CHAIN: Enhancing Generalization in Data-Efficient GANs via lips*CH*itz continuity constr*AI*ned *N*ormalization (*Supplementary Material*)

Yao Ni[†] Piotr Koniusz[*,§,†]
[†]The Australian National University [§]Data61♥CSIRO
[†]firstname.lastname@anu.edu.au

The supplementary material contains notations (§A), theoretical proofs (§B), an explanation for stochastic $\boldsymbol{M}$ design (§C), implementation guidelines (§D), extra experimental results (§E), training overhead (§F), and examples of generated images (§G).

A. Notations

Below, we explain the notations used in this work.
Scalars: Represented by lowercase letters (*e.g.*, m, n, p).
Vectors: Bold lowercase letters (*e.g.*, $\boldsymbol{x}$, $\boldsymbol{z}$, $\boldsymbol{\mu}$).
Matrices: Bold uppercase letters (*e.g.*, $\boldsymbol{W}$, $\boldsymbol{M}$, $\boldsymbol{H}$).
Functions: Letters followed by brackets (*e.g.*, $\phi(\cdot)$, $h(\cdot)$, $\text{diag}(\cdot)$).
Function sets: Calligraphic uppercase letters are used (*e.g.*, $\mathcal{H}$, $\mathcal{G}$, $\mathcal{F}$). But note $\mathcal{B}$ specifically denotes the Bernoulli distribution.
Probability measures: Denoted by letters μ, ν, π, $\hat{\rho}$ and p_z.
Expectation: $\mathbb{E}[\cdot]$ represents the average or expected value of a random variable.

B. Proofs

We start with a lemma on the Pac-Bayesian bound, followed by in-depth proofs for the theories outlined in the main paper.

Lemma B.1 *(A variant of the PAC-Bayesian bound adapted from Theorem 4.1 in [2] and from [10].) Let $\mathcal{D}$ be a distribution over $\mathcal{X}$. Denote the prior and posterior probability measure on a hypothesis set $\mathcal{F}$ as $\pi(\cdot), \hat{\rho}(\cdot) \in \mathcal{M}^1_+$, where $\mathcal{M}^1_+$ (positive and normalized to 1) is the set of all probability measures on $\mathcal{F}$. Denote $\phi : \mathcal{F} \times \mathcal{X} \to \mathbb{R}$ and $\mathcal{L}^{\phi}_{\mathcal{D}} := \mathbb{E}_{\mathcal{D}}[\phi]$ as the loss. For $\alpha > 0$ and $\delta \in (0, 1]$, with probability at least $1 - \delta$ over the choice of $\boldsymbol{x} \sim \mathcal{D}^n$ (a subset from $\mathcal{D}$ with size of n), we have:*

$$\mathbb{E}_{f\sim\hat{\rho}}\mathcal{L}^{\phi}_{\mathcal{D}}(f) \leqslant \mathbb{E}_{f\sim\hat{\rho}}\widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f) + \frac{1}{\alpha}\Big[KL(\hat{\rho}\|\pi) + \ln\frac{1}{\delta} + \Omega(\alpha, n)\Big]$$
$$\textit{where} \qquad \Omega(\alpha, n) = \ln \mathbb{E}_{f\sim\pi}\mathbb{E}_{\mathcal{D}^n} \exp\big\{\alpha\big(\mathcal{L}^{\phi}_{\mathcal{D}}(f) - \widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f)\big)\big\}. \qquad (17)$$

Proof: The Donsker-Varadhan change of measure states that for any measurable function $\varphi : \mathcal{F} \to \mathbb{R}$ and $\forall\hat{\rho}$ on $\mathcal{F}$, we have:

$$\mathbb{E}_{f\sim\hat{\rho}}\varphi(f) \leqslant \text{KL}(\hat{\rho}\|\pi) + \ln \mathbb{E}_{f\sim\pi}e^{\varphi(f)}.$$

Denoting $\varphi(f) := \alpha\big(\mathcal{L}^{\phi}_{\mathcal{D}}(f) - \widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f)\big)$, the above inequality yields:

$$\begin{aligned}\alpha\big(\mathbb{E}_{f\sim\hat{\rho}}\mathcal{L}^{\phi}_{\mathcal{D}}(f) - \mathbb{E}_{f\sim\hat{\rho}}\widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f)\big) &= \mathbb{E}_{f\sim\hat{\rho}}\alpha\big(\mathcal{L}^{\phi}_{\mathcal{D}}(f) - \widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f)\big)\\ &\leqslant \text{KL}(\hat{\rho}\|\pi) + \ln \mathbb{E}_{f\sim\pi}e^{\alpha(\mathcal{L}^{\phi}_{\mathcal{D}}(f) - \widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f))}.\end{aligned}$$

Applying Markov's inequality to the random variable $\xi_\pi(X) := \mathbb{E}_{f\sim\pi}e^{\alpha(\mathcal{L}^{\phi}_{\mathcal{D}}(f) - \widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f))}$, we obtain:

$$\Pr\Big(\xi_\pi \leqslant \frac{1}{\delta}\mathbb{E}[\xi_\pi]\Big) \geqslant 1 - \delta.$$

Thus, with probability at least $1 - \delta$ over the choice of $\boldsymbol{x} \sim \mathcal{D}^n$, we obtain:

$$\mathbb{E}_{f\sim\hat{\rho}}\mathcal{L}^{\phi}_{\mathcal{D}}(f) \leqslant \mathbb{E}_{f\sim\hat{\rho}}\widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f) + \frac{1}{\alpha}\Big[\text{KL}(\hat{\rho}\|\pi) + \ln\frac{1}{\delta} + \ln \mathbb{E}_{f\sim\pi}\mathbb{E}_{\mathcal{D}^n}e^{\alpha(\mathcal{L}^{\phi}_{\mathcal{D}}(f) - \widehat{\mathcal{L}}^{\phi}_{\mathcal{D}^n}(f))}\Big].$$

B.1. Proof of Lemma 3.1

Lemma 3.1 *(Partial results of Theorem 1 in [32].) Assume the discriminator set $\mathcal{H}$ is even,* i.e., *$h \in \mathcal{H}$ implies $-h \in \mathcal{H}$ and $\|h\|_\infty \leqslant \Delta$. Let $\hat{\mu}_n$ and $\hat{\nu}_n$ be empirical measures of μ and ν_n with size n. Denote $\nu_n^* = \inf_{\nu \in \mathcal{G}} d_{\mathcal{H}}(\hat{\mu}_n, \nu)$. The generalization error of GAN, defined as $\epsilon_{gan} := d_{\mathcal{H}}(\mu, \nu_n) - \inf_{\nu \in \mathcal{G}} d_{\mathcal{H}}(\mu, \nu)$, is bounded as:*

$$\epsilon_{gan} \leqslant 2\big(\sup_{h \in \mathcal{H}} \left|\mathbb{E}_\mu[h] - \mathbb{E}_{\hat{\mu}_n}[h]\right| + \sup_{h \in \mathcal{H}} \left|\mathbb{E}_{\nu_n^*}[h] - \mathbb{E}_{\hat{\nu}_n}[h]\right|\big) = 2d_{\mathcal{H}}(\mu, \hat{\mu}_n) + 2d_{\mathcal{H}}(\nu_n^*, \hat{\nu}_n).$$

Proof:

$$\begin{aligned}\epsilon_{\text{gan}} :=& d_{\mathcal{H}}(\mu, \nu_n) - \inf_{\nu \in \mathcal{G}} d_{\mathcal{H}}(\mu, \nu) = d_{\mathcal{H}}(\mu, \nu_n) - d_{\mathcal{H}}(\hat{\mu}_n, \nu_n) + d_{\mathcal{H}}(\hat{\mu}_n, \nu_n) - \inf_{\nu \in \mathcal{G}} d_{\mathcal{H}}(\mu, \nu) \\ =& \underbrace{d_{\mathcal{H}}(\mu, \nu_n) - d_{\mathcal{H}}(\hat{\mu}_n, \nu_n)}_{\textcircled{1}} + \underbrace{\inf_{v \in \mathcal{G}} d_{\mathcal{H}}(\hat{\mu}_n, \nu) - \inf_{v \in \mathcal{G}} d_{\mathcal{H}}(\mu, \nu)}_{\textcircled{2}} + \underbrace{d_{\mathcal{H}}(\hat{\mu}_n, \nu_n) - \inf_{v \in \mathcal{G}} d_{\mathcal{H}}(\hat{\mu}_n, \nu)}_{\textcircled{3}}.\end{aligned}$$

The three components in the above equation are upper-bounded as follows:

Upper bound ① :

$$\begin{aligned}&d_{\mathcal{H}}(\mu, \nu_n) - d_{\mathcal{H}}(\hat{\mu}_n, \nu_n) = \sup_{h \in \mathcal{H}} |\mathbb{E}_\mu[h] - \mathbb{E}_{\nu_n}[h]| - \sup_{h \in \mathcal{H}} |\mathbb{E}_{\hat{\mu}_n}[h] - \mathbb{E}_{\nu_n}[h]| \\ \leqslant& \sup_{h \in \mathcal{H}} \left|\mathbb{E}_\mu[h] - \mathbb{E}_{\nu_n}[h] - \mathbb{E}_{\hat{\mu}_n}[h] + \mathbb{E}_{\nu_n}[h]\right| = \sup_{h \in \mathcal{H}} \left|\mathbb{E}_\mu[h] - \mathbb{E}_{\hat{\mu}_n}[h]\right|.\end{aligned}$$

Upper bound ② : Denote $\nu^* = \inf_{\nu \in \mathcal{G}} d_{\mathcal{H}}(\mu, \nu)$. Then similar to derivation for ① , we obtain:

$$\begin{aligned}&\inf_{v \in \mathcal{G}} d_{\mathcal{H}}(\hat{\mu}_n, \nu) - \inf_{v \in \mathcal{G}} d_{\mathcal{H}}(\mu, \nu) = \inf_{v \in \mathcal{G}} d_{\mathcal{H}}(\hat{\mu}_n, \nu) - d_{\mathcal{H}}(\mu, \nu^*) \\ \leqslant& d_{\mathcal{H}}(\hat{\mu}_n, \nu) - d_{\mathcal{H}}(\mu, \nu^*) \leqslant \sup_{h \in \mathcal{H}} \left|\mathbb{E}_\mu[h] - \mathbb{E}_{\hat{\mu}_n}[h]\right|.\end{aligned}$$

Upper bound ③ : Here, we consider a practical scenario where the discriminator only has access to finite fake data during optimization. Recall that we denote $\nu_n^* := \inf_{\nu \in \mathcal{G}} d_{\mathcal{H}}(\hat{\mu}_n, \nu)$, thus $d_{\mathcal{H}}(\hat{\mu}_n, \nu_n) \geqslant d_{\mathcal{H}}(\hat{\mu}_n, \nu_n^*)$, leading to the inequality that:

$$\begin{aligned}&d_{\mathcal{H}}(\hat{\mu}_n, \nu_n) - \inf_{v \in \mathcal{G}} d_{\mathcal{H}}(\hat{\mu}_n, \nu) = d_{\mathcal{H}}(\hat{\mu}_n, \nu_n) - d_{\mathcal{H}}(\hat{\mu}_n, \nu_n^*) \\ =& \big(d_{\mathcal{H}}(\hat{\mu}_n, \nu_n) - d_{\mathcal{H}}(\hat{\mu}_n, \hat{\nu}_n)\big) + \big(d_{\mathcal{H}}(\hat{\mu}_n, \hat{\nu}_n) - d_{\mathcal{H}}(\hat{\mu}_n, \nu_n^*)\big) \\ \leqslant& \sup_{h \in \mathcal{H}} \left|\mathbb{E}_{\nu_n}[h] - \mathbb{E}_{\hat{\nu}_n}[h]\right| + \sup_{h \in \mathcal{H}} \left|\mathbb{E}_{\nu_n^*}[h] - \mathbb{E}_{\hat{\nu}_n}[h]\right| \leqslant 2 \sup_{h \in \mathcal{H}} \left|\mathbb{E}_{\nu_n^*}[h] - \mathbb{E}_{\hat{\nu}_n}[h]\right|.\end{aligned}$$

Integrating the three bounds we achieve the final result.

B.2. Proof of Proposition 3.1

Proposition 3.1 *Utilizing notations from Lemma 3.1, we define ϵ_{gan}^{nn} as the generalization error of GAN parameterized as neural network classes. Let $\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}$ and $\boldsymbol{H}_{\boldsymbol{\theta}_d}$ represent the gradient and Hessian matrix of discriminator h evaluated at $\boldsymbol{\theta}_d$ over real training data $\hat{\mu}_n$, and $\widetilde{\boldsymbol{\nabla}}_{\boldsymbol{\theta}_d}$ and $\widetilde{\boldsymbol{H}}_{\boldsymbol{\theta}_d}$ over observed fake data $\hat{\nu}_n$. Denoting $\lambda_{max}^{\boldsymbol{H}}$ and $\lambda_{max}^{\widetilde{\boldsymbol{H}}}$ as the largest eigenvalues of $\boldsymbol{H}_{\boldsymbol{\theta}_d}$ and $\widetilde{\boldsymbol{H}}_{\boldsymbol{\theta}_d}$, respectively, and for any $\omega > 0$, the generalization error is bounded as:*

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

where $R\left(\frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}, \frac{1}{n}\right)$, a term related to discriminator weights norm, is inversely related to the data size n.

Proof: We start by deriving a PAC-Bayesian bound for GAN generalization error on real data. This is followed by an approach similar to Theorem 1 in [21], establishing a connection between this error and the discriminator's gradient direction. Finally, a Taylor expansion of the discriminator in the gradient direction is applied, paralleled by a similar formulation for fake data, culminating in our final results.

PAC-Bayesian bound for GAN. Denoting $\mathcal{L}_\mu := \mathbb{E}_\mu[h]$ and the parameter of the discriminator as $\boldsymbol{\theta}_d \in \boldsymbol{\Theta}_d$, and applying Lemma B.1, we obtain:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\theta}_d \sim \hat{\rho}} \mathcal{L}_\mu(\boldsymbol{\theta}_d) \leqslant \mathbb{E}_{\boldsymbol{\theta}_d \sim \hat{\rho}} \widehat{\mathcal{L}}_{\hat{\mu}_n}(\boldsymbol{\theta}_d) + \frac{1}{\alpha}\left[\mathrm{KL}(\hat{\rho}\|\pi) + \ln\frac{1}{\delta} + \Omega(\alpha, n)\right] \\ &\text{where} \qquad \Omega(\alpha, n) = \ln \mathbb{E}_{\boldsymbol{\theta}_d \sim \pi} \mathbb{E}_{\hat{\mu}_n} \exp\left\{\alpha\left(\mathcal{L}_\mu(\boldsymbol{\theta}_d) - \widehat{\mathcal{L}}_{\hat{\mu}_n}(\boldsymbol{\theta}_d)\right)\right\}.\end{aligned} \tag{18}$$

We then derive the upper bound for $\Omega(\alpha, n)$ on the discriminator. Let ℓ_i represent a realization of the random variable $\mathcal{L}_\mu - h(\boldsymbol{x}_i; \boldsymbol{\theta}_d)$. Given that $h \in [-\Delta, \Delta]$ stated in Lemma 3.1, changing variable $\boldsymbol{x}_i$ to another independent copy $\boldsymbol{x}_i'$, alters

ℓ_i by at most $\frac{2\Delta}{n}$. Utilizing Hoeffding's lemma, we obtain:

$$\begin{aligned}\mathbb{E}_{\hat{\mu}_n} e^{\alpha(\mathcal{L}_\mu(\boldsymbol{\theta}_d)-\widehat{\mathcal{L}}_{\hat{\mu}_n}(\boldsymbol{\theta}_d))} &= \mathbb{E}_{\hat{\mu}_n} \exp\Big\{\frac{\alpha}{n}\sum_{i=1}^{n}\ell_i\Big\} = \prod_{i=1}^{n}\mathbb{E}\exp\Big\{\frac{\alpha}{n}\ell_i\Big\} \\ &\leqslant \prod_{i=1}^{n}\exp\Big\{\frac{\alpha^2(2\Delta)^2}{8n^2}\Big\} = \exp\Big\{\frac{\alpha^2\Delta^2}{2n}\Big\}. \end{aligned} \tag{19}$$

By inserting Eq. 19 into Eq. 18, and setting $\alpha = n$, we arrive at:

$$\mathbb{E}_{\boldsymbol{\theta}_d\sim\hat{\rho}}\mathcal{L}_\mu(\boldsymbol{\theta}_d) \leqslant \mathbb{E}_{\boldsymbol{\theta}_d\sim\hat{\rho}}\widehat{\mathcal{L}}_{\hat{\mu}_n}(\boldsymbol{\theta}_d) + \frac{1}{n}\Big[\mathrm{KL}(\hat{\rho}\|\pi) + \ln\frac{1}{\delta}\Big] + \frac{\Delta^2}{2}. \tag{20}$$

Generalization error and the gradient direction of the weight. Continuing, we adopt an analysis parallel to the proof of Theorem 1 in [21]. According to Eq. 12 in their work, if π is a measure on $\mathcal{N}(\mathbf{0}, \sigma_\pi^2\boldsymbol{I})$ and $\hat{\rho}$ is a measure on $\mathcal{N}(\boldsymbol{\theta}_d, \sigma^2\boldsymbol{I})$ with the dimension of $\boldsymbol{\theta}$ being k, it follows that:

$$\mathrm{KL}(\hat{\rho}\|\pi) = \frac{1}{2}\Big[1 + k\ln\big(1 + \frac{\|\boldsymbol{\theta}_d\|_2^2}{k\sigma^2}\big)\Big].$$

Subsequently, Eq. 20 transforms into:

$$\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}\mathcal{L}_\mu(\boldsymbol{\theta}_d+\boldsymbol{\varepsilon}) \leqslant \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}\widehat{L}_{\hat{\mu}_n}(\boldsymbol{\theta}_d+\boldsymbol{\varepsilon}) + \frac{1}{n}\Big[\frac{1}{2} + \frac{k}{2}\ln\big(1 + \frac{\|\boldsymbol{\theta}_d\|_2^2}{k\sigma^2}\big) + \ln\frac{1}{\delta}\Big] + \frac{\Delta^2}{2}. \tag{21}$$

By Lemma 1 of [51], for any positive t, we have:

$$\Pr(\|\boldsymbol{\varepsilon}\|_2^2 - k\sigma^2 \geqslant 2\sigma^2\sqrt{kt} + 2t\sigma^2) \leqslant e^{-t}.$$

Thus, with probability $1 - 1/n$ (where $t = \ln n$), it follows that:

$$\|\boldsymbol{\varepsilon}\|_2^2 \leqslant \sigma^2\big(2\ln n + k + 2\sqrt{k\ln n}\big) \leqslant \sigma^2 k\Bigg(1 + \sqrt{\frac{2\ln n}{k}}\Bigg)^2 \leqslant \omega^2.$$

Assuming, as in [21], that perturbations in discriminator weights have negligible impact on performance over an infinite dataset, and integrating σ back into Eq. 21, we deduce:

$$\begin{aligned}\mathcal{L}_\mu(\boldsymbol{\theta}_d) &\leqslant \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}\widehat{\mathcal{L}}_{\hat{\mu}_n}(\boldsymbol{\theta}_d+\boldsymbol{\varepsilon}) + \frac{1}{n}\Big[\frac{1}{2} + \frac{k}{2}\ln\big(1 + \frac{\|\boldsymbol{\theta}_d\|_2^2}{k\sigma^2}\big) + \ln\frac{1}{\delta}\Big] + \frac{\Delta^2}{2} \\ &\leqslant \max_{\|\boldsymbol{\varepsilon}\|_2^2\leqslant\omega^2}\widehat{\mathcal{L}}_{\hat{\mu}_n}(\boldsymbol{\theta}_d+\boldsymbol{\varepsilon}) + \frac{1}{n}\Bigg[\frac{1}{2} + \frac{k}{2}\ln\Bigg(1 + \frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}\Big(1 + \sqrt{(2\ln n)/k}\Big)^2\Bigg) + \ln\frac{1}{\delta}\Bigg] + \frac{\Delta^2}{2}.\end{aligned}$$

Taylor expansion in the weight gradient direction. Observe that the maximum of $\widehat{\mathcal{L}}_{\hat{\mu}_n}$ occurs when $\boldsymbol{\varepsilon}$ is chosen as $\boldsymbol{\varepsilon} = \frac{\omega\boldsymbol{\nabla}_{\hat{\mu}_n,\boldsymbol{\theta}_d}}{\|\boldsymbol{\nabla}_{\hat{\mu}_n,\boldsymbol{\theta}_d}\|_2}$, which is aligned with the gradient of $\widehat{\mathcal{L}}_{\hat{\mu}_n}$ at $\boldsymbol{\theta}_d$ over $\hat{\mu}_n$. We perform a second-order Taylor expansion of $\widehat{\mathcal{L}}_{\hat{\mu}_n}$ around $\boldsymbol{\theta}_d$. Incorporating the remainder and the higher-order terms from the Taylor expansion into $R\Big(\frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}, \frac{1}{n}\Big)$, we derive:

$$\begin{aligned}L_\mu(\boldsymbol{\theta}_d) &\leqslant \widehat{L}_{\hat{\mu}_n}\Big(\boldsymbol{\theta}_d + \frac{\omega\boldsymbol{\nabla}_{\hat{\mu}_n,\boldsymbol{\theta}_d}}{\|\boldsymbol{\nabla}_{\hat{\mu}_n,\boldsymbol{\theta}_d}\|_2}\Big) + R\Big(\frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}, \frac{1}{n}\Big) \\ &\approx \widehat{L}_{\hat{\mu}_n}(\boldsymbol{\theta}_d) + \omega\|\boldsymbol{\nabla}_{\hat{\mu}_n,\boldsymbol{\theta}_d}\|_2 + \frac{\omega^2}{2\|\boldsymbol{\nabla}_{\hat{\mu}_n,\boldsymbol{\theta}_d}\|_2^2}\boldsymbol{\nabla}_{\hat{\mu}_n,\boldsymbol{\theta}_d}^T\boldsymbol{H}_{\hat{\mu}_n,\boldsymbol{\theta}_d}\boldsymbol{\nabla}_{\hat{\mu}_n,\boldsymbol{\theta}_d} + R\Big(\frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}, \frac{1}{n}\Big).\end{aligned}$$

Simplifying notations, we use $\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}$ and $\boldsymbol{H}_{\boldsymbol{\theta}_d}$ for the gradient and Hessian matrix evaluated at $\boldsymbol{\theta}_d$ over real seen data $\hat{\mu}_n$, and similar $\widetilde{\boldsymbol{\nabla}}_{\boldsymbol{\theta}_d}$ and $\widetilde{\boldsymbol{H}}_{\boldsymbol{\theta}_d}$ for observed fake data $\hat{\nu}_n$. Considering the largest eigenvalue of $\boldsymbol{H}_{\boldsymbol{\theta}_d}$ as $\lambda_{\max}^{\boldsymbol{H}}$, implying $\boldsymbol{v}^T\boldsymbol{H}_{\boldsymbol{\theta}_d}\boldsymbol{v} \leqslant \lambda_{\max}^{\boldsymbol{H}}\|\boldsymbol{v}\|_2^2$, we bound the real data part of the generalization error of a GAN (Lemma 3.1) parameterized as network as follows:

$$\begin{aligned}\sup_{h\in\mathcal{H}_{\text{nn}}}\big|\mathbb{E}_\mu[h] - \mathbb{E}_{\hat{\mu}_n}[h]\big| &\leqslant \omega\|\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}\|_2 + \frac{\omega^2}{2\|\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}\|_2^2}\Big|\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}^T\boldsymbol{H}_{\boldsymbol{\theta}_d}\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}\Big| + R\big(\frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}, \frac{1}{n}\big) \\ &\leqslant \omega\|\boldsymbol{\nabla}_{\boldsymbol{\theta}_d}\|_2 + \frac{\omega^2}{2}|\lambda_{\max}^{\boldsymbol{H}}| + R\big(\frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}, \frac{1}{n}\big).\end{aligned}$$

Similarly, the fake data part in the generalization error of GAN is:

$$\sup_{h\in\mathcal{H}_{\text{nn}}}\big|\mathbb{E}_{\nu_n^*}[h] - \mathbb{E}_{\hat{\nu}_n}[h]\big| \leqslant \omega\|\widetilde{\boldsymbol{\nabla}}_{\boldsymbol{\theta}_d}\|_2 + \frac{\omega^2}{2}|\lambda_{\max}^{\widetilde{\boldsymbol{H}}}| + R\big(\frac{\|\boldsymbol{\theta}_d\|_2^2}{\omega^2}, \frac{1}{n}\big).$$

By integrating the aforementioned two inequalities into the generalization error as detailed in Lemma 3.1, we arrive at:

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

B.3. Proof of Theorem 3.1

Theorem 3.1 *(The issue of the centering step.) Consider $\boldsymbol{y}_1, \boldsymbol{y}_2$ as i.i.d. samples from a symmetric distribution centered at $\boldsymbol{\mu}$, where the presence of $\boldsymbol{y}$ implies $2\boldsymbol{\mu} - \boldsymbol{y}$ is also included. After the centering step, $\overset{c}{\boldsymbol{y}}_1, \overset{c}{\boldsymbol{y}}_2$ are i.i.d. samples from the centered distribution. The expected cosine similarity between these samples is given by:*

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

Proof: Given that the distribution is symmetric and even, and $\boldsymbol{\mu}_Y \neq \boldsymbol{0}$, the mean of the L_2 normalized distribution $\mathbb{E}\big[\frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_2}\big] \neq \boldsymbol{0}$. Denoting the mean of the L_2 normalized sample as $\boldsymbol{\mu}_Z \neq \boldsymbol{0}$, we can derive the expectation of the cosine similarity as follows:

$$\mathbb{E}_{\boldsymbol{y}_1,\boldsymbol{y}_2}\big[\cos(\boldsymbol{y}_1, \boldsymbol{y}_2)\big] = \mathbb{E}_{\boldsymbol{y}_1,\boldsymbol{y}_2}\Big[\big\langle\frac{\boldsymbol{y}_1}{\|\boldsymbol{y}_1\|_2}, \frac{\boldsymbol{y}_2}{\|\boldsymbol{y}_2\|_2}\big\rangle\Big] = \mathbb{E}_{\boldsymbol{z}_1,\boldsymbol{z}_2}[\langle\boldsymbol{z}_1, \boldsymbol{z}_2\rangle] = \langle\boldsymbol{\mu}_Z, \boldsymbol{\mu}_Z\rangle = \|\boldsymbol{\mu}_Z\|_2^2 \geqslant 0.$$

In the centered distribution with $\overset{c}{\boldsymbol{\mu}}_Y = \boldsymbol{0}$ and the symmetric probability, the presence of $\overset{c}{\boldsymbol{y}}_2$ implies the inclusion of $-\overset{c}{\boldsymbol{y}}_2$. This leads us to the following derivation:

$$\mathbb{E}_{\overset{c}{\boldsymbol{y}}_1,\overset{c}{\boldsymbol{y}}_2}\big[\cos(\overset{c}{\boldsymbol{y}}_1, \overset{c}{\boldsymbol{y}}_2)\big] = \mathbb{E}_{\overset{c}{\boldsymbol{y}}_1,\overset{c}{\boldsymbol{y}}_2}\Big[\big\langle\frac{\overset{c}{\boldsymbol{y}}_1}{\|\overset{c}{\boldsymbol{y}}_1\|_2}, \frac{\overset{c}{\boldsymbol{y}}_2}{\|\overset{c}{\boldsymbol{y}}_2\|_2}\big\rangle\Big] = \frac{1}{2}\mathbb{E}_{\overset{c}{\boldsymbol{y}}_1,\overset{c}{\boldsymbol{y}}_2}\Big[\big\langle\frac{\overset{c}{\boldsymbol{y}}_1}{\|\overset{c}{\boldsymbol{y}}_1\|_2}, \frac{\overset{c}{\boldsymbol{y}}_2}{\|\overset{c}{\boldsymbol{y}}_2\|_2}\big\rangle + \big\langle\frac{\overset{c}{\boldsymbol{y}}_1}{\|\overset{c}{\boldsymbol{y}}_1\|_2}, \frac{-\overset{c}{\boldsymbol{y}}_2}{\|\overset{c}{\boldsymbol{y}}_2\|_2}\big\rangle\Big] = 0$$

Comparing the above two Equations we obtain the final inequality.

B.4. Proof of Theorem 3.2

Theorem 3.2 *(The issue of the scaling step.) The scaling step, defined in Eq. 7, can be expressed as matrix multiplication $\overset{s}{\boldsymbol{Y}} = \overset{c}{\boldsymbol{Y}} \mathit{diag}(1/\boldsymbol{\sigma})$. The Lipschitz constant w.r.t. the 2-norm of the scaling step is:*

$$\Big\|\mathit{diag}\Big(\frac{1}{\boldsymbol{\sigma}}\Big)\Big\|_{lc} = \frac{1}{\sigma_{min}},$$

where $\sigma_{min} = \min_c \sigma_c$ represents the minimum value in $\boldsymbol{\sigma}$.

Proof: Consider $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_d)$, a diagonal matrix. We establish that:

$$\|\boldsymbol{\Lambda}\|_{\mathrm{lc}} = \max_{\|\boldsymbol{x}\|_2=1} \|\Lambda\boldsymbol{x}\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \Big(\sum_{i=1}^{d} \lambda_i x_i^2\Big)^{1/2} \leqslant \max_{\|\boldsymbol{x}\|_2=1} \max_i |\lambda_i| \Big(\sum_{i=1}^{d} x_i^2\Big)^{1/2} = \max_i |\lambda_i| \cdot \max_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{x}\|_2 = \max_i |\lambda_i|.$$

From this, it follows that:

$$\Big\|\mathrm{diag}(\frac{1}{\boldsymbol{\sigma}})\Big\|_{\mathrm{lc}} = \max_c \Big|\frac{1}{\sigma_c}\Big| = \frac{1}{\min_c \sigma_c} = \frac{1}{\sigma_{\mathrm{min}}}.$$

B.5. Proof of Theorem 3.3

Theorem 3.3 *(CHAIN reduces the gradient norm of weights/latent features.) Denote the loss of discriminator with CHAIN as $\mathcal{L}$, and the resulting batch features as $\dot{\boldsymbol{Y}}$. Let $\breve{\boldsymbol{y}}_c \in \mathbb{R}^B$ be c-th column of $\breve{\boldsymbol{Y}}$, $\Delta\boldsymbol{y}_c, \Delta\dot{\boldsymbol{y}}_c \in \mathbb{R}^B$ be the c-th column of gradient $\frac{\partial\mathcal{L}}{\partial\boldsymbol{Y}}, \frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{Y}}}$. Denote $\Delta\boldsymbol{w}_c$ as the c-th column of weight gradient $\frac{\partial\mathcal{L}}{\partial\boldsymbol{W}}$ and λ_{max} as the largest eigenvalue of pre-layer features $\boldsymbol{A}$. Then we have:*

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

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

Proof: Aligning with Theorem 4.1 from [80] we derive the gradients of the latent feature and the weight. For convenience, we define $\dot{\boldsymbol{Y}}$ as the resulted interpolated batch features from Eq. 14. By applying the expectation over the $\boldsymbol{M}$, replacing it with p, and using the chain rule of the backward propagation, we determine the expected gradient for each $\Delta y_c^{(b)}$ within $\Delta\boldsymbol{y}_c \in \mathbb{R}^B$ as follows:

$$\begin{aligned}
\Delta y_c^{(b)} &= \Delta\dot{y}_c^{(b)}(1-p) + p\frac{\psi_{\mathrm{min}}}{\psi_c}\big(\Delta\dot{y}_c^{(b)} - \breve{y}_c^{(b)} \cdot \frac{1}{B}\sum_i^B(\Delta\dot{y}_c^{(i)} \cdot \breve{y}_c^{(i)})\big) \\
&= \Delta\dot{y}_c^{(b)}\Big(\frac{(1-p)\psi_c + p\psi_{\mathrm{min}}}{\psi_c}\Big) - p\frac{\psi_{\mathrm{min}}}{\psi_c}\breve{y}_c^{(b)}\frac{1}{B}\sum_{i=1}^{B}\Delta\dot{y}_c^{(i)} \cdot \breve{y}_c^{(i)}.
\end{aligned}$$

The squared gradient norm for $\Delta \boldsymbol{y}_c$ is calculated as follows:

$$\begin{aligned}\|\Delta \boldsymbol{y}_c\|_2^2 &= \|\Delta \dot{\boldsymbol{y}}_c\|_2^2 \Big(\frac{(1-p)\psi_c + p\psi_{\text{min}}}{\psi_c}\Big)^2 - \big(\frac{2(1-p)p\psi_{\text{min}}}{B\psi_c} + \frac{p^2\psi_{\text{min}}^2}{B\psi_c^2}\big)(\Delta \dot{\boldsymbol{y}}_c^T \breve{\boldsymbol{y}}_c)^2 \\ &\leqslant \|\Delta \dot{\boldsymbol{y}}_c\|_2^2 \Big(\frac{(1-p)\psi_c + p\psi_{\text{min}}}{\psi_c}\Big)^2 - \frac{2(1-p)p\psi_{\text{min}}}{B\psi_c}(\Delta \dot{\boldsymbol{y}}_c^T \breve{\boldsymbol{y}}_c)^2.\end{aligned}$$

Using the chain rule, we derive the gradient w.r.t. the weight as follows:

$$\frac{\partial \mathcal{L}}{\partial W_{ic}} = \sum_{b=1}^{B} \frac{\partial L}{\partial y_c^{(b)}} \frac{\partial y_c^{(b)}}{\partial W_{ic}} = \Delta \boldsymbol{y}_c^T \boldsymbol{a}_c.$$

This leads to:

$$\Delta \boldsymbol{w}_c = \boldsymbol{A}^T \Delta \boldsymbol{y}_c.$$

Considering λ_{max} as the largest eigenvalues of $\boldsymbol{A}$, which suggests $\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} \leqslant \lambda_{\text{max}} \|\boldsymbol{v}\|_2^2$, we obtain the following result:

$$\|\Delta \boldsymbol{w}_c\|_2^2 = \Delta \boldsymbol{y}_c^T \boldsymbol{A}\boldsymbol{A}^T \Delta \boldsymbol{y}_c \leqslant \lambda_{\text{max}}^2 \|\Delta \boldsymbol{y}_c\|_2^2.$$

C. The decorrelation effect of the stochastic design $\boldsymbol{M}$

To analyze why the stochastic design $\boldsymbol{M}$ outperforms the deterministic p, we examine the correlation coefficient between two random variables Y_i, Y_j from two different channels.

Theorem C.1 *Let Y_i, Y_j be random variables from the i-th and j-th channels, respectively, where $i \neq j$. Define $\widehat{Y}_i = \frac{Y_i}{\psi_i}\psi_{min}$ as the normalized random variable from channel i after root mean square normalization. Considering an adaptive p under our control, we distinguish between the deterministic version of CHAIN,* i.e. *CHAIN$_{\text{Dtm.}}$ and our stochastic CHAIN as:*

$$\textit{Deterministic (CHAIN}_{\textit{Dtm.}}\textit{):} \quad Y_i' = (1-p)Y_i + p\widehat{Y}_i, \tag{22}$$

$$\textit{Stochastic (CHAIN):} \quad \dot{Y}_i = (1-m)Y_i + m\widehat{Y}_i, \; \textit{where } m \sim \mathcal{B}(p). \tag{23}$$

Assuming $\mathbb{E}[Y_i] = \mathbb{E}[Y_j] = 0$, achievable through our zero mean regularization in Eq. 12, and letting $\sigma_i, \sigma_i', \dot{\sigma}_i$ represent the standard deviations of $Y_i, Y_i', \dot{Y}_i$, respectively, we define and relate the correlation coefficients of the two versions as follows:

$$\varrho_{ij}' = \frac{Cov(Y_i', Y_j')}{\sigma_i'\sigma_j'} \quad \geqslant \quad \dot{\varrho}_{ij} = \frac{Cov(\dot{Y}_i, \dot{Y}_j)}{\dot{\sigma}_i\dot{\sigma}_j}. \tag{24}$$

Theorem C.1 reveals that the stochastic CHAIN has a lower correlation coefficient among features from different channels than the deterministic CHAIN$_{\text{Dtm.}}$, indicating that the stochastic design $\boldsymbol{M}$ exhibits a decorrelation effect.

Proof: Given $\mathbb{E}[Y_i] = 0$, it follows that $\mathbb{E}[Y_i'] = \mathbb{E}[\dot{Y}_i] = 0$. Using the covariance definition $Cov(Z_1, Z_2) = \mathbb{E}[(Z_1 - \mu_{Z_1})(Z_2 - \mu_{Z_2})]$ for any two random variables Z_1, Z_2, we get:

$$Cov(Y_i', Y_j') = \mathbb{E}[Y_i'Y_j'], \quad Cov(\dot{Y}_i, \dot{Y}_j) = \mathbb{E}[\dot{Y}_i\dot{Y}_j].$$

Since m is stochastic noise independent of $Y_i, \widehat{Y}_i$, and $m \sim \mathcal{B}(p)$ implying in $\mathbb{E}[m] = p$, we conclude:

$$\mathbb{E}[Y_i'Y_j'] = \mathbb{E}[\dot{Y}_i\dot{Y}_j] \;\rightarrow\; Cov(Y_i', Y_j') = Cov(\dot{Y}_i, \dot{Y}_j). \tag{25}$$

Next, we explore the relationship between the variances $\sigma_i'^2$ and $\dot{\sigma}_i^2$:

$$\sigma_i'^2 = \mathbb{E}[Y_i'^2] - \mathbb{E}[Y_i']^2 = \mathbb{E}\Big[\Big((1-p+\frac{p\psi_{\text{min}}}{\psi_i})Y_i\Big)^2\Big] - 0 = \big(1-p+\frac{p\psi_{\text{min}}}{\psi_i}\big)^2\mathbb{E}[Y_i^2], \tag{26}$$

$$\dot{\sigma}_i^2 = \mathbb{E}[\dot{Y}_i^2] - \mathbb{E}[\dot{Y}_i]^2 = (1-p)\mathbb{E}[Y_i^2] + p\mathbb{E}[\widehat{Y}_i^2] - 0 = (1-p+p\frac{\psi_{\text{min}}^2}{\psi_i^2})\mathbb{E}[Y_i^2]. \tag{27}$$

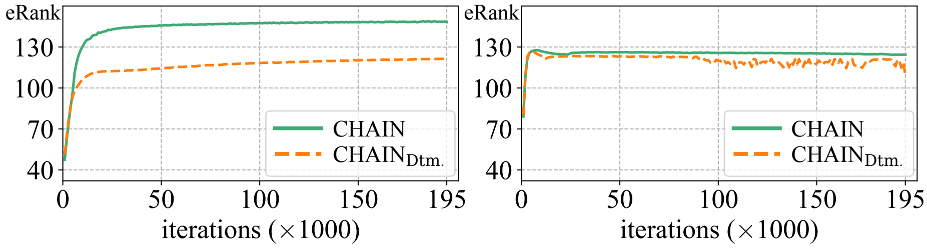
Comparing Eq. 26 and 27, and considering $p \in [0, 1]$, we establish the following relationship:

$$\begin{aligned}&\Big(1-p+p\frac{\psi_{\text{min}}^2}{\psi_i^2}\Big) - \Big(1-p+\frac{p\psi_{\text{min}}}{\psi_i}\Big)^2 = p(1-p) + p(1-p)\frac{\psi_{\text{min}}^2}{\psi_i^2} - 2p(1-p)\frac{\psi_{\text{min}}}{\psi_i} \\ =&p(1-p)\Big(1-\frac{\psi_{\text{min}}}{\psi_i}\Big)^2 \geqslant 0.\end{aligned}$$

Therefore, $\sigma_i' \leqslant \dot{\sigma}_i$, and similarly $\sigma_j' \leqslant \dot{\sigma}_j$. Coupled with Eq. 25, we derive the following conclusion:

$$\begin{cases} Cov(Y_i', Y_j') = Cov(\dot{Y}_i, \dot{Y}_j) \\ \sigma_i'\sigma_j' \leqslant \dot{\sigma}_i\dot{\sigma}_j \end{cases} \rightarrow \varrho_{ij}' = \frac{Cov(Y_i', Y_j')}{\sigma_i'\sigma_j'} \geqslant \dot{\varrho}_{ij} = \frac{Cov(\dot{Y}_i, \dot{Y}_j)}{\dot{\sigma}_i\dot{\sigma}_j}.$$

Experimental validation. Decorrelation diversifies feature patterns, promoting a higher feature rank. This is demonstrated in Figure 8, where CHAIN, employing the stochastic $\boldsymbol{M}$ over the deterministic value p used by CHAIN$_{\text{Dtm.}}$, achieves a higher effective rank (eRank) [78]. This supports Theorem C.1, underscoring the beneficial effect of stochastic design in $\boldsymbol{M}$ for decorrelation, and validates the design choice of CHAIN.



(a) 10% CIFAR-10 with OmniGAN ($d = 256$). (b) 10% CIFAR-100 with BigGAN ($d = 256$).

Figure 8. Effective rank [78] of all pre-activation features in the discriminator for CHAIN and CHAIN$_{\text{Dtm.}}$ on (a) 10% CIFAR-10 using OmniGAN ($d = 256$) and (b) 10% CIFAR-100 with BigGAN ($d = 256$).

D. Implementation Details

In this section, we overcome the mini-batch size limitation of CHAIN$_{\text{batch}}$, which relies solely on current batch data statistics, by developing it to CHAIN, which ultilizes cumulative running forward/backward statistics across training. We also provide detailed implementation for Network and hyper-parameter choices, and methods applied in our ablation studies.

D.1. Implementation of CHAIN (running cumulative forward/backward statistics across training)

Inspired by [30, 85, 107], we enhance CHAIN to use running cumulative forward/backward statistics. We simplify our analysis by focusing on the Root Mean Square Normalization (RMSNorm), considering features of a single channel and omitting the channel index. Additionally, we exclude the constant ϵ, used to avoid division by zero, as it is unnecessary for this analysis. This refinement enables the representation of the forward process for the root mean square normalization as follows:

$$\psi^2 = \frac{1}{B}\sum_{b=1}^{B}(y^{(b)})^2, \tag{28}$$

$$\psi = \sqrt{\psi^2}, \tag{29}$$

$$\breve{y}^{(b)} = \frac{y^{(b)}}{\psi}, \tag{30}$$

$$\widehat{y}^{(b)} = \breve{y}^{(b)} \cdot \psi_{\text{min}}. \tag{31}$$

Leveraging the chain rule, the gradient calculation can be expressed as follows:

$$\frac{\partial \mathcal{L}}{\partial \breve{y}^{(b)}} = \frac{\partial \mathcal{L}}{\partial \widehat{y}^{(b)}} \cdot \psi_{\text{min}}, \tag{32}$$

$$\frac{\partial \mathcal{L}}{\partial y^{(b)}} = \frac{1}{\psi}\Big[\frac{\partial \mathcal{L}}{\partial \breve{y}^{(b)}} - \breve{y}^{(b)} \cdot \Psi\Big], \tag{33}$$

$$\text{where} \quad \Psi = \frac{1}{B}\sum_{i=1}^{B}\frac{\partial \mathcal{L}}{\partial \breve{y}^{(i)}} \cdot \breve{y}^{(i)}. \tag{34}$$

Examining the forward and backward processes reveals that Eq. 28 and 34 are dependent on the batch size. To eliminate this dependency, we propose updating the cumulative statistics for these terms as follows:

$$\overline{\psi^2}_{t+1} = \overline{\psi^2}_{t} \cdot \alpha_{\text{d}} + \psi^2 \cdot (1 - \alpha_{\text{d}}), \tag{35}$$

$$\overline{\Psi}_{t+1} = \overline{\Psi}_{t} \cdot \alpha_{\text{d}} + \Psi \cdot (1 - \alpha_{\text{d}}), \tag{36}$$

where α_{d}, a decay hyperpamameter, is typically set as 0.9. We replace ψ^2, Ψ with their cumulative versions $\overline{\psi^2}, \overline{\Psi}$. This forms an effective algorithm for the normalization part of CHAIN, using cumulative forward/backward statistics, as shown in Alg. 1

Algorithm 1: PyTorch-style pseudo code for Root Mean Square Normalization (RMSNorm) in CHAIN.

```
# Y:BxdxHxW feature, running_psi_sqr:$\overline{\psi^2}$, decay:$\alpha_d$, eps:a small constant
def RMSNorm_forward(Y, running_psi_sqr, decay=0.9, eps=1e-5):
    psi_sqr=Y.square().mean(axis=[0,2,3], keepdim=True) # Eq.28
    running_psi_sqr.data.mul_(decay).add_(psi_sqr, alpha=1-decay) # Eq.35
    running_psi=(running_psi_sqr + eps).sqrt() # Eq.29
    psi_min = running_psi.min().detach()
    Ycheck = Y / running_psi # Eq.30
    return Ycheck * psi_min # Eq.31

# grad_Yhat:BxdxHxW $\frac{\partial \mathcal{L}}{\partial \hat{Y}}$, running_psi:$\overline{\psi}$, running_Psi_grad:$\overline{\Psi}$, psi_min:$\psi_{\min}$ decay:$\alpha_d$
def RMSNorm_backward(grad_Yhat, Ycheck, running_psi, running_Psi_grad, psi_min, decay=0.9):
    grad_Ycheck = grad_Yhat * psi_min # Eq.32
    Psi_grad = (Ycheck * grad_Ycheck).mean(axis=[0,2,3], keepdim=True) # Eq.34
    running_Psi_grad.data.mul_(decay).add_(Psi_grad, alpha=1-decay) # Eq.36
    return (grad_Ycheck - Ycheck * running_Psi_grad) / running_psi # Eq.33
```

D.2. Network and hyper-parameters

CIFAR-10/100. We utilize OmniGAN ($d = 256$ and 1024) and BigGAN ($d = 256$) with a batch size of 32. Following [120], OmniGAN and BigGAN are trained for $1K$ epochs on full data and $5K$ epochs on 10%/20% data setting. CHAIN is integrated into the discriminator, after convolutional layers $c \in \{C_1, C_2, C_S\}$ at all blocks $l \in \{1, 2, 3, 4\}$, with hyperparameters set as $\Delta_p = 0.001, \tau = 0.5, \lambda = 20$.

ImageNet. We build CHAIN upon BigGAN with 512 batch size. We adopt learning rate of 1e-4 for generator and 2e-4 for discriminator. CHAIN is applied after convolutional layers $c \in \{C_1, C_2, C_S\}$ at all blocks $l \in \{1, 2, 3, 4, 5\}$, with hyperparameters $\Delta_p = 0.001, \tau = 0.5, \lambda = 20$.

5 Low-shot images (256 × 256). We build CHAIN upon StyleGAN2 with a batch size of 64, training until the discriminator has seen 25M real images. CHAIN is applied after convolutions $c \in \{C_1, C_2\}$ at blocks $l \in \{3, 4, 5, 6\}$. We set $\Delta_p = 0.0001, \tau = 0.9, \lambda = 0.05$.

7 Few-shot images (1024 × 1024) We replace the large discriminator in FastGAN with the one from BigGAN while removing the smaller discriminator. This modification yields FastGAN−D_{big}, with the discriminator network architecture illustrated in Figure 9. We employ a batch size of 8 and run for $100K$ iterations. We equip the discriminator with CHAIN after convolutional layers $c \in \{C_1, C_2, C_S\}$ at blocks $l \in \{1, 2, 3, 4, 5\}$. We set $\Delta_p = 0.001, \tau = 0.5, \lambda = 20$.

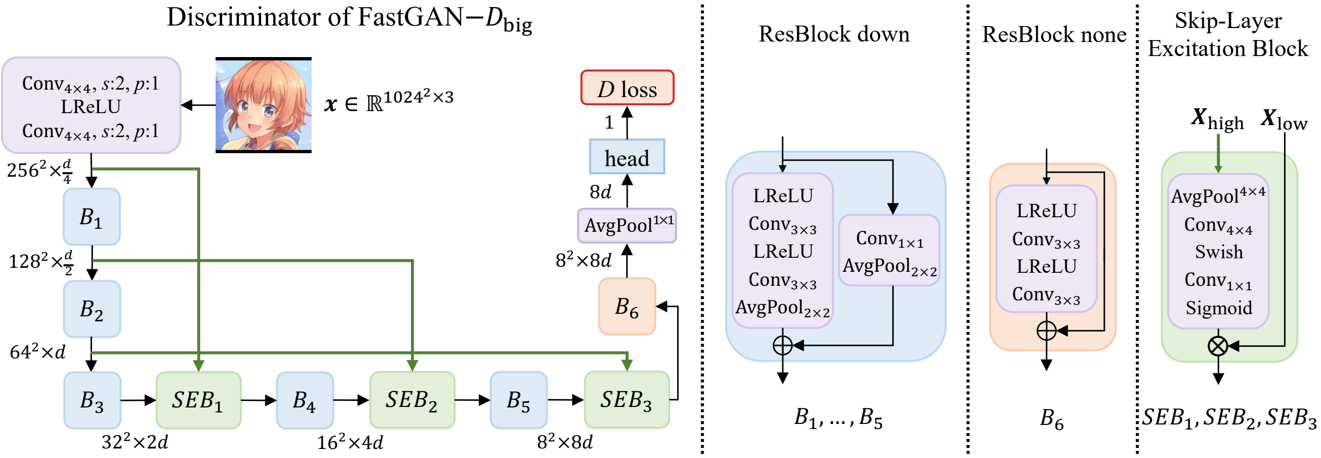


Figure 9. The discriminator of FastGAN−D_{big}. d: The base feature dimension. Conv$_{4\times4}$: A convolutional layer with a 4 × 4 kernel size. LReLU: Leaky ReLU activation with a slope 0.2. AvgPool$_{2\times2}$: Average pooling downscales by a factor of 2. AvgPool$^{4\times4}$: Adaptive average pooling with a 4 × 4 output spatial size. $\boldsymbol{X}_{\text{high}}$: The higher resolution feature map. $\boldsymbol{X}_{\text{low}}$: The lower resolution feature map. For more details on skip-layer excitation block, please refer to [57] and [28].

D.3. Implementation for AGP$_{\text{input}}$ and AGP$_{\text{weight}}$

In Table 6, we provide a comparison of CHAIN with two gradient penalization methods: AGP$_{\text{input}}$ and AGP$_{\text{weight}}$. For AGP$_{\text{input}}$, we implement $\|\frac{\partial D}{\partial \boldsymbol{x}}\|_2^2$ and $\|\frac{\partial f}{\partial \boldsymbol{x}}\|_2^2$ where f represents the feature extractor of discriminator D. Regarding AGP$_{\text{weight}}$, we also implement $\|\frac{\partial D}{\partial \boldsymbol{\theta}_d}\|_2^2$ and $\|\frac{\partial f}{\partial \boldsymbol{\theta}_d}\|_2^2$. We search the penalization strength λ_{GP} within the range [1e-10, 20] for each dataset and variant. For 10% CIFAR-10 w/ OmniGAN ($d = 256$), the optimal settings are: AGP$_{\text{input}}$ with $\|\frac{\partial f}{\partial \boldsymbol{x}}\|_2^2$ and λ_{GP}=5, and AGP$_{\text{weight}}$ with $\|\frac{\partial f}{\partial \boldsymbol{\theta}_d}\|_2^2$ and λ_{GP} set to 1e-6. For 10% CIFAR-100 w/ BigGAN (d=256), the best configurations are: AGP$_{\text{input}}$ with $\|\frac{\partial f}{\partial \boldsymbol{x}}\|_2^2$ and λ_{GP}=5, and AGP$_{\text{weight}}$ with $\|\frac{\partial D}{\partial \boldsymbol{\theta}_d}\|_2^2$ and λ_{GP} set to 2e-6.

E. Additional Experiments

E.1. Comparison with leading methods

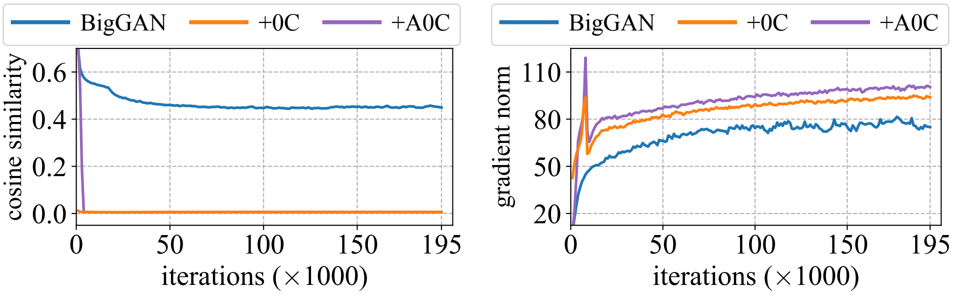
Table 7 compares CHAIN with Lottery-GAN [12], LCSA [70], AugSelf-GAN [27], and NICE [68], showing the superiority of CHAIN. Unlike AugSelf-GAN, LotteryGAN, and NICE, which need extra forward or backward passes for augmentation, and LCSA, which demands more computation and weights for dictionary learning, CHAIN is more efficient, needing negligible computation for normalization.

Table 7. Comparing CIFAR-10/100 results with varying data percentages, using CHAIN *vs*. other leading methods, on BigGAN (d=256).

Method	CIFAR-10									CIFAR-100								
	10% data			20% data			100% data			10% data			20% data			100% data		
	IS↑	tFID↓	vFID↓	IS↑	tFID↓	vFID↓	IS↑	tFID↓	vFID↓	IS↑	tFID↓	vFID↓	IS↑	tFID↓	vFID↓	IS↑	tFID↓	vFID↓
LeCam+DA	8.81	12.64	16.42	9.01	8.53	12.47	9.45	4.32	8.40	9.17	22.75	27.14	10.12	15.96	20.42	11.25	6.45	11.26
+Lottery-GAN	8.77	11.47	15.48	8.99	7.91	11.83	9.39	4.21	8.25	9.05	20.63	25.31	9.55	15.18	20.01	11.28	6.32	11.10
+LCSA	8.96	10.05	13.88	9.04	6.95	10.95	9.47	3.75	7.83	**10.28**	18.24	23.12	**10.67**	10.16	15.00	11.17	5.85	10.64
+NICE	8.99	9.86	13.81	9.12	6.92	10.89	9.52	3.72	7.81	9.35	14.95	19.60	10.54	10.02	14.93	11.28	5.72	10.40
+AugSelf-GAN	**9.04**	8.98	12.94	9.13	6.42	10.54	9.48	3.68	7.73	9.89	14.02	18.84	10.43	11.32	16.02	11.25	5.43	10.14
+CHAIN	8.96	**8.54**	**12.51**	**9.27**	**5.92**	**9.90**	**9.52**	**3.51**	**7.47**	10.11	**12.69**	**17.49**	10.62	**9.02**	**13.75**	**11.37**	**5.26**	**9.85**

E.2. Gradient analysis on 10% CIFAR-100 using BigGAN (d = 256)

In this section, we present experiments conducted on 10% CIFAR-100 using BigGAN ($d = 256$). Figure 10 provides additional validation of Theorem 3.1, illustrating how the centering step leads to feature differences and an associated increase in gradients. Meanwhile, Figure 11 confirms Theorem 3.2, highlighting that the scaling step causes gradient explosions during GAN training and results in rank deficiency.



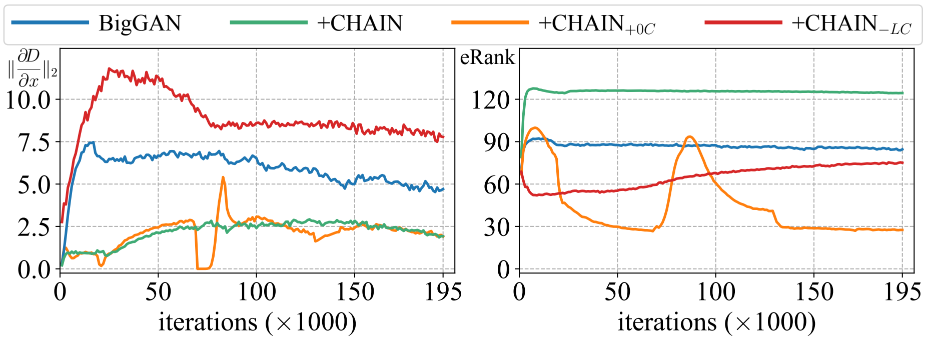
(a) Mean cosine similarity.

(b) Gradient norm.

Figure 10. (a) Mean cosine similarity of discriminator pre-activation features, and (b) gradient norm of the feature extractor w.r.t. the input are evaluated for BigGAN, BigGAN+0C (using the centering step in Eq. 6), and BigGAN+A0C (adaptive interpolation between centered and uncentered features). Evaluation conducted on 10% CIFAR-100 data with BigGAN (d=256).

E.3. The rank efficiency of CHAIN over AGP$_{\text{weight}}$

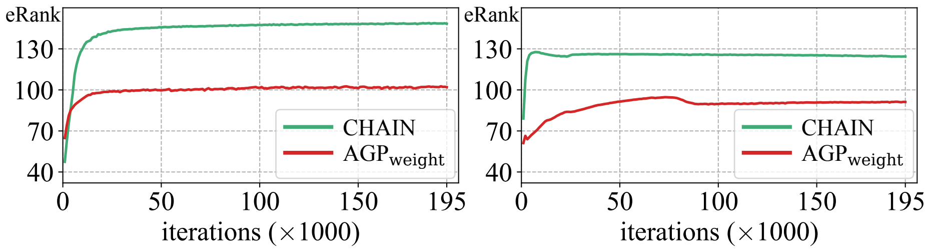
Both CHAIN and AGP$_{\text{weight}}$ can reduce the discriminator weight gradient to improve generalization, but CHAIN gains a crucial advantage from normalization. The normalization step in CHAIN balances features among channels and orthogonalizes features [16, 17]. Figure 12 clearly illustrates that CHAIN achieves a higher effective rank compared to AGP$_{\text{weight}}$. Discriminators with higher rank efficiency can fully utilize their width (balanced channels) and depth, resulting in enhanced expressivity and superior representation capability.



(a) Gradient norm.

(b) Effective rank.

Figure 11. (a) Gradient norm of discriminator output w.r.t. input during training, and (b) effective rank [78] of the pre-activation features in discriminator, are evaluated on 10% CIFAR-100 data with BigGAN ($d = 256$). CHAIN$_{+0C}$ indicates CHAIN with the centering step included, while CHAIN$_{-LC}$ represents CHAIN without the Lipschitzness constraint.



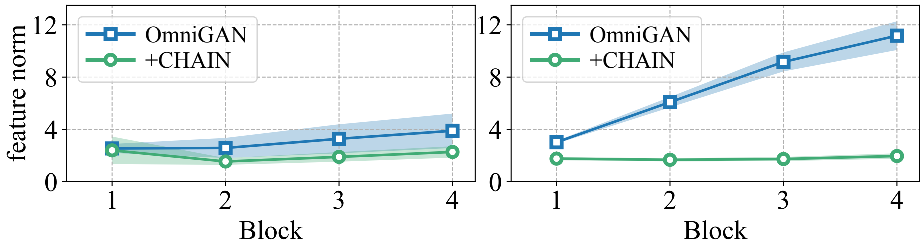
(a) 10% CIFAR-10 with OmniGAN ($d = 256$).

(b) 10% CIFAR-100 with BigGAN ($d = 256$.)

Figure 12. Effective Rank [78] for CHAIN and AGP$_{\text{weight}}$ on (a) 10% CIFAR-10 using OmniGAN ($d = 256$) and (b) 10% CIFAR-100 with BigGAN ($d = 256$).

E.4. The stability of feature norm of CHAIN during training

Our work examines modern discriminators with residual blocks, where the main and skip branch features are added at the end of each block (see Figure 1). Despite the scaling factor $\leqslant 1$ induced by the Lipschitz constraint (as in Eq. 13), feature norms remain stable across layers thanks to the skip connections. Figure 13 presents feature norms at the end of each block, averaged over early ($0 - 5k$ iteration) and later training stages ($> 5k$ iteration). Initially, both methods exhibit similar feature norms, but as training processes, baseline norms increase while CHAIN maintains stable norms across layers due to the adaptive interpolation between normalized and unnormalized features (as in Eq. 14).



(a) $< 5k$ iteration.

(b) $> 5k$ iteration.

Figure 13. Feature norms during training w/ *vs*. w/o CHAIN, are evaluated on 10% CIFAR-10 using OmniGAN ($d = 256$).

F. Training overhead

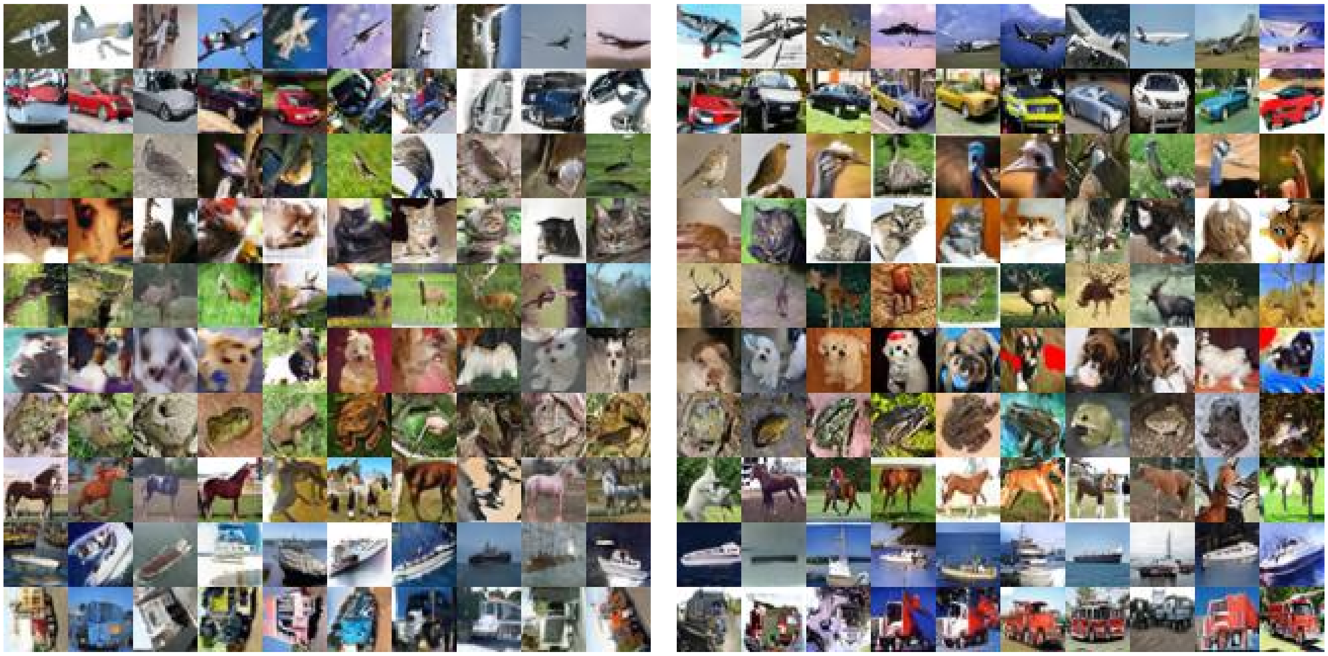
Table 8 presents the number of parameters, multiply-accumulate (MACs) operations (for both generator and discriminator), the number of GPUs, and the cost in time (seconds per 1000 images, secs/$k$img). Notably, CHAIN introduces only a small fraction of the time cost, ranging from 6.3% to 9.6% across these datasets.

Table 8. Number of parameters, MACs and secs/$k$img for models with *vs*. without the CHAIN. Experiments were performed on NVIDA A100 GPUs.

Dataset	Resolution	Backbone	d	GPUs	Baseline			+CHAIN		
					#Par.	MACs	sec/$k$img	#Par.	MACs	sec/$k$img
CIFAR-10	32×32	BigGAN	256	1	8.512M	2.788G	0.79	8.512M	2.791G	0.84
		OmniGAN			8.512M	2.788G	0.80	8.512M	2.790G	0.85
CIFAR-100	32×32	BigGAN	256	1	8.811M	2.788G	0.80	8.811M	2.791G	0.85
		OmniGAN			8.811M	2.788G	0.81	8.811M	2.791G	0.85
ImageNet	64×64	BigGAN	384	2	115.69M	18.84G	1.79	115.69M	19.12G	1.91
5 Low-shot datasets	256×256	StyleGAN2	512	2	48.77M	44.146G	5.66	48.77M	44.151G	6.06
7 Few-shot datasets	1024×1024	FastGAN$-D_{\text{big}}$	64	1	42.11M	23.98G	32.79	42.11M	24.00G	35.94

G. Generated Images

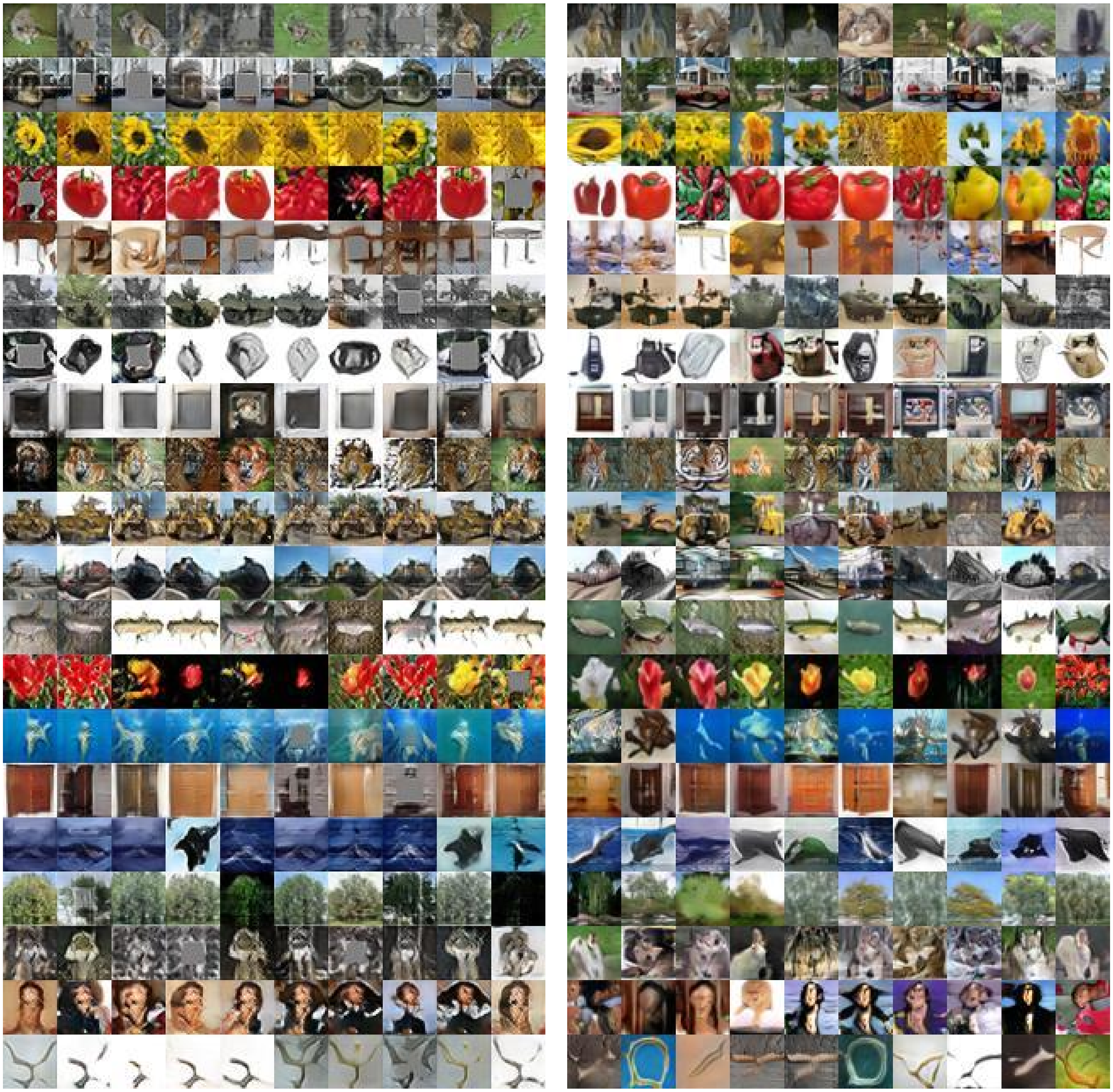
Figures 14, 15, 16, 17 and 18 provide images generated on CIFAR-10, CIFAR-100, ImageNet, the 5 low-shot image and the 7 few-shot image datasets, with or without CHAIN. The comparison highlights the enhancement in image quality and diversity achieved with the application of CHAIN.



(a) ADA

(b) ADA+CHAIN

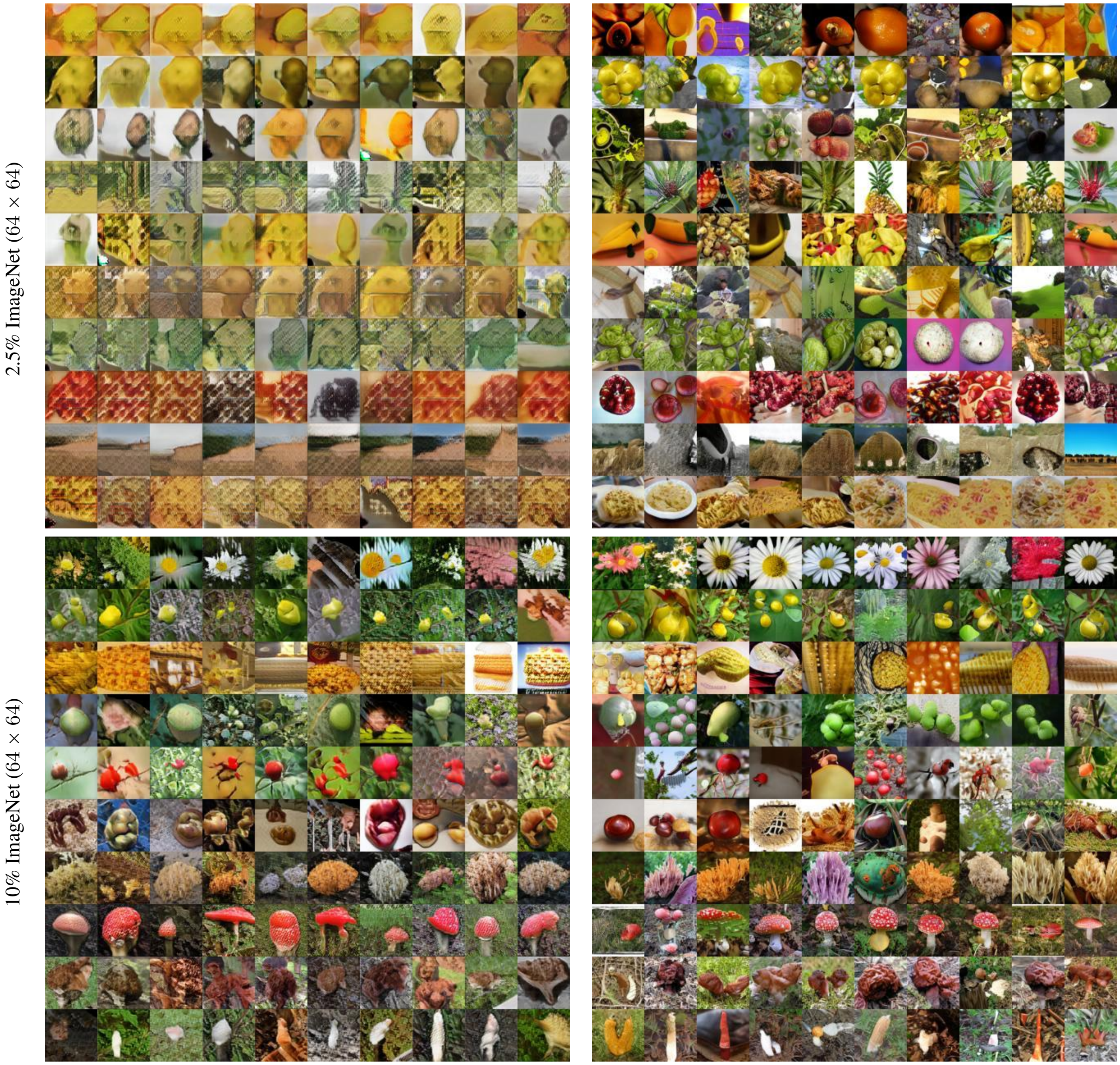
Figure 14. Generated images using (a) ADA and (b) ADA+CHAIN on 10% CIFAR-10 with OmniGAN ($d=1024$). Note that ADA leaks the rotation augmentation artifacts (row 1, 2 and 10).



(a) DA

(b) DA+CHAIN

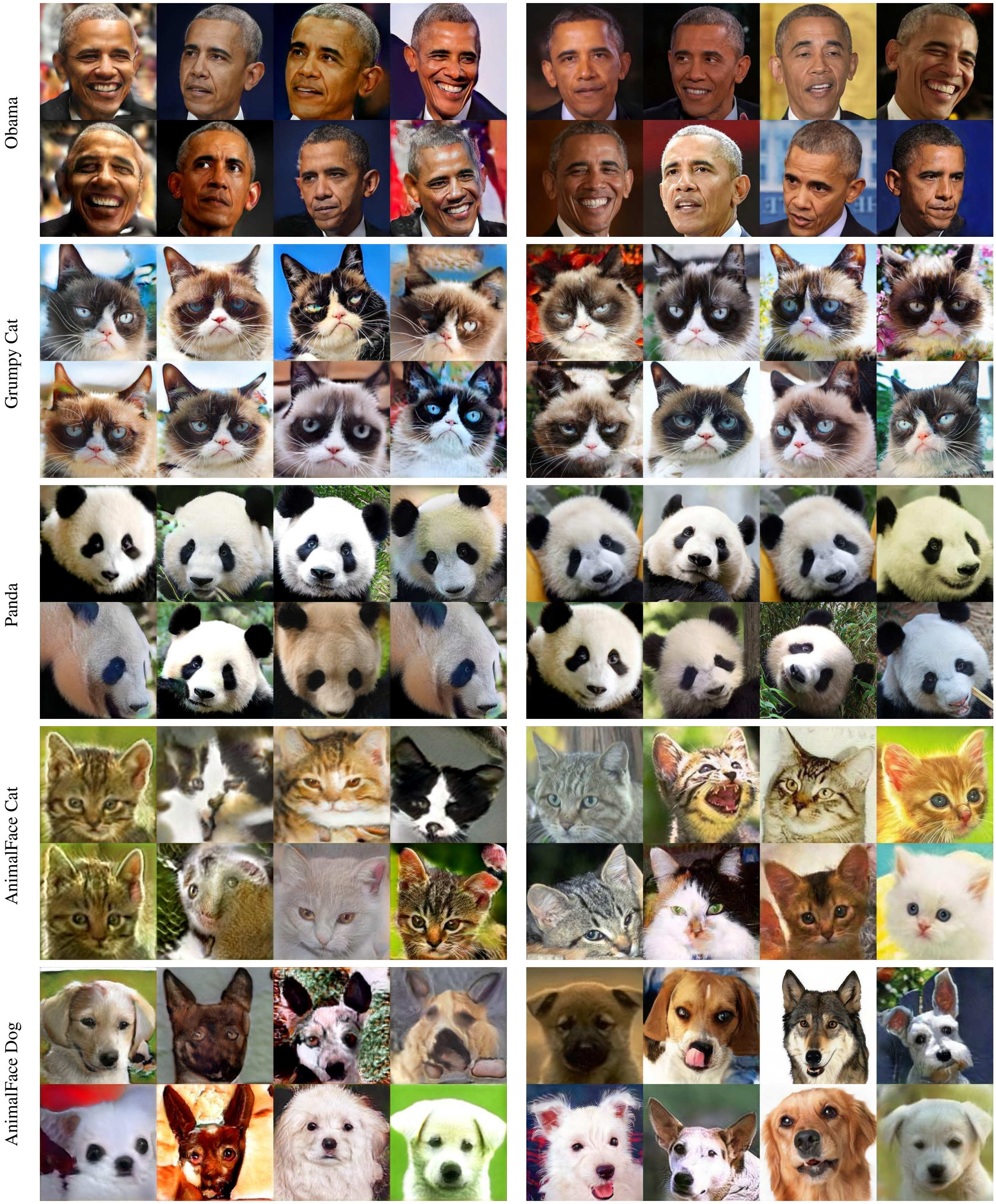
Figure 15. Generated images using (a) DA and (b) DA+CHAIN on 10% CIFAR-100 with BigGAN ($d = 256$). We present the last 20 of 100 classes. CHAIN clearly enhances the diversity and quality of the generated images. Notably, DA leaks the cutout augmentation artifacts (row 1, 2, 4 and 18).



(a) BigGAN+ADA

(b) BigGAN+ADA+CHAIN

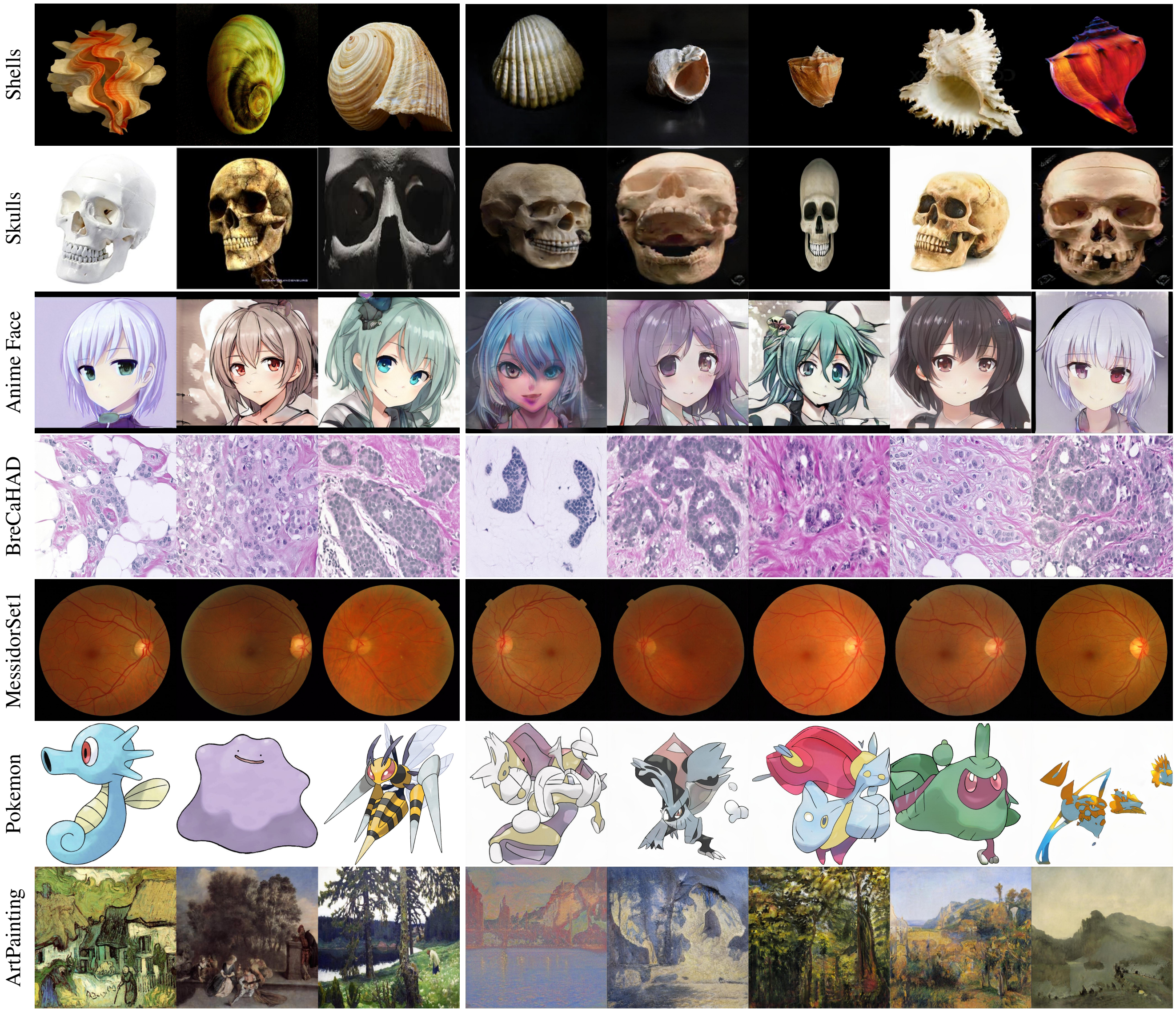
Figure 16. Visual comparison between ADA *vs*. ADA+CHAIN on 2.5% and 10% ImageNet(64 × 64) data. ADA struggles to capture the structure and diversity of the data, while CHAIN clear improves the diversity and visual quality of generated images.



(a) StyleGAN2+ADA

(b) StyleGAN2+ADA+CHAIN

Figure 17. Visual comparison between ADA and ADA+CHAIN on 100-shot and AnimalFace datasets (256 × 256). The integration of CHAIN clearly improves the image quality.



(a) Real images

(b) FastGAN−D_{big}+CHAIN.

Figure 18. Qualitative results of FastGAN−D_{big}+CHAIN on 7 few-shot image dataset (1024×1024). (a) shows real training images and the (b) presents images generated by FastGAN−D_{big}+CHAIN. CHAIN is capable of generating photo-realistic images with fine details even from a limited number of training samples.