Theoretical analysis of two time-scale update rule for training GANs

Naoki Sato[1], Hideaki Iiduka[1]

[1]Meiji University

Aug.23, 15:55-16:20

Prior knowledge

【I. mini-batch size】

- Training datasets $S := (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n)$
- Deep Neural Networks' parameter $\boldsymbol{\theta} \in \mathbb{R}^{\Theta}$
- Empirical risk minimization problem

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{\Theta}} f(\boldsymbol{\theta}; S) = \frac{1}{n} \sum_{i=1}^{n} l(\boldsymbol{\theta}; \boldsymbol{z}_i) = \frac{1}{n} \sum_{i=1}^{n} l_i(\boldsymbol{\theta})$$

where, $l_i(\cdot) := l(\cdot; z_i)$ is the loss function for the i-th training data $\boldsymbol{z}_i$.

Prior knowledge

【I. mini-batch size】

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{\Theta}} f(\boldsymbol{\theta}; S) = \frac{1}{n}\sum_{i=1}^{n} l(\boldsymbol{\theta}; \boldsymbol{z}_i) = \frac{1}{n}\sum_{i=1}^{n} l_i(\boldsymbol{\theta})$$

- Deep learning optimizers are used to solve this problem.
- Stochastic Gradient Descent (SGD)

$$\boldsymbol{d}_n := -\nabla f_{B_n}(\boldsymbol{\theta}_n) = \frac{1}{b}\sum_{i=1}^{b} \mathsf{G}_{\xi_{n,i}}(\boldsymbol{\theta}_n)$$

$$\boldsymbol{\theta}_{n+1} := \boldsymbol{\theta}_n + \alpha_n \boldsymbol{d}_n$$

Prior knowledge

【I. mini-batch size】

▷Gradient Descent Method

$$\boldsymbol{d}_n := -\nabla f(\boldsymbol{\theta}_n)$$

Full Gradient

$$\boldsymbol{\theta}_{n+1} := \boldsymbol{\theta}_n + \alpha_n \boldsymbol{d}_n$$

▷Stochastic Gradient Descent (SGD)

$$\boldsymbol{d}_n := -\nabla f_{B_n}(\boldsymbol{\theta}_n) = \frac{1}{b}\sum_{i=1}^{b} \mathsf{G}_{\xi_{n,i}}(\boldsymbol{\theta}_n)$$

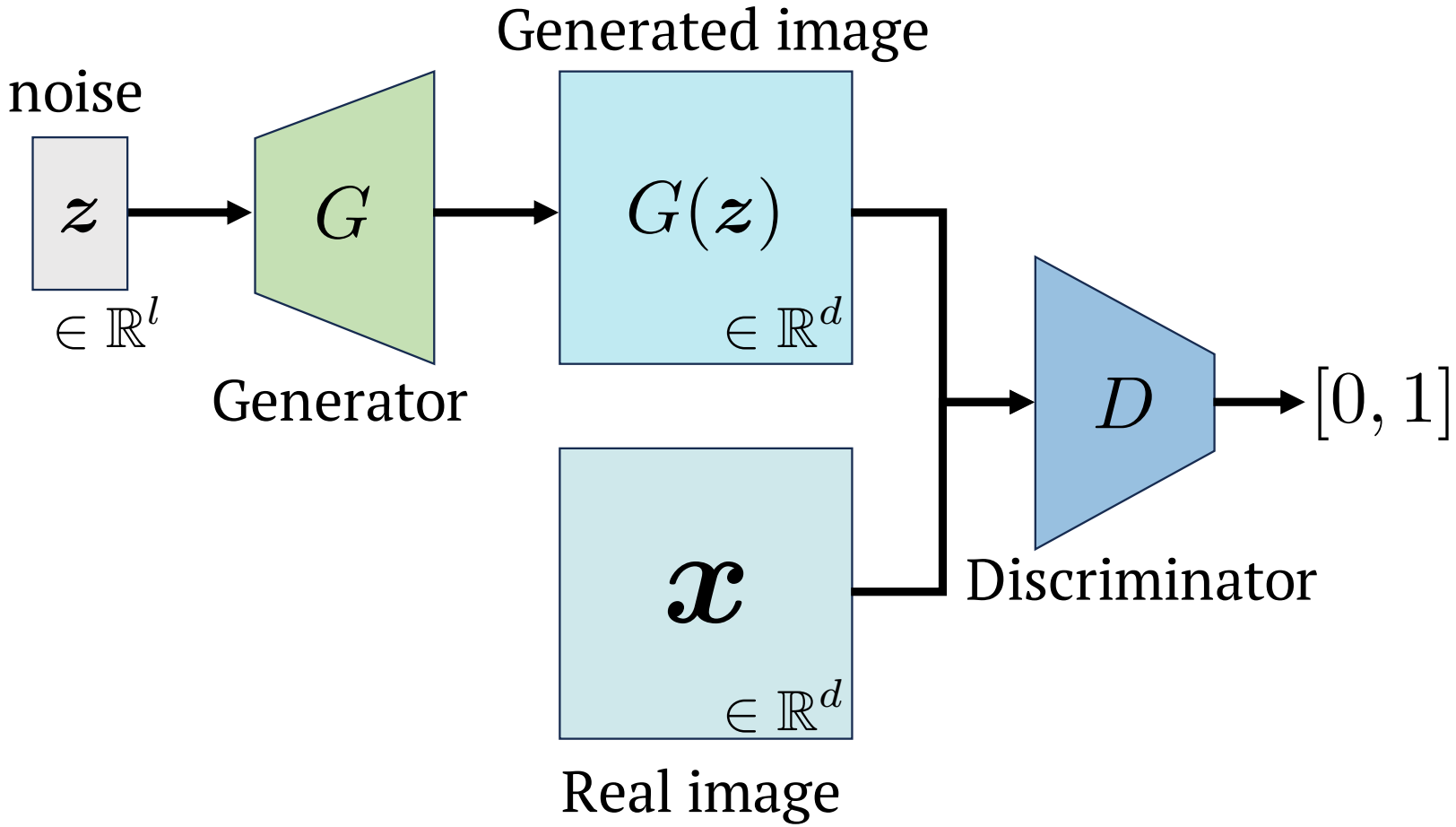
Stochastic Gradient

$$\boldsymbol{\theta}_{n+1} := \boldsymbol{\theta}_n + \alpha_n \boldsymbol{d}_n$$

mini-batch size

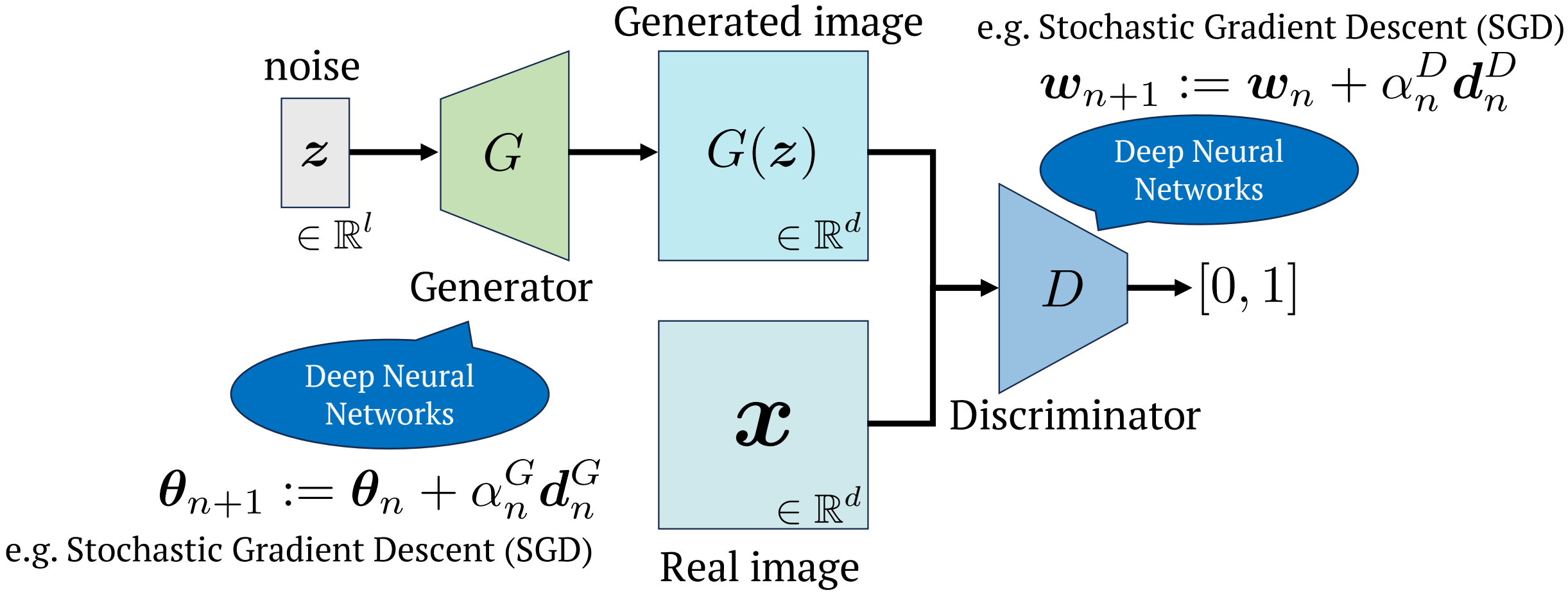
Prior knowledge

【II. Generative Adversarial Networks (GANs)】



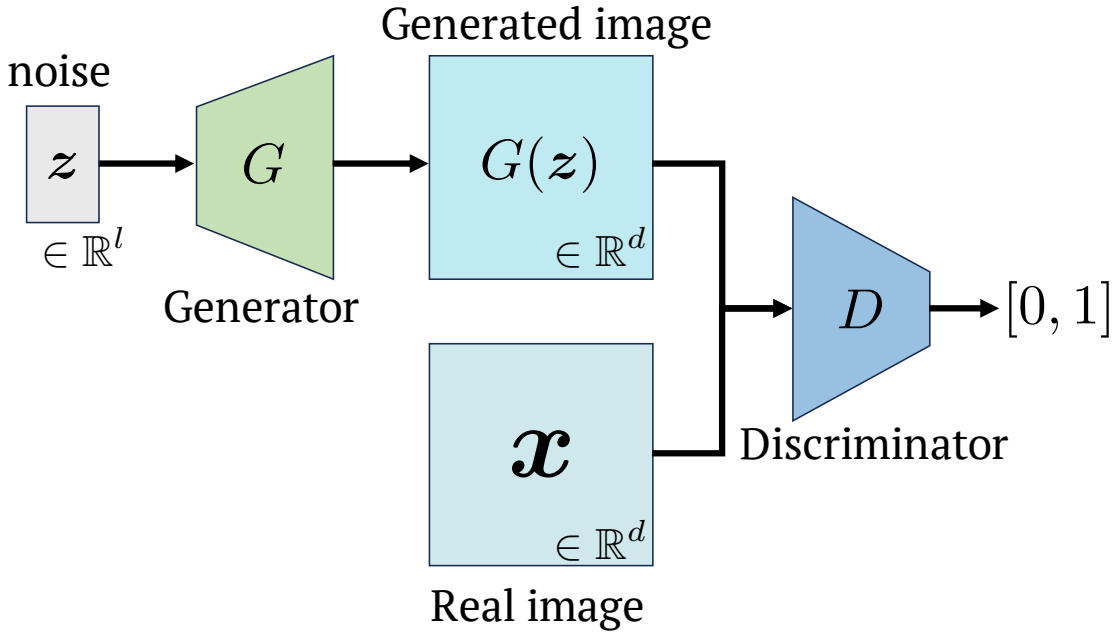
Prior knowledge

【II. Generative Adversarial Networks (GANs)】



Prior knowledge

【II. Generative Adversarial Networks (GANs)】



Real images

Generated images

$$\min_G \max_D V(D, G) := \mathbb{E}_{\boldsymbol{x}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}}[\log(1 - D(G(\boldsymbol{z})))]$$

Prior knowledge

【III. Fréchet Inception Distance (FID)】

Real images
(FID=0)

Generated images
(FID=287)

Generated images
(FID=12)

Introduction

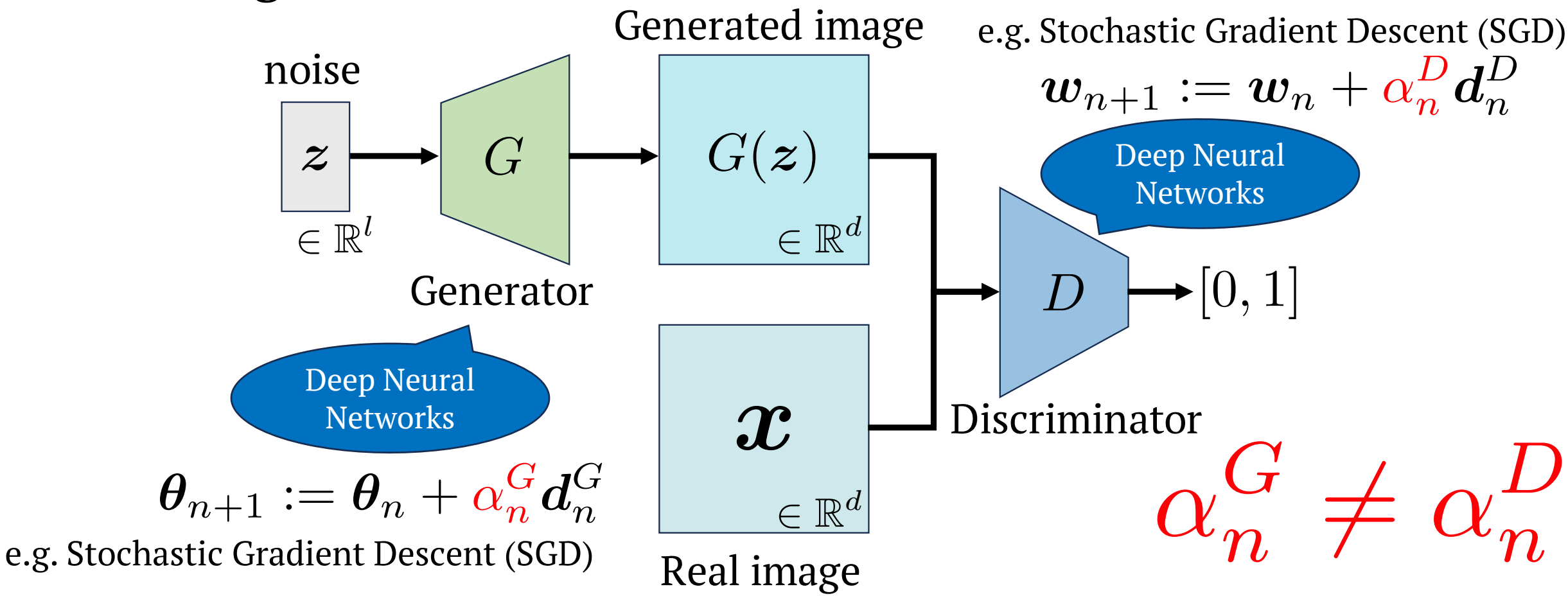
【Background】

- [Heu+17] have shown that a two time-scale update rule (TTUR) is useful for training Generative Adversarial Networks (GANs) in theory and in practice.
- TTUR implies using different learning rates for the generator and discriminator.

[Heu+17] M. Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6629–6640, 2017.

Introduction

【Background】



Introduction

【Motivation】

- There is no convergence proof for TTUR with constant learning rates.
- For DNN training, there is a critical batch size that is optimal for training in terms of computational complexity. [Sha+19]

[Sha+19] C.J. Shallue et al. “Measuring the effects of data parallelism on neural network training.” *Journal of Machine Learning Research*, 20:1–49, 2019.

Introduction

【Contribution】

▷We provide convergence for TTUR with constant learning rates.

▷We showed that there is a critical batch size for training GANs.

▷We also showed that the critical batch size can be estimated.

＊Note＊

Critical batch size implies the optimal batch size for training in terms of computational complexity.

Mathematical Preliminaries

【Definition】

$\mathcal{S}$: A set of synthetic samples $\boldsymbol{z}^{(i)}$.

$\mathcal{R}$: A set of real-world samples $\boldsymbol{x}^{(i)}$.

$\mathcal{S}_n$: Mini-batch of b synthetic samples $\boldsymbol{z}^{(i)}$ at time n.

$\mathcal{R}_n$:Mini-batch of b real world samples $\boldsymbol{x}^{(i)}$ at time n.

$L_G^{(i)}(\cdot, \boldsymbol{w})$: A loss function of the generator for $\boldsymbol{w} \in \mathbb{R}^W$ and $\boldsymbol{z}^{(i)}$.

$L_G(\cdot, \boldsymbol{w})$: The total loss function of the generator for $\boldsymbol{w} \in \mathbb{R}^W$.

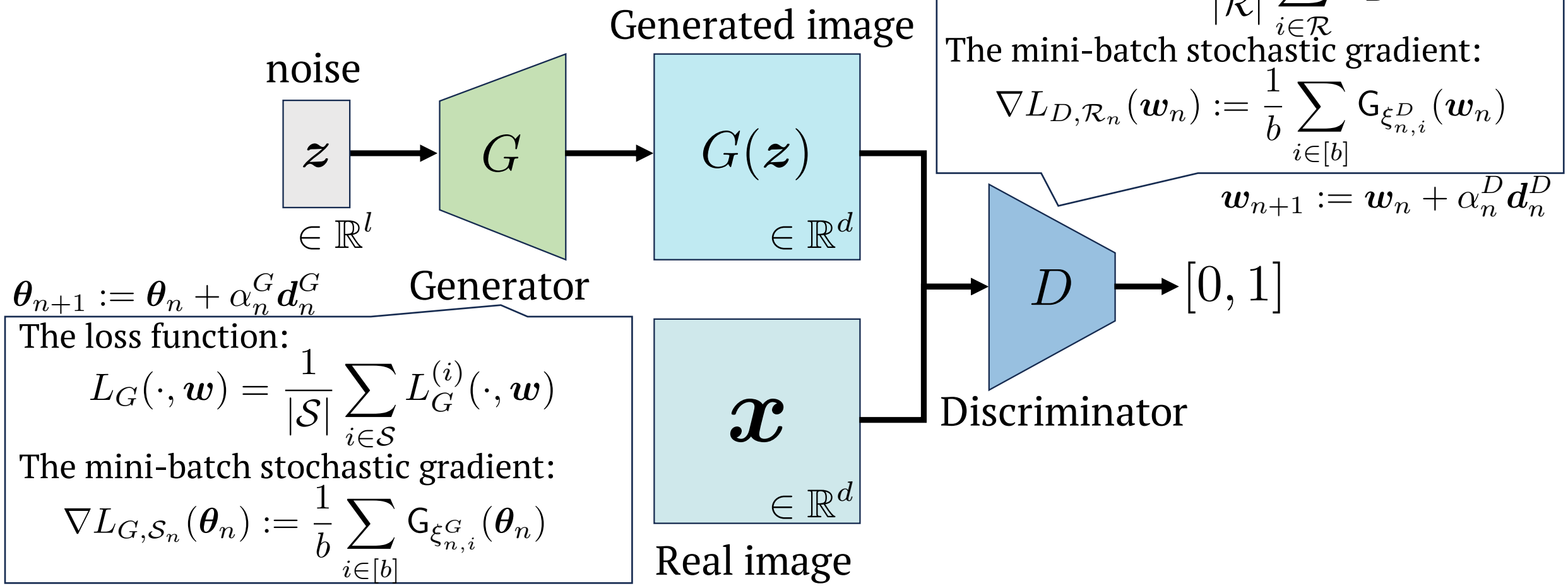
ξ^G : A random variable that does not depend on $\boldsymbol{w} \in \mathbb{R}^W$ and $\boldsymbol{\theta} \in \mathbb{R}^\Theta$.

$\mathsf{G}_{\xi^G}(\boldsymbol{\theta})$: The stochastic gradient of $L_G(\cdot, \boldsymbol{w})$ at $\boldsymbol{\theta} \in \mathbb{R}^\Theta$.

$\nabla L_{G,\mathcal{S}_n}(\boldsymbol{\theta}_n)$: The mini-batch stochastic gradient of $L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n)$.

Mathematical Preliminaries

【Definition】



Mathematical Preliminaries

【Assumptions】

(S1) $L_G^{(i)}(\cdot, \boldsymbol{w}) \colon \mathbb{R}^{\Theta} \to \mathbb{R}$ and $L_D^{(i)}(\boldsymbol{\theta}, \cdot) \colon \mathbb{R}^{W} \to \mathbb{R}$ are continuously differentiable.

(S2) Let $((\boldsymbol{\theta}_n, \boldsymbol{w}_n))_{n \in \mathbb{N}} \subset \mathbb{R}^{\Theta} \times \mathbb{R}^{W}$ be the sequence generated by an optimizer.

(i) For each iteration n,

$$\mathbb{E}_{\xi_n^G}\left[\mathsf{G}_{\xi_n^G}(\boldsymbol{\theta}_n)\right] = \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n),\ \mathbb{E}_{\xi_n^D}\left[\mathsf{G}_{\xi_n^D}(\boldsymbol{w}_n)\right] = \nabla_{\boldsymbol{w}} L_D(\boldsymbol{\theta}_n, \boldsymbol{w}_n)$$

(ii) There exist nonnegative constants σ_G^2 and σ_D^2 such that

$$\mathbb{E}_{\xi_n^G}\left[\|\mathsf{G}_{\xi_n^G}(\boldsymbol{\theta}_n) - \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n)\|^2\right] \le \sigma_G^2,\ \mathbb{E}_{\xi_n^D}\left[\|\mathsf{G}_{\xi_n^D}(\boldsymbol{w}_n) - \nabla_{\boldsymbol{w}} L_D(\boldsymbol{\theta}_n, \boldsymbol{w}_n)\|^2\right] \le \sigma_D^2$$

Mathematical Preliminaries

【Assumptions】

(S3) For each iteration n, the optimizer samples mini-batches $\mathcal{S}_n \subset \mathcal{S}$ and $\mathcal{R}_n \subset \mathcal{R}$ and estimates the full gradient ∇L_G and ∇L_D.

$$\nabla L_{G,\mathcal{S}_n}(\boldsymbol{\theta}_n) := \frac{1}{b}\sum_{i\in[b]} \mathsf{G}_{\xi^G_{n,i}}(\boldsymbol{\theta}_n) = \frac{1}{b}\sum_{\{i:\, \boldsymbol{z}^{(i)}\in\mathcal{S}_n\}} \nabla_{\boldsymbol{\theta}} L_G^{(i)}(\boldsymbol{\theta}_n, \boldsymbol{w}_n)$$

$$\nabla L_{D,\mathcal{R}_n}(\boldsymbol{w}_n) := \frac{1}{b}\sum_{i\in[b]} \mathsf{G}_{\xi^D_{n,i}}(\boldsymbol{w}_n) = \frac{1}{b}\sum_{\{i:\, \boldsymbol{x}^{(i)}\in\mathcal{R}_n\}} \nabla_{\boldsymbol{w}} L_D^{(i)}(\boldsymbol{\theta}_n, \boldsymbol{w}_n)$$

Mathematical Preliminaries

【Problems】

▷Find a pair $(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) \in \mathbb{R}^\Theta \times \mathbb{R}^W$ satisfying

$$\nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) = \mathbf{0} \text{ and } \nabla_{\boldsymbol{w}} L_D(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) = \mathbf{0}.$$

$$\Updownarrow$$

▷Find a pair $(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) \in \mathbb{R}^\Theta \times \mathbb{R}^W$ satisfying

$$\forall \boldsymbol{\theta} \in \mathbb{R}^\Theta, \forall \boldsymbol{w} \in \mathbb{R}^W :$$
$$\left\langle \boldsymbol{\theta}^\star - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) \right\rangle \leq 0,$$
$$\left\langle \boldsymbol{w}^\star - \boldsymbol{w}, \nabla_{\boldsymbol{w}} L_D(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) \right\rangle \leq 0.$$

Mathematical Preliminaries

【Problem 1】

▷Find a pair $(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) \in \mathbb{R}^\Theta \times \mathbb{R}^W$ satisfying

$$\forall \boldsymbol{\theta} \in \mathbb{R}^\Theta, \forall \boldsymbol{w} \in \mathbb{R}^W :$$

$$\langle \boldsymbol{\theta}^\star - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) \rangle \leq 0,$$

$$\langle \boldsymbol{w}^\star - \boldsymbol{w}, \nabla_{\boldsymbol{w}} L_D(\boldsymbol{\theta}^\star, \boldsymbol{w}^\star) \rangle \leq 0.$$

【Algorithms】

▷We use Adam, AdaBelief and RMSProp.

Algorithm 1 Adaptive Method for Solving Problem (1)

Require: $(\alpha_n^G)_{n\in\mathbb{N}}, (\alpha_n^D)_{n\in\mathbb{N}} \subset \mathbb{R}_{++}$, $\beta_1^G, \beta_1^D \in [0,1)$, $\gamma^G, \gamma^D \in [0,1)$

$n \leftarrow 0$, $(\boldsymbol{\theta}_0, \boldsymbol{w}_0) \in \mathbb{R}^\Theta \times \mathbb{R}^W$, $\boldsymbol{m}_{-1}^G = \boldsymbol{0} \in \mathbb{R}^\Theta$, $\boldsymbol{m}_{-1}^D = \boldsymbol{0} \in \mathbb{R}^W$

loop

  loop

    $\boldsymbol{m}_n^G := \beta_1^G \boldsymbol{m}_{n-1}^G + (1-\beta_1^G)\nabla L_{G,\mathcal{S}_n}(\boldsymbol{\theta}_n)$

    $\hat{\boldsymbol{m}}_n^G := (1-\gamma^{G^{n+1}})^{-1} \boldsymbol{m}_n^G$

    $\mathsf{H}_n^G \in \mathbb{S}_{++}^\Theta \cap \mathbb{D}^\Theta$

    Find $\boldsymbol{d}_n^G \in \mathbb{R}^\Theta$ that solves $\mathsf{H}_n^G \boldsymbol{d} = -\hat{\boldsymbol{m}}_n^G$

    $\boldsymbol{\theta}_{n+1} := \boldsymbol{\theta}_n + \alpha_n^G \boldsymbol{d}_n^G$

  end loop

  loop

    $\boldsymbol{m}_n^D := \beta_1^D \boldsymbol{m}_{n-1}^D + (1-\beta_1^D)\nabla L_{D,\mathcal{R}_n}(\boldsymbol{w}_n)$

    $\hat{\boldsymbol{m}}_n^D := (1-\gamma^{D^{n+1}})^{-1} \boldsymbol{m}_n^D$

    $\mathsf{H}_n^D \in \mathbb{S}_{++}^W \cap \mathbb{D}^W$

    Find $\boldsymbol{d}_n^D \in \mathbb{R}^W$ that solves $\mathsf{H}_n^D \boldsymbol{d} = -\hat{\boldsymbol{m}}_n^D$

    $\boldsymbol{w}_{n+1} := \boldsymbol{w}_n + \alpha_n^D \boldsymbol{d}_n^D$

  end loop

  $n \leftarrow n+1$

end loop

Theoretical Analysis

【Convergence Analysis】

▷For all $\boldsymbol{\theta} \in \mathbb{R}^{\Theta}$, all $\boldsymbol{w} \in \mathbb{R}^{W}$, and all $N \geq 1$,

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\langle \boldsymbol{\theta}_n - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n)\rangle\right] \leq \underbrace{\frac{\Theta \mathrm{Dist}(\boldsymbol{\theta}) H^G}{2\alpha^G \tilde{\beta}_1^G}}_{A_G}\frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G}}_{B_G}\frac{1}{b} + \underbrace{\frac{M_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G} + \frac{\beta_1^G}{\tilde{\beta}_1^G}\sqrt{\Theta \mathrm{Dist}(\boldsymbol{\theta})(\sigma_G^2 + M_G^2)}}_{C_G}$$

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\langle \boldsymbol{w}_n - \boldsymbol{w}, \nabla_{\boldsymbol{w}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n)\rangle\right] \leq \underbrace{\frac{W \mathrm{Dist}(\boldsymbol{w}) H^D}{2\alpha^D \tilde{\beta}_1^D}}_{A_D}\frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D}}_{B_D}\frac{1}{b} + \underbrace{\frac{M_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D} + \frac{\beta_1^D}{\tilde{\beta}_1^D}\sqrt{W \mathrm{Dist}(\boldsymbol{w})(\sigma_D^2 + M_D^2)}}_{C_D}$$

Theoretical Analysis

【Relationship between b and N】

▷For all $\boldsymbol{\theta} \in \mathbb{R}^{\Theta}$, all $\boldsymbol{w} \in \mathbb{R}^{W}$, and all $N \geq 1$,

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\langle \boldsymbol{\theta}_n - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n)\rangle\right] \leq \underbrace{\frac{\Theta \mathrm{Dist}(\boldsymbol{\theta}) H^G}{2\alpha^G \tilde{\beta}_1^G}}_{A_G}\frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G}}_{B_G}\frac{1}{b} + \underbrace{\frac{M_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G} + \frac{\beta_1^G}{\tilde{\beta}_1^G}\sqrt{\Theta \mathrm{Dist}(\boldsymbol{\theta})(\sigma_G^2 + M_G^2)}}_{C_G}$$

$\rightarrow$ small

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\langle \boldsymbol{w}_n - \boldsymbol{w}, \nabla_{\boldsymbol{w}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n)\rangle\right] \leq \underbrace{\frac{W \mathrm{Dist}(\boldsymbol{w}) H^D}{2\alpha^D \tilde{\beta}_1^D}}_{A_D}\frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D}}_{B_D}\frac{1}{b} + \underbrace{\frac{M_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D} + \frac{\beta_1^D}{\tilde{\beta}_1^D}\sqrt{W \mathrm{Dist}(\boldsymbol{w})(\sigma_D^2 + M_D^2)}}_{C_D}$$

$\rightarrow$ small

Theoretical Analysis

【Relationship between b and N】

▷For all $\boldsymbol{\theta} \in \mathbb{R}^{\Theta}$, all $\boldsymbol{w} \in \mathbb{R}^{W}$, and all $N \geq 1$,

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\langle \boldsymbol{\theta}_n - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n)\rangle\right] \leq \underbrace{\frac{\Theta \mathrm{Dist}(\boldsymbol{\theta}) H^G}{2\alpha^G \tilde{\beta}_1^G}}_{A_G}\frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G}}_{B_G}\frac{1}{b} + \underbrace{\frac{M_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G} + \frac{\beta_1^G}{\tilde{\beta}_1^G}\sqrt{\Theta \mathrm{Dist}(\boldsymbol{\theta})(\sigma_G^2 + M_G^2)}}_{C_G}$$

$$=: \epsilon_G^2$$

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\langle \boldsymbol{w}_n - \boldsymbol{w}, \nabla_{\boldsymbol{w}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n)\rangle\right] \leq \underbrace{\frac{W \mathrm{Dist}(\boldsymbol{w}) H^D}{2\alpha^D \tilde{\beta}_1^D}}_{A_D}\frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D}}_{B_D}\frac{1}{b} + \underbrace{\frac{M_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D} + \frac{\beta_1^D}{\tilde{\beta}_1^D}\sqrt{W \mathrm{Dist}(\boldsymbol{w})(\sigma_D^2 + M_D^2)}}_{C_D}$$

$$=: \epsilon_D^2$$

Theoretical Analysis

【Relationship between b and N】

- Suppose that the generated images achieves a low FID and the GAN's training has been completed.
- Let $\epsilon_G, \epsilon_D > 0$ be sufficiently small positive numbers such that,

$$\frac{A_G}{N_G} + \frac{B_G}{b} + C_G = \epsilon_G^2, \ \frac{A_D}{N_D} + \frac{B_D}{b} + C_D = \epsilon_D^2$$

i.e.

$$N_G(b) = \frac{A_G b}{(\epsilon_G^2 - C_G)b - B_G}, \ N_D(b) = \frac{A_D b}{(\epsilon_D^2 - C_D)b - B_D}$$

- We see that $N_G(b)$ and $N_D(b)$ are monotone decreasing and convex with respect to b.

Theoretical Analysis

【Existence of a Critical Batch Size】

- Stochastic First-Order Oracle (SFO) complexity is a computational complexity.
- SFO can be defined by $N(b)b$.

$$N_G(b)b = \frac{A_G b^2}{(\epsilon_G^2 - C_G)b - B_G}, \quad N_D(b)b = \frac{A_D b^2}{(\epsilon_D^2 - C_D)b - B_D}$$

- We see that $N_G(b)b$ and $N_D(b)b$ are convex function.
- Also, there exists a $b^\star$ minimizing $N(b)b$,

$$b_G^\star := \frac{2B_G}{\epsilon_G^2 - C_G}, \quad b_D^\star := \frac{2B_D}{\epsilon_D^2 - C_D}$$

Theoretical Analysis

【Estimation of the Critical Batch Size】

▷The lower bound of $b^\star$ can be expressed as follows

(i) for Adam,

$$b^\star_G \geq \frac{\sigma^2_G}{\epsilon^3_G} \frac{\alpha^G}{(1-\beta_1^G)^3 \sqrt{\frac{\Theta}{1-\beta_2^G} \frac{1}{|S|^2}}}$$

(ii) for AdaBelief,

$$b^\star_G \geq \frac{\sigma^2_G}{\epsilon^3_G} \frac{\alpha^G}{(1-\beta_1^G)^3 \sqrt{\frac{4\Theta}{1-\beta_2^G} \frac{1}{|S|^2}}}$$

(iii) for RMSProp,

$$b^\star_G \geq \frac{\sigma^2_G}{\epsilon^3_G} \frac{\alpha^G}{\sqrt{\frac{\Theta}{|S|^2}}}$$

Theoretical Analysis

【Estimation of the Critical Batch Size】

▷The lower bound of $b^\star$ can be expressed as follows

(i) for Adam,

$$b_G^\star \geq \frac{\sigma_G^2}{\epsilon_G^3} \frac{\alpha^G}{(1-\beta_1^G)^3 \sqrt{\frac{\Theta}{1-\beta_2^G} \frac{1}{|S|^2}}}$$

【Notation】

α^G ：the learning rate for the Generator

β_1^G, β_2^G ：parameters for the adaptive optimizer

Θ ：the dimensions of the Generator model

$|S|$ ：the number of items in the datasets

Theoretical Analysis

【Estimation of the Critical Batch Size】

▷The lower bound of $b^\star$ can be expressed as follows

(i) for Adam,

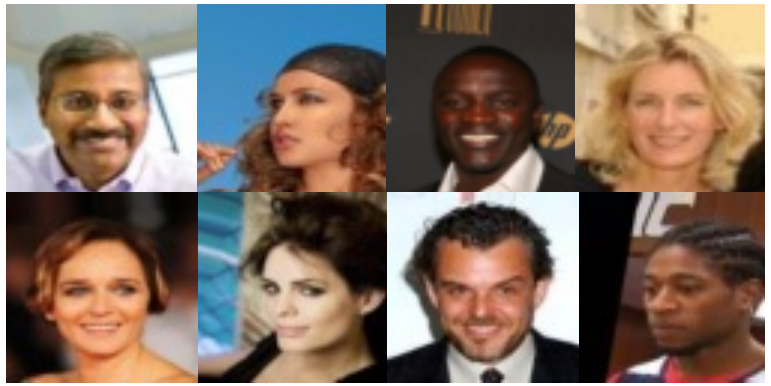
$$b_G^\star \geq \frac{\sigma_G^2}{\epsilon_G^3} \frac{\alpha^G}{(1-\beta_1^G)^3 \sqrt{\frac{\Theta}{1-\beta_2^G} \frac{1}{|S|^2}}}$$

▷This indicates that it is possible to estimate the critical batch size specific to the model-dataset-optimizer combination.

Numerical Results

- Training DCGAN on the LSUN-Bedroom dataset
- Training WGAN-GP on the CelebA dataset
- Training BigGAN on the ImageNet dataset

- LSUN- Bedroom dataset
  - Bedroom images
  - 64×64 size
  - Nearly 3 million images



- CelebA dataset
  - Human face images
  - 64×64 size
  - About 170,000 images

- ImageNet dataset
  - 1000-class images
  - 256×256 size
  - About 1.28 million images

Numerical Results

- Training DCGAN on the LSUN-Bedroom dataset (Section 4.1)
- Training WGAN-GP on the CelebA dataset (Section 4.2)
- Training BigGAN on the ImageNet dataset (Section 4.3)

Table 2. Parameters used to train GANs

	Section 4.1	Section 4.2	Section 4.3
the dimensions of the Generator model → Θ	$3,576,704$	$3,576,704$	$70,433,795$
the dimensions of the Discriminator model → $\mathbf{W}$	$2,765,568$	$2,765,568$	$87,982,369$
the number of items in the datasets → $\lvert S \rvert$	$3,033,042$	$162,770$	$1,281,167$

Training DCGAN

Training WGAN-GP

Training BigGAN

Numerical Results

【Relationship between b and N】

$$N_G(b) = \frac{A_G b}{(\epsilon_G^2 - C_G)b - B_G}$$

- From our theory, $N_G(b)$ is monotone decreasing and convex with respect to b.
- $N_G(b)$ is the number of steps required for training to converge.

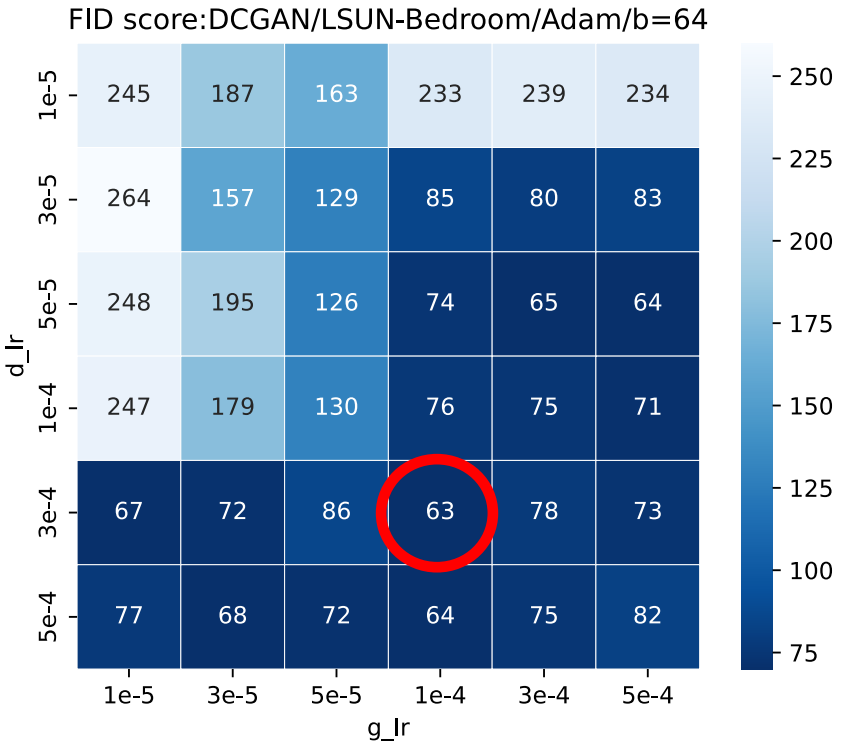
≈

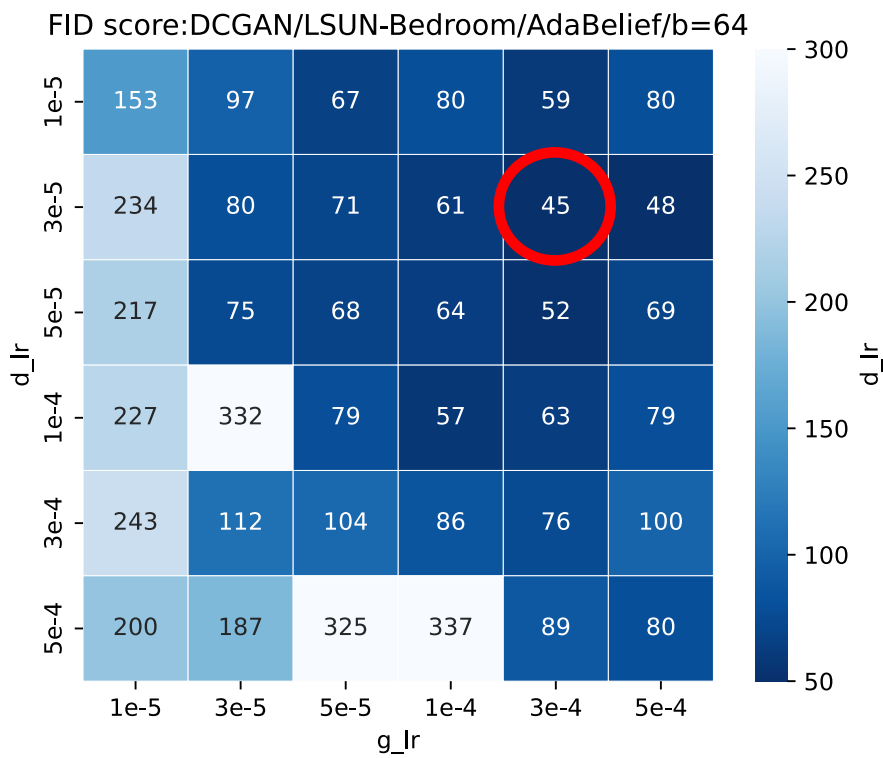
$N_G(b)$ is the number of steps required to achieve a sufficiently low FID.

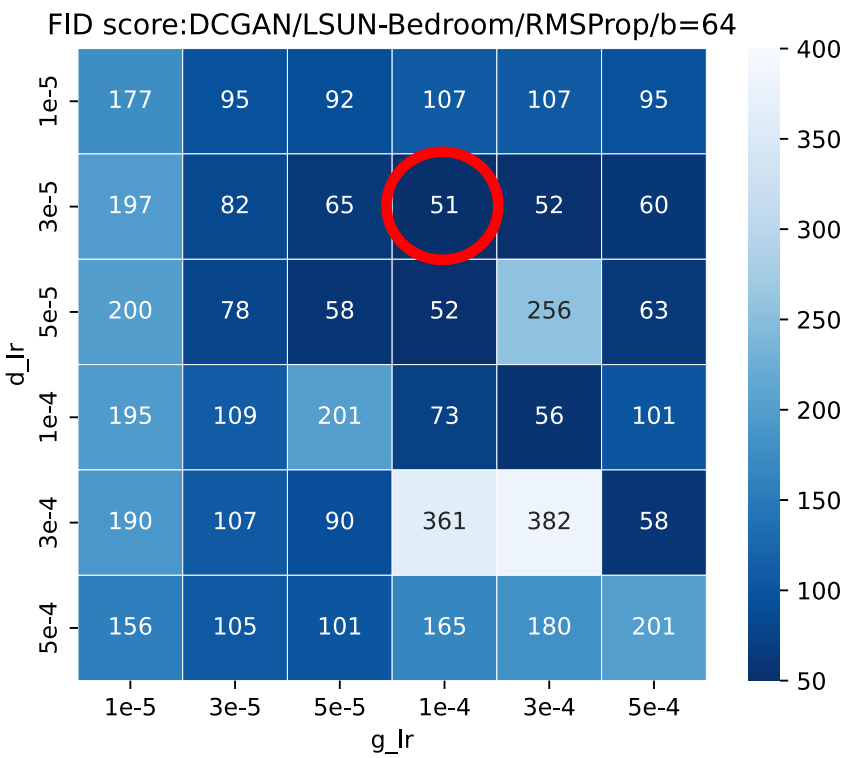
Numerical Results

【Relationship between b and N】

▷Training DCGAN on the LSUN-Bedroom dataset





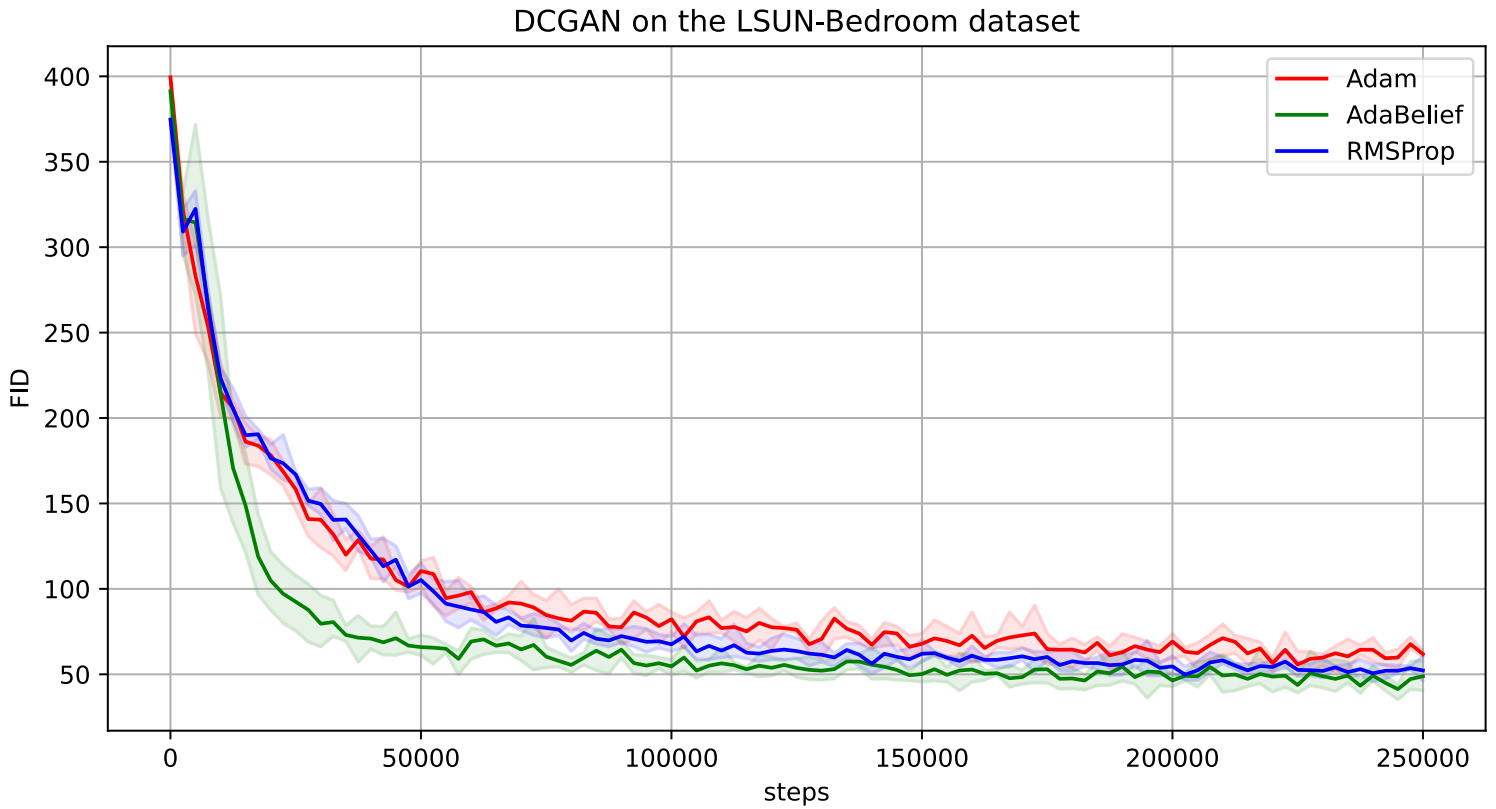


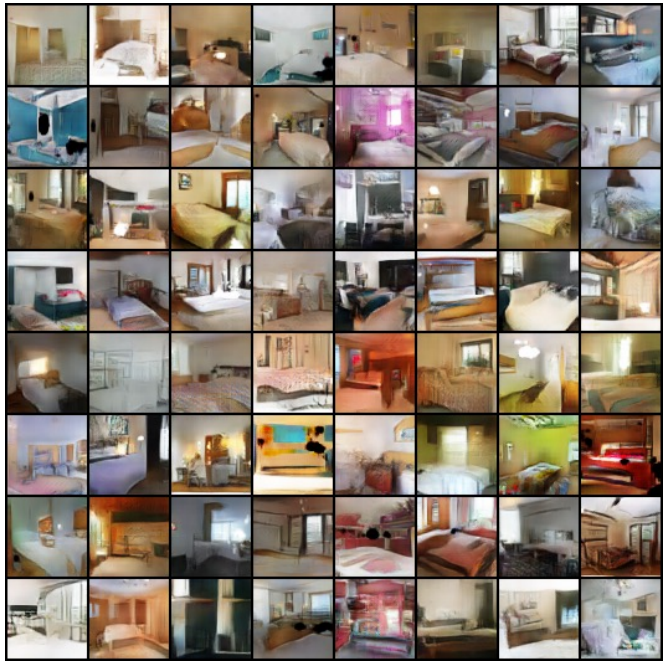
Numerical Results

【Relationship between b and N】

▷Training DCGAN on the Bedroom dataset

FID vs N graph





Generated images by Adam (FID=54)

Numerical Results

【Relationship between b and N】

$$N_G(b) = \frac{A_G b}{(\epsilon_G^2 - C_G)b - B_G}$$

- From our theory, $N_G(b)$ is monotone decreasing and convex with respect to b.
- $N_G(b)$ is the number of steps required for training to converge.

≈

$N_G(b)$ is the number of steps required to achieve a sufficiently low FID.

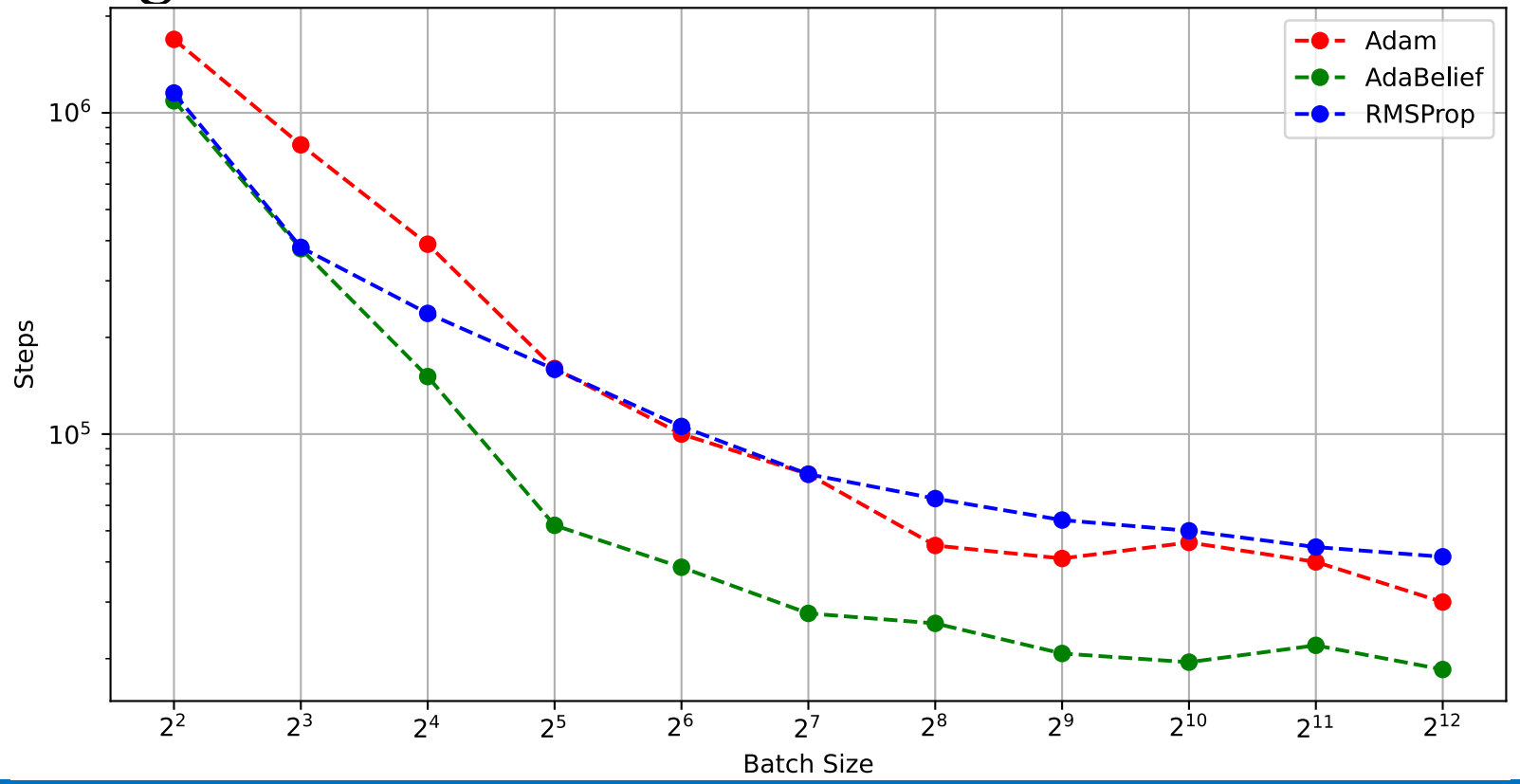
≈

$N_G(b)$ is the number of steps required to achieve FID ≦ 70.

Numerical Results

【Relationship between b and N】

▷Training DCGAN on the LSUN-Bedroom dataset

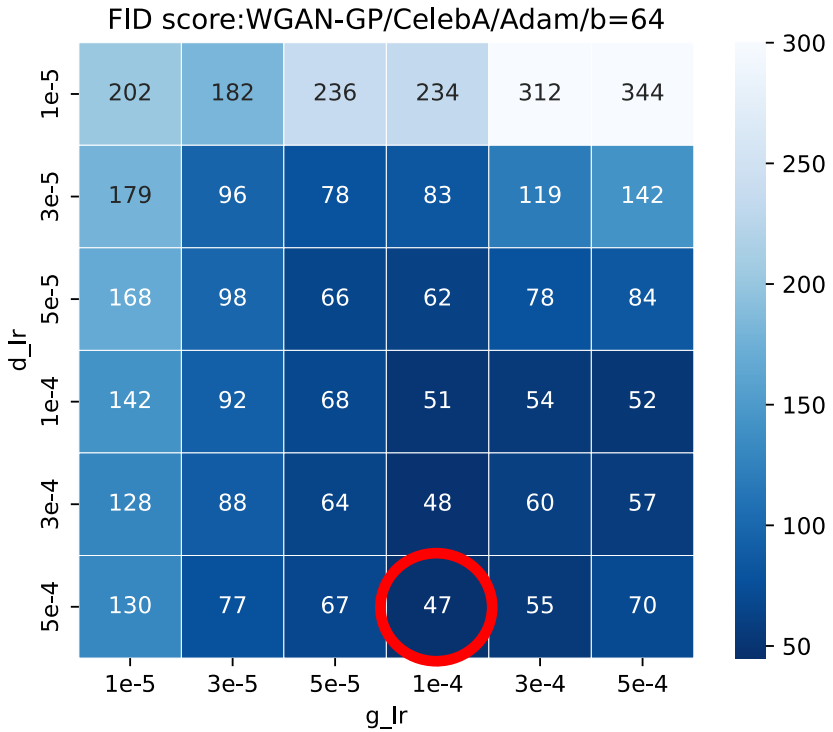


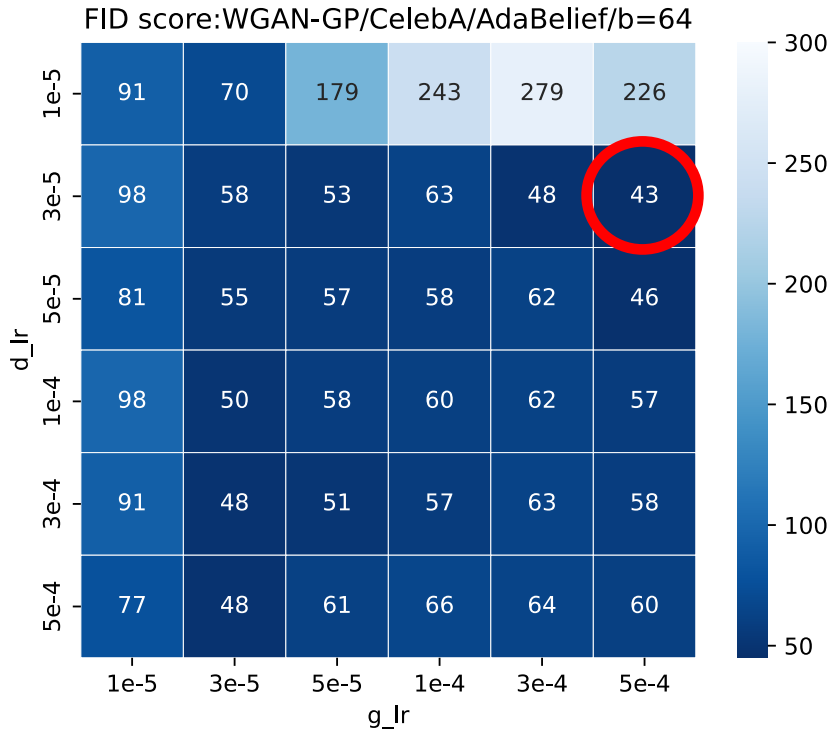
N vs b graph

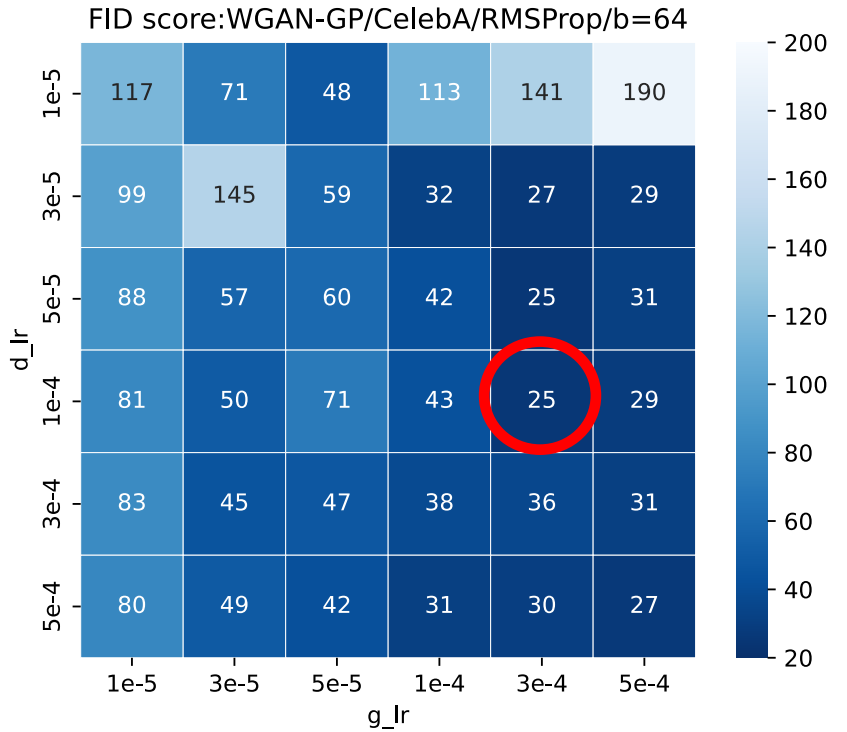
Numerical Results

【Relationship between b and N】

▷Training WGAN-GP on the CelebA dataset





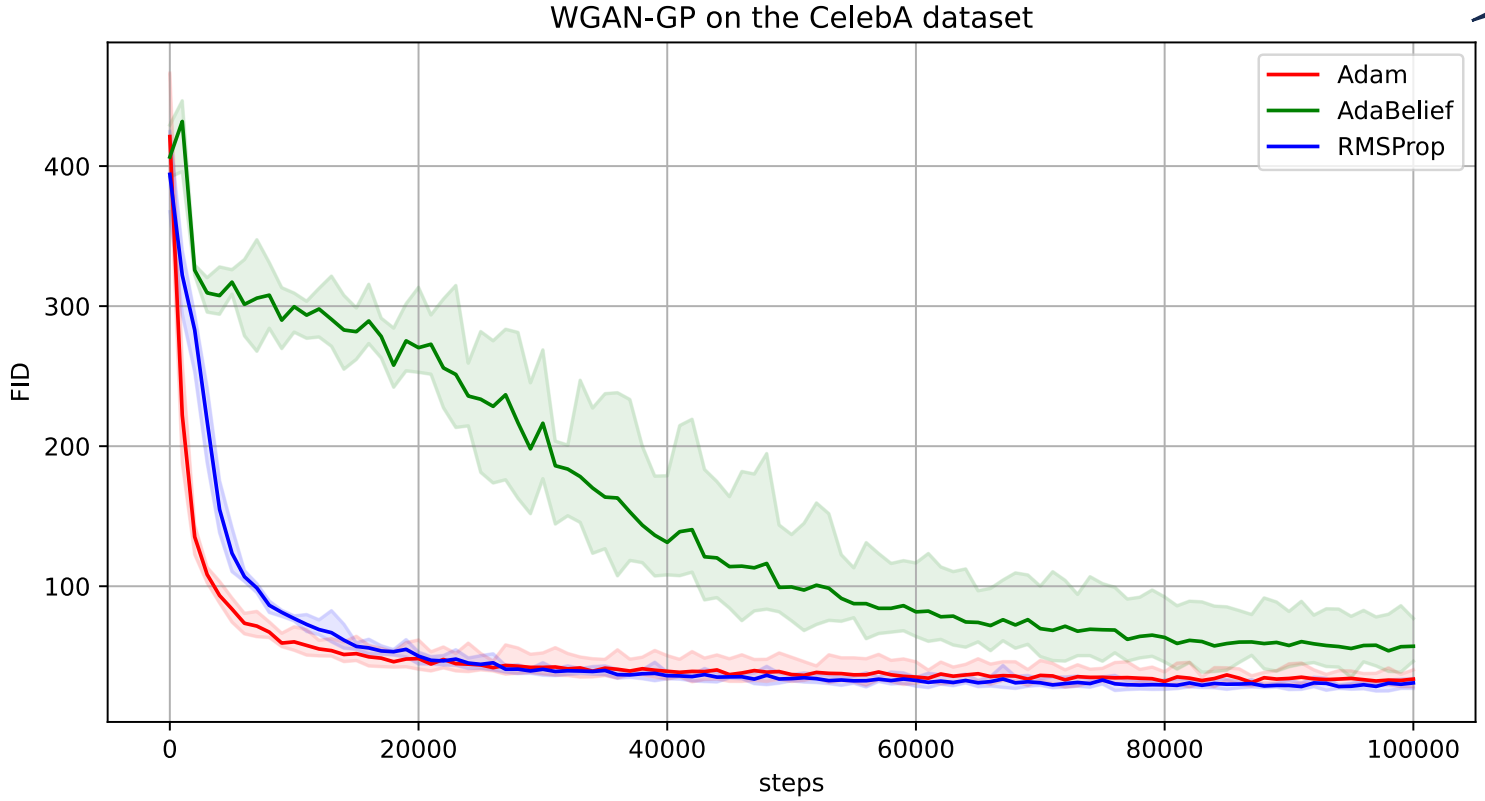


Numerical Results

【Relationship between b and N】

FID vs N graph

▷Training WGAN-GP on the CelebA dataset

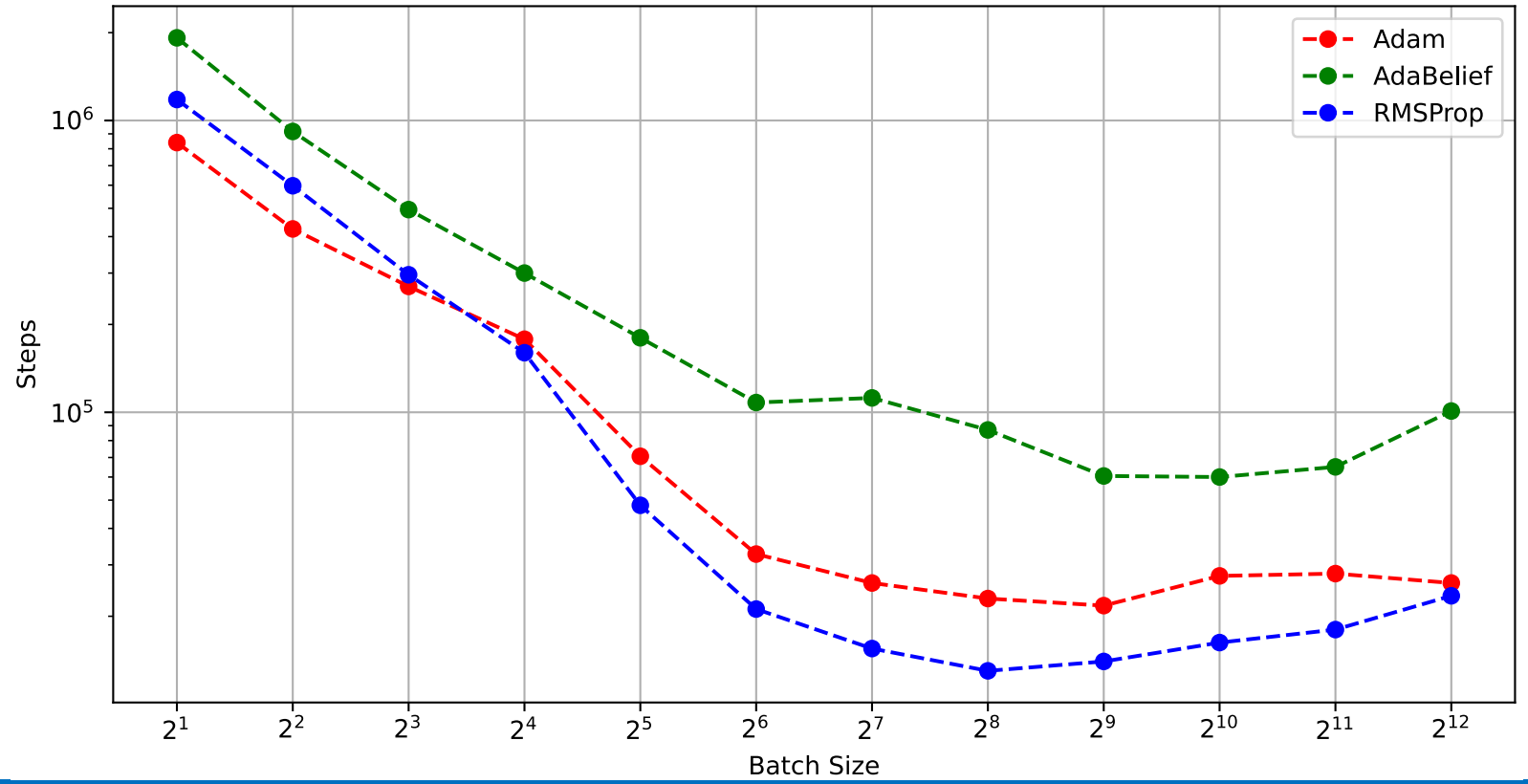


Generated images by Adam
(FID=40)

Numerical Results

【Relationship between b and N】

▷Training DCGAN on the LSUN-Bedroom dataset



N vs b graph

Numerical Results

【 Existence of a Critical Batch Size 】

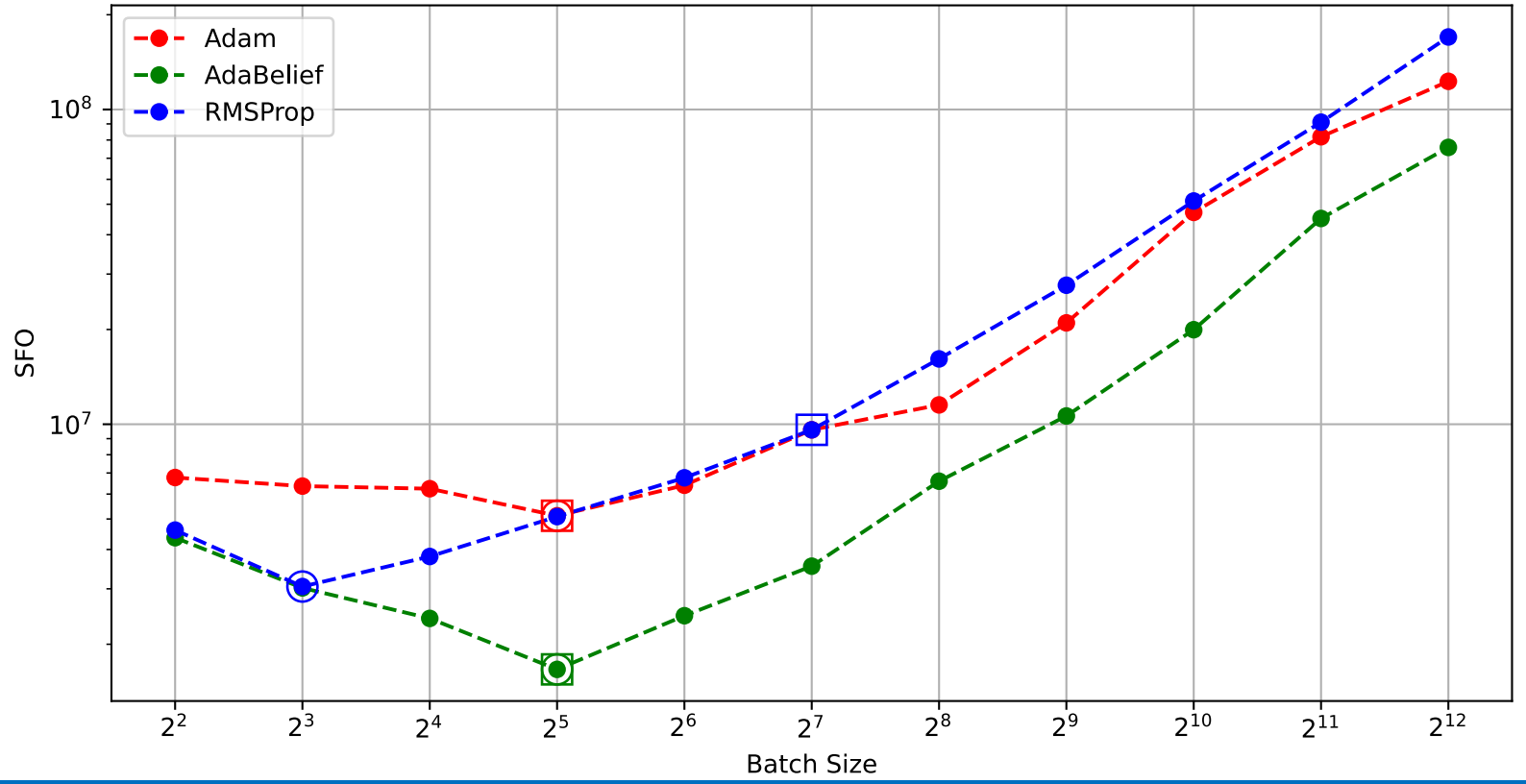
$$N_G(b)b = \frac{A_G b^2}{(\epsilon_G^2 - C_G)b - B_G}$$

- From our theory, SFO can be defined by $N_G(b)b$.
- Also, $N_G(b)b$ is a convex function.

Numerical Results

【 Existence of a Critical Batch Size 】

▷ Training DCGAN on the LSUN-Bedroom dataset



▷ Measured
- ▷ Adam: 2^5
- ▷ AdaBelief: 2^5
- ▷ RMSProp: 2^3

Nb vs b graph

Numerical Results

【 Estimation of the Critical Batch Size 】

▷From our theory, Adam's the lower bound of critical batch size can be expressed as follows

$$b_G^\star \geq \frac{\sigma_G^2}{\epsilon_G^3} \frac{\alpha^G}{(1-\beta_1^G)^3 \sqrt{\frac{\Theta}{1-\beta_2^G} \frac{1}{|\mathcal{S}|^2}}}$$

$$= \frac{\sigma_G^2}{\epsilon_G^3} \times \frac{0.0001}{(1-0.5)^3 \times \sqrt{\frac{3576704}{1-0.999} \times \frac{1}{3033042}}}$$

▷From the measured critical batch size; 2^5, the σ_G^2/ϵ_G^3 can be back-calculated as follows

$$\sigma_G^2/\epsilon_G^3 \leq 788.7.$$

Numerical Results

【 Estimation of the Critical Batch Size 】

$$\sigma_G^2/\epsilon_G^3 \leq 788.7.$$

- Using this ratio, we can estimate other optimizer's critical batch size.
- For example, for AdaBelief in training DCGAN,

$$b_G^\star \geq 788.7 \times \frac{0.0003}{(1-0.5)^3 \times \sqrt{\frac{4\times 3576704}{1-0.999} \times \frac{1}{3033042}}}$$

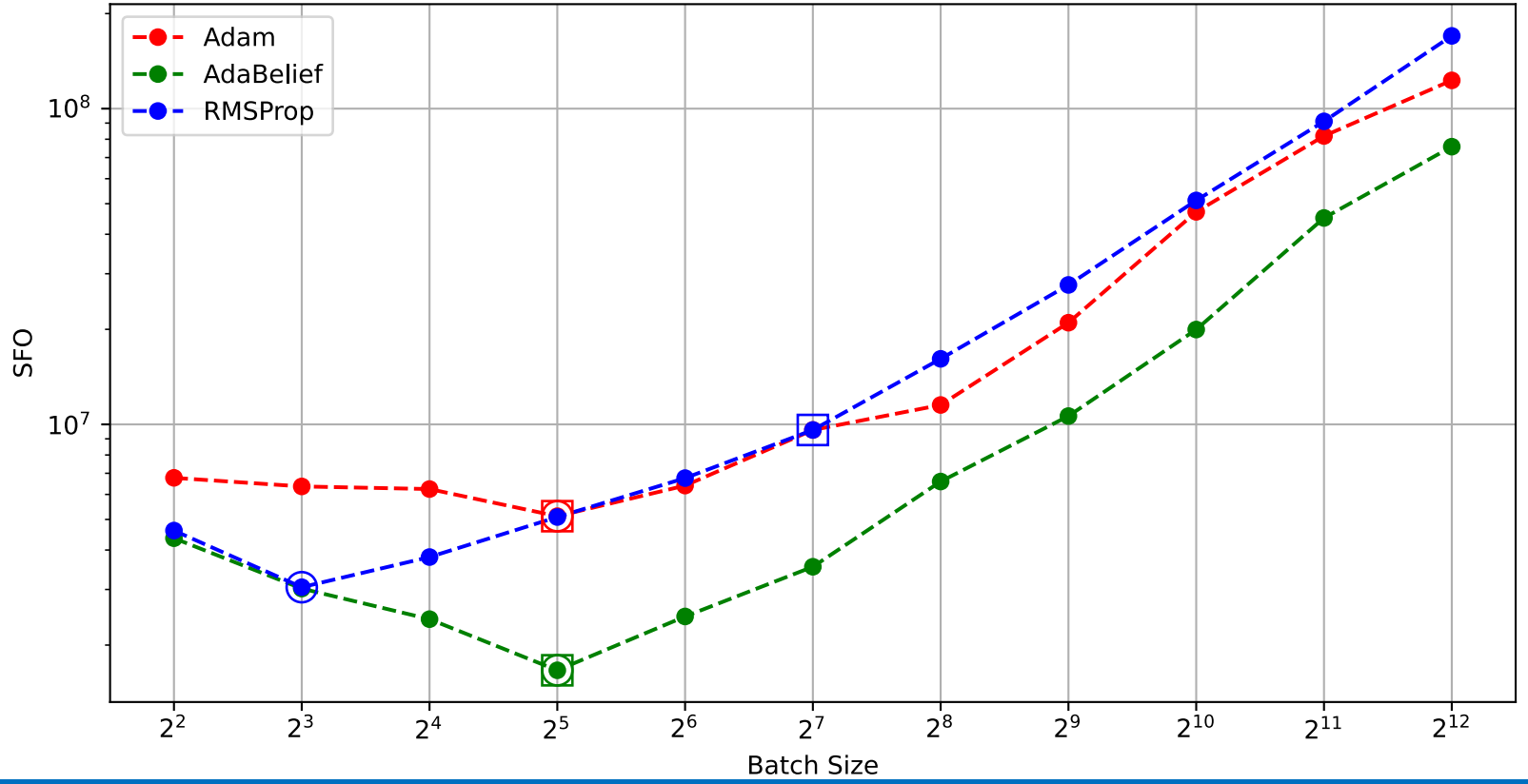
$$\approx 47.9$$

Numerical Results

【 Estimation of the Critical Batch Size 】

Nb vs b graph

▷Training DCGAN on the LSUN-Bedroom dataset



- ▷Measured
  - ▷Adam: 2^5
  - ▷AdaBelief: 2^5
  - ▷RMSProp: 2^3
- ▷Estimated
  - ▷Adam: 2^5
  - ▷AdaBelief: 2^5
  - ▷RMSProp: 2^7

Numerical Results

【 Estimation of the Critical Batch Size 】

$$\sigma_G^2/\epsilon_G^3 \leq 788.7.$$

- Using this ratio, we can estimate other optimizer's critical batch size.
- Moreover, this ratio can be appropriated for another GAN's training, as long as the model used is the same.
- For example, for Adam in training WGAN-GP on the CelebA dataset,

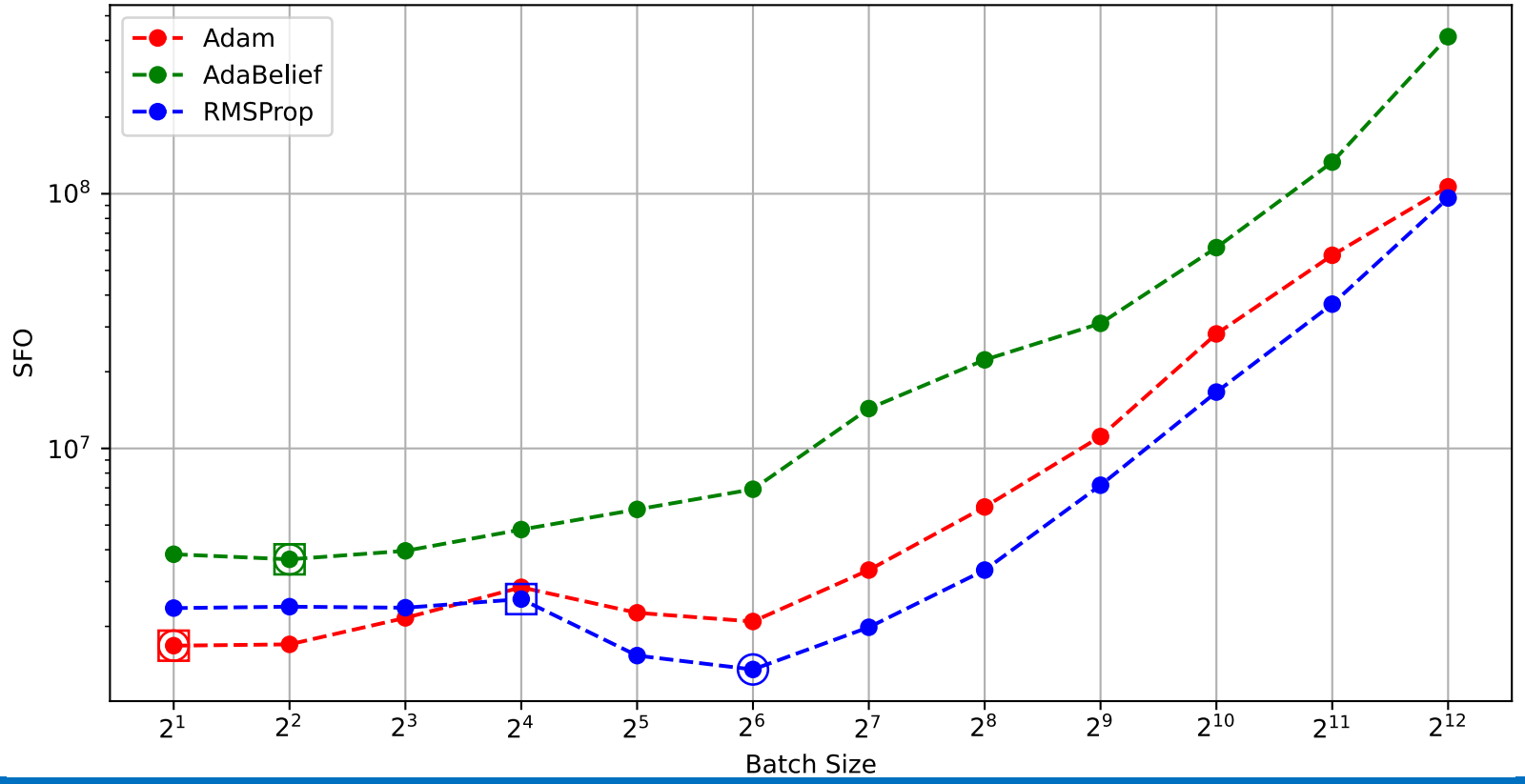
$$b_G^\star \geq 788.7 \times \frac{0.0001}{(1-0.5)^3 \times \sqrt{\frac{3576704}{1-0.999} \times \frac{1}{3033042}}}$$

$$\approx 1.7$$

Numerical Results

【 Estimation of the Critical Batch Size 】

▷ Training WGAN-GP on the CelebA dataset



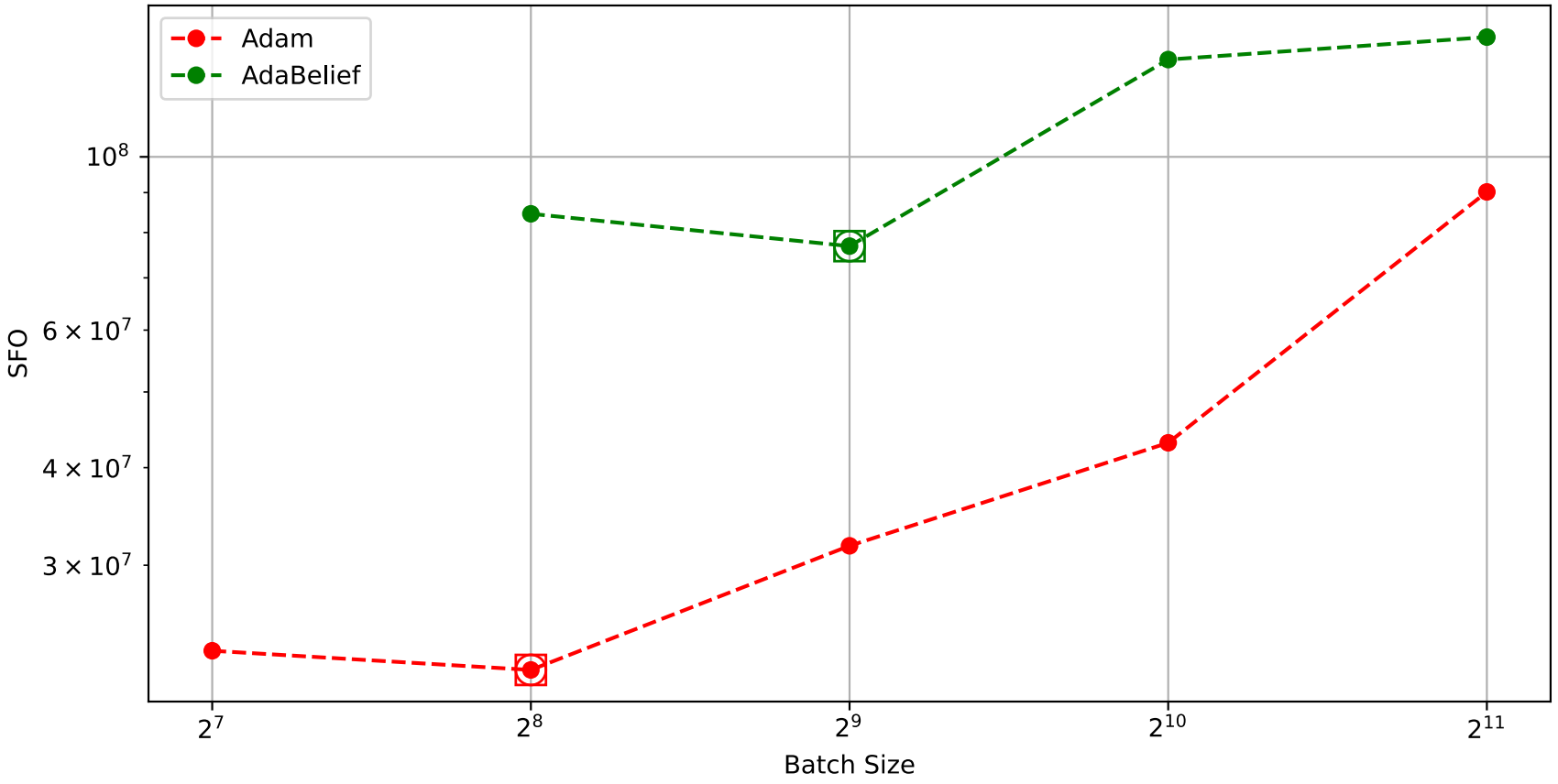
Nb vs b graph

- ▷ Measured
  - ▷ Adam: 2^1
  - ▷ AdaBelief: 2^2
  - ▷ RMSProp: 2^6
- ▷ Estimated
  - ▷ Adam: 2^1
  - ▷ AdaBelief: 2^2
  - ▷ RMSProp: 2^4

Numerical Results

【Training BigGAN on the ImageNet dataset】

Nb vs b graph



- ▷Measured
  - ▷Adam: 2^8
  - ▷AdaBelief: 2^9
- ▷Estimated
  - ▷Adam: 2^8
  - ▷AdaBelief: 2^9

Conclusion

- We performed a theoretical analysis of TTUR with constant learning rates.
- We examined the relationship between N and b .
- We showed that there is a critical batch size minimizing the SFO complexity.
- We also showed that we can estimate the critical batch size specific to the model-dataset-optimizer combination.
- Our experiments showed that the estimated and measured critical batch size are close.