boldsymbol{\nabla}}_{\boldsymbol{\theta}_d}\|_2\big) + 4R\Big(\frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}, \frac{1}{n}\Big) + \omega^2\big(|\lambda_{\max}^{\boldsymbol{H}}| + |\lambda_{\max}^{\widetilde{\boldsymbol{H}}}|\big)$$

$\epsilon_{\text{gan}}^{\text{nn}}$: ϵ_{gan} on neural networks, $\omega > 0$. $\boldsymbol{\theta}_d$: D's weights. $\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}, \lambda_{\max}^{\boldsymbol{H}}$: real gradient, top Hessian eigenvalue. $\widetilde{\boldsymbol{\nabla}}_{\boldsymbol{\theta}_d}, \lambda_{\max}^{\widetilde{\boldsymbol{H}}}$: fake versions. R: related to $\|\boldsymbol{\theta}_d\|_2^2$ and data size n.

Separate BN reduces discrepancy but enlarges gradient:

transform: $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{W}$ centering: $\overset{c}{\boldsymbol{Y}} = \boldsymbol{Y} - \boldsymbol{\mu}$ scaling: $\overset{s}{\boldsymbol{Y}} = \overset{c}{\boldsymbol{Y}}/\boldsymbol{\sigma}$

(Centering) similarity dropping causes feature divergence:

$$\mathbb{E}_{\boldsymbol{y}_1, \boldsymbol{y}_2}\big[\cos(\boldsymbol{y}_1, \boldsymbol{y}_2)\big] \geqslant \mathbb{E}_{\overset{c}{\boldsymbol{y}}_1, \overset{c}{\boldsymbol{y}}_2}\big[\cos(\overset{c}{\boldsymbol{y}}_1, \overset{c}{\boldsymbol{y}}_2)\big] = 0$$

$\boldsymbol{y}_1, \overset{c}{\boldsymbol{y}}_1$: pre- & post-centering features. Features similar in early layers diverge in later layers.

(Scaling) unbounded Lipschitz causes gradient explosion:

$$\|\text{diag}(1/\boldsymbol{\sigma})\|_{\text{lc}} = 1/\sigma_{\min}$$

Lipschitz constant (lc) is large when $\sigma_{\min} = \min_c \sigma_c$, is small.

CHAIN replaces centering/scaling with 0MR/ARMS:

mean $\boldsymbol{\mu}$: $\mu_c = \frac{1}{B \times H \times W} \sum_{b,h,w} Y_{b,c,h,w}$

root mean square $\boldsymbol{\psi}$: $\psi_c = \sqrt{\Big(\frac{1}{B \times H \times W} \sum_{b,h,w} Y_{b,c,h,w}^2\Big) + \epsilon}$

0-mean regularization: $\ell^{0\text{MR}}(\boldsymbol{Y}) = \lambda \cdot p \cdot \|\boldsymbol{\mu}\|_2^2$

Adaptive root mean square normalization:

$$\text{ARMS}(\boldsymbol{Y}) = (1 - \boldsymbol{M}) \odot \boldsymbol{Y} + \boldsymbol{M} \odot \frac{\boldsymbol{Y}}{\boldsymbol{\psi}} \cdot \psi_{\min}, \quad \psi_{\min} = \min_c \psi_c$$

$\epsilon = 10^{-5}$. λ: a hyperparameter. p controls $\ell^{0\text{MR}}$ and Bernoulli mask $\boldsymbol{M} \sim \mathcal{B}(p)$

CHAIN reduces gradients of features and weights in D:

$$\|\Delta\boldsymbol{y}_c\|_2^2 \leqslant \|\Delta\dot{\boldsymbol{y}}_c\|_2^2 \Big(\frac{(1-p)\psi_c + p\psi_{\min}}{\psi_c}\Big)^2 - \frac{2(1-p)p\psi_{\min}}{B\psi_c}(\Delta\dot{\boldsymbol{y}}_c^T \check{\boldsymbol{y}}_c)^2$$

$$\|\Delta\boldsymbol{w}_c\|_2^2 \leqslant \lambda_{\max}^2 \|\Delta\boldsymbol{y}_c\|_2^2$$

$\Delta\boldsymbol{y}_c, \Delta\dot{\boldsymbol{y}}_c$: c-th column of gradient for CHAIN input/output $\boldsymbol{Y}, \dot{\boldsymbol{Y}}$. $\lambda_{\max}$: top eigenvalue of $\boldsymbol{A}$. $\check{\boldsymbol{y}}_c$: c-th column of $\check{\boldsymbol{Y}} = \boldsymbol{Y}/\boldsymbol{\psi}$. $\Delta\boldsymbol{w}_c$: c-th column of grad for $\boldsymbol{W}$.

Experimental Results

Comparison with the state-of-the-art methods:

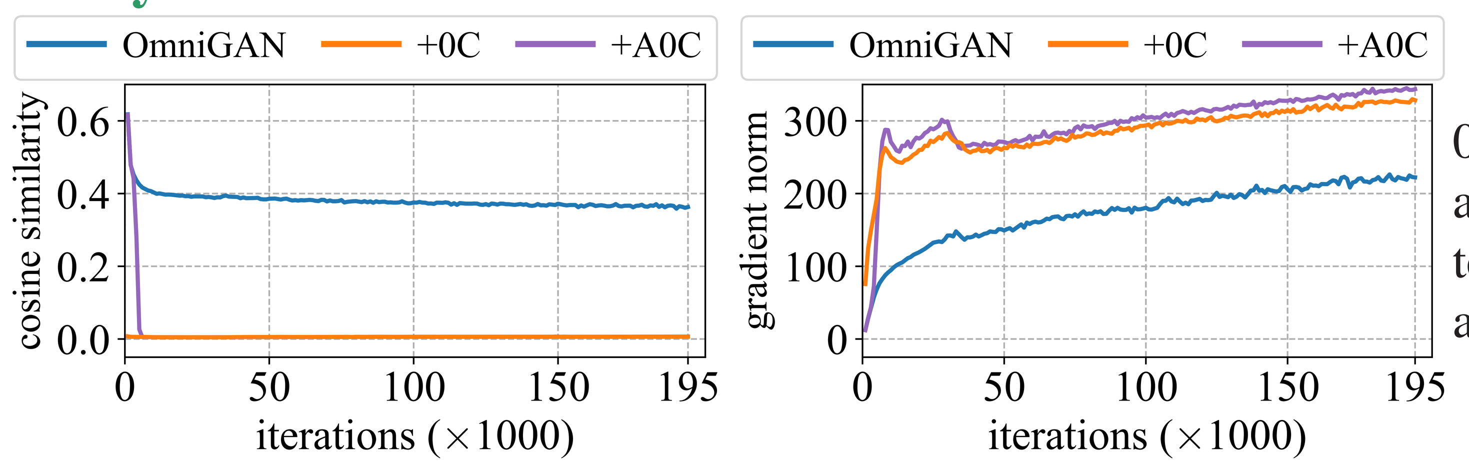
Method	CIFAR-10						CIFAR-100					
	10% data		20% data		100% data		10% data		20% data		100% data	
	IS↑	tFID↓	IS↑	tFID↓	IS↑	tFID↓	IS↑	tFID↓	IS↑	tFID↓	IS↑	tFID↓
BigGAN	8.24	31.45	8.74	16.20	9.21	5.48	7.58	50.79	9.94	25.83	11.02	7.86
+CHAIN	**8.63**	**12.02**	**8.98**	**8.15**	**9.49**	**4.18**	**10.04**	**13.13**	**10.15**	**11.58**	**11.16**	**6.04**
LeCam+DA	8.81	12.64	9.01	8.53	9.45	4.32	9.17	22.75	10.12	15.96	11.25	6.45
+CHAIN	**8.96**	**8.54**	**9.27**	**5.92**	**9.52**	**3.51**	**10.11**	**12.69**	**10.62**	**9.02**	**11.37**	**5.26**
OmniGAN+ADA	7.86	40.05	9.41	27.04	10.24	4.95	8.95	44.65	12.07	13.54	13.07	6.12
+CHAIN	**10.10**	**6.22**	**10.26**	**3.98**	**10.31**	**2.22**	**12.70**	**9.49**	**12.98**	**7.02**	**13.98**	**4.02**

Method	2.5% ImageNet			5% ImageNet			10% ImageNet		
	IS↑	tFID↓	vFID↓	IS↑	tFID↓	vFID↓	IS↑	tFID↓	vFID↓
BigGAN	8.61	101.62	100.09	6.27	90.32	88.01	12.44	50.75	49.84
+CHAIN	**14.68**	**30.66**	**29.32**	**17.34**	**21.13**	**19.95**	**20.45**	**14.70**	**13.84**
ADA	7.93	67.84	66.55	11.56	47.56	46.25	14.82	31.75	30.68
+CHAIN	**16.57**	**23.01**	**21.90**	**19.15**	**16.14**	**15.17**	**22.04**	**12.91**	**12.17**

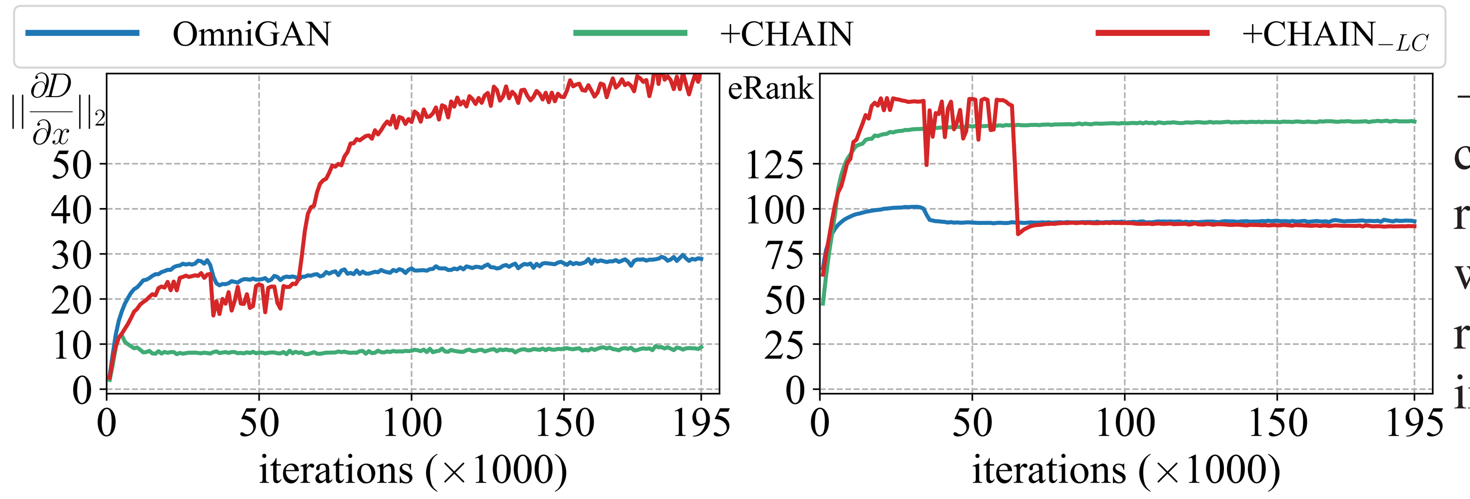
Method (FID↓)	Shells	Skulls	AnimeFace	BreCaHAD	MessidorSet1	Pokemon	ArtPainting
	64 imgs	97 imgs	120 imgs	162 imgs	400 imgs	833 imgs	1000 imgs
FastGAN	138.50	97.87	54.05	63.83	38.33	45.70	43.21
FreGAN	123.75	84.58	49.09	**57.87**	34.61	39.09	43.14
FastGAN$-D_{\text{big}}$	171.35	165.64	76.02	68.63	37.38	53.48	43.04
+CHAIN	**78.62**	**82.47**	**46.27**	58.98	**28.76**	**31.94**	**38.83**

Method (FID↓)	100-shot			Animal Face	
	Obama	GrumpyCat	Panda	Cat	Dog
StyleGAN2	80.20	48.90	34.27	71.71	131.90
+CHAIN	**28.72**	**27.21**	**9.51**	**38.93**	**53.27**
AdvAug	52.86	31.02	14.75	47.40	68.28
DA	46.87	27.08	12.06	42.44	58.85
InsGen	32.42	22.01	9.85	33.01	44.93
FakeCLR	26.95	19.56	8.42	26.34	42.02
KDDLGAN	29.38	19.65	8.41	31.89	50.22
AugSelfGAN	26.00	19.81	8.36	30.53	48.19
DA+CHAIN	**22.87**	**17.57**	**6.93**	**19.58**	**30.88**

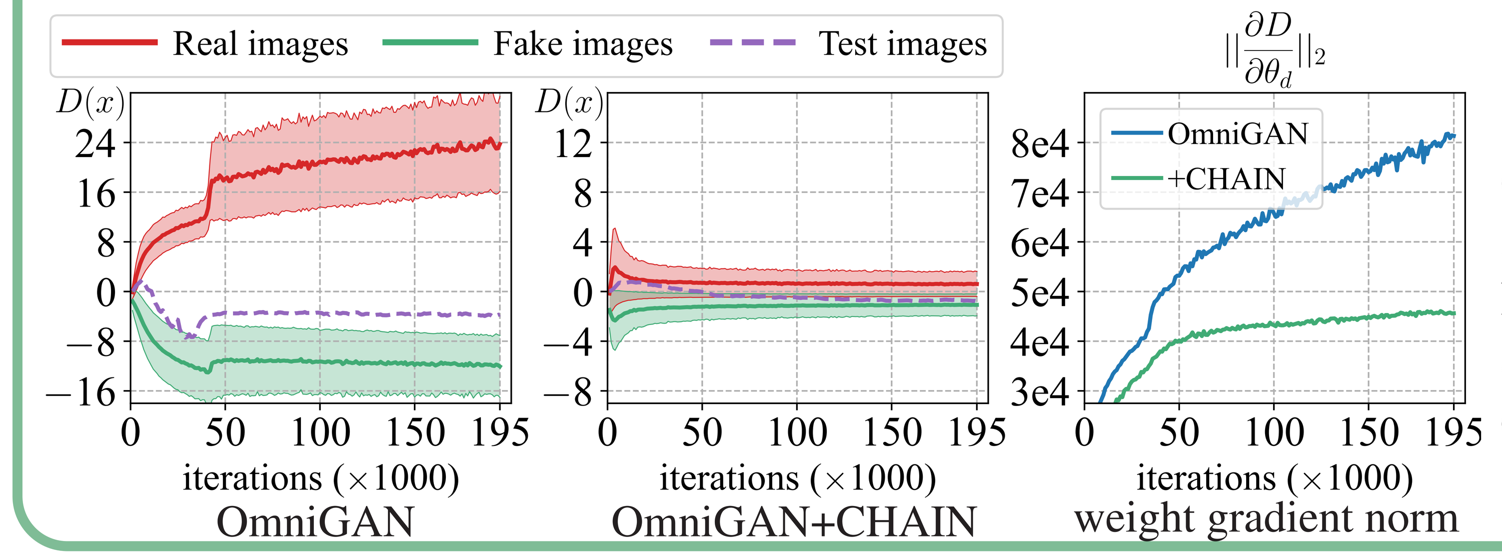
Analyses on 10% CIFAR-10:



0C: centering. A0C: adaptive centering. Centering reduces similarity and raises gradient.



—LC: without Lipschitz constraint. CHAIN reduces latent gradients while removing LC raises gradient and impairs feature eRank.



$D(x)$: discriminator output. CHAIN reduces D's weight gradient, lowering discrepancies among unseen/seen real data and fake data.

Generated Images: