

Existence and Estimation of Critical Batch Size for Training GANs with Two Time-Scale Update Rule

Naoki Sato¹, Hideaki Iiduka¹

¹Meiji University

Aug.28, 15:50-16:20



Prior knowledge

【I. mini-batch size】

- ▷ Training datasets $S := (z_1, z_2, \dots, z_n)$
- ▷ Deep Neural Networks' parameter $\theta \in \mathbb{R}^\Theta$
- ▷ Empirical risk minimization problem (経験損失最小化問題)

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

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

Prior knowledge

【I. mini-batch size】

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

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

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

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

Prior knowledge

【I. mini-batch size】

▷ Gradient Descent Method (最急降下法)

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

全勾配

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

ミニバッチ
確率的勾配

▷ Stochastic Gradient Descent

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

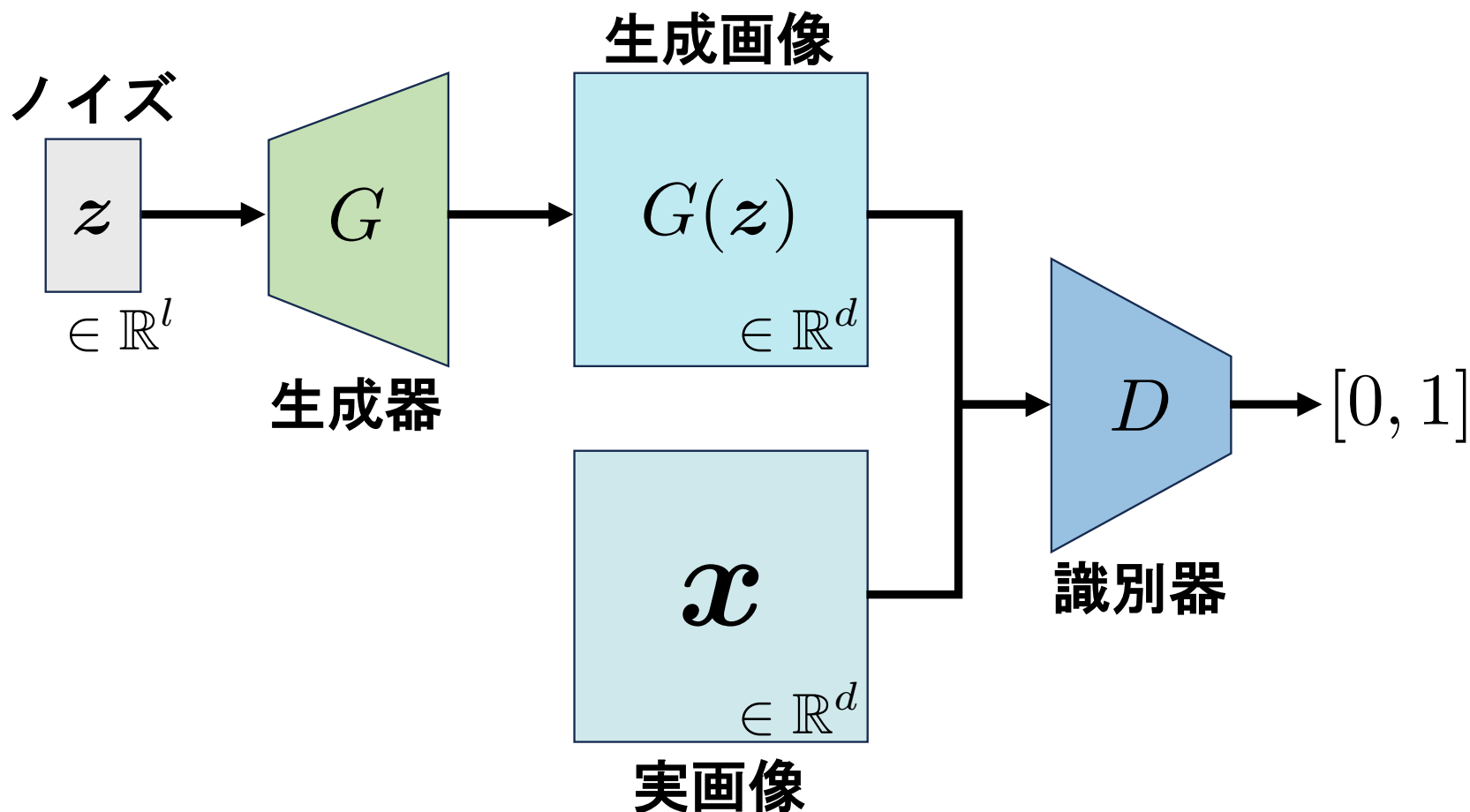
確率的勾配

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

mini-batch size

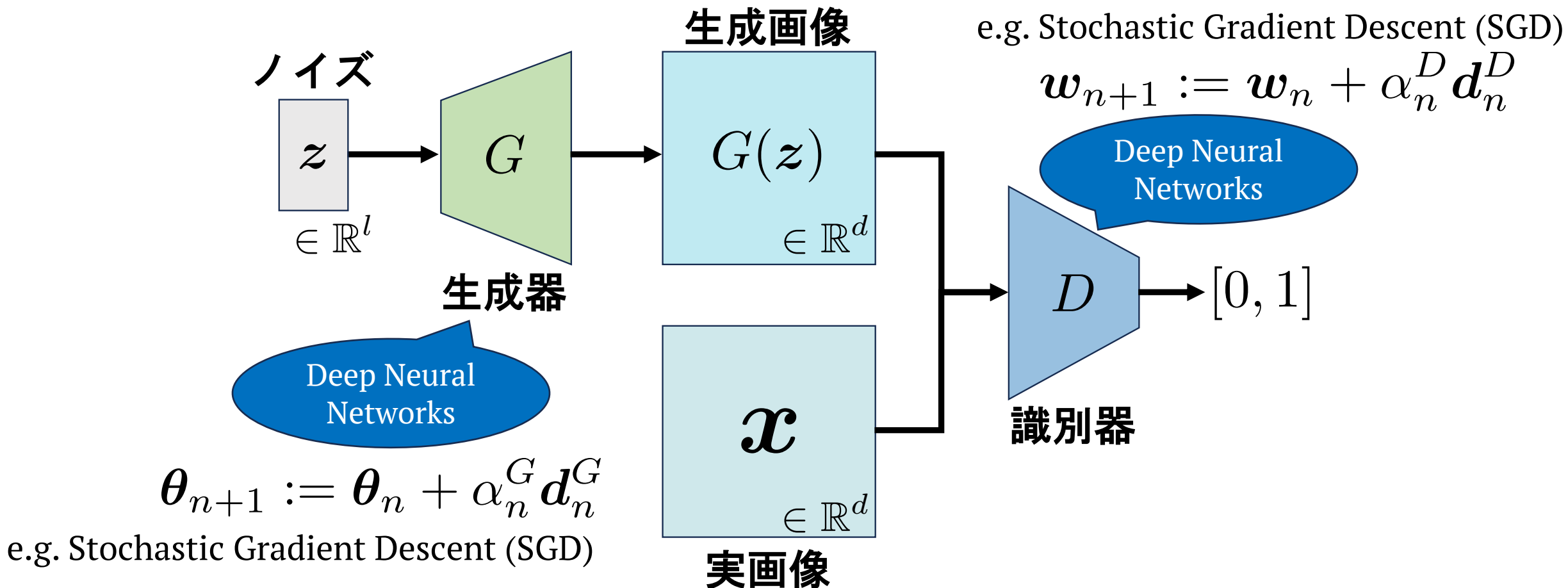
Prior knowledge

【II. Generative Adversarial Networks (GANs)】



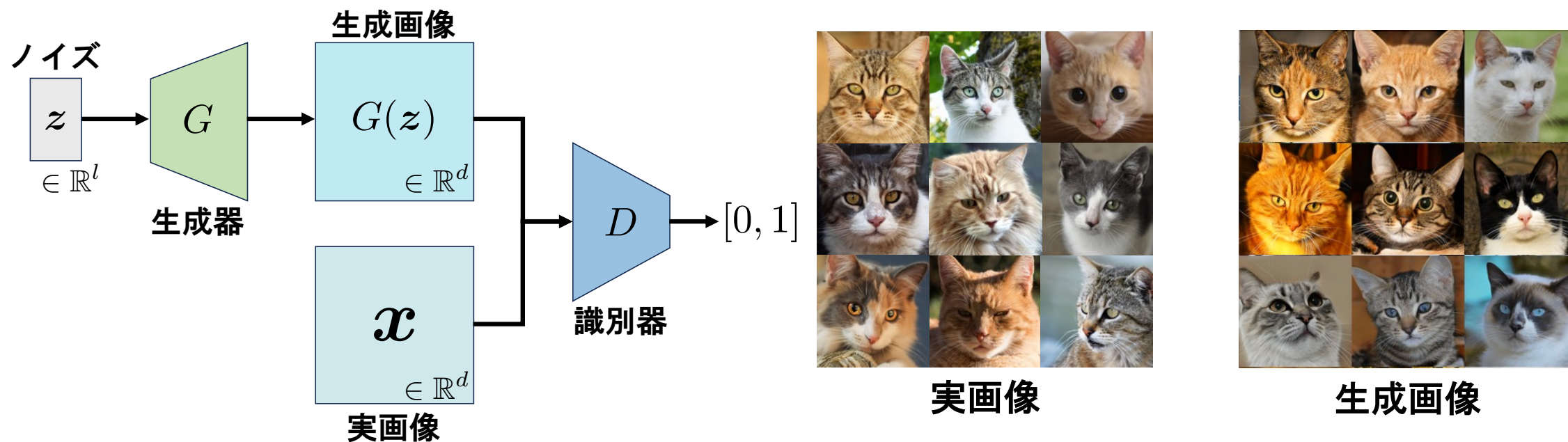
Prior knowledge

【II. Generative Adversarial Networks (GANs)】



Prior knowledge

【II. Generative Adversarial Networks (GANs)】



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

Prior knowledge

【III. Fréchet Inception Distance (FID)】



実画像(FID=0)



生成画像(FID=287)



生成画像(FID=12)

Introduction

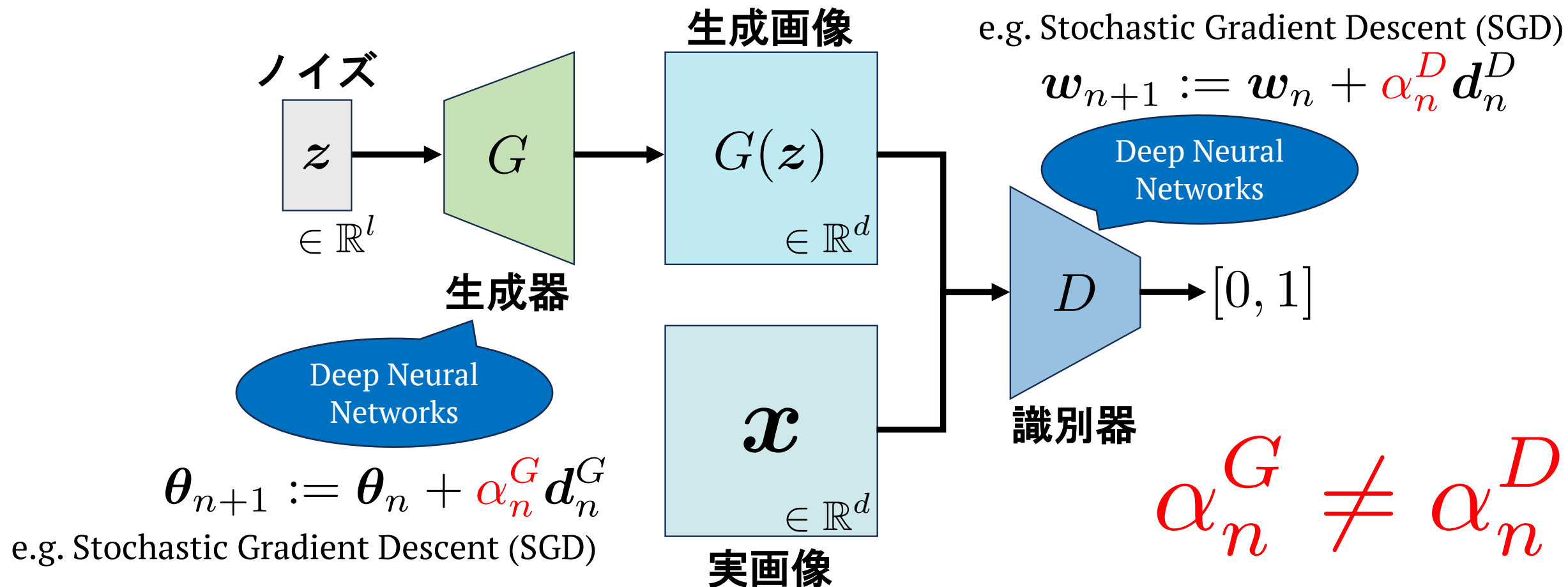
【Background】

- ▷ [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 $\mathbf{z}^{(i)}$.

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

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

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

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

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

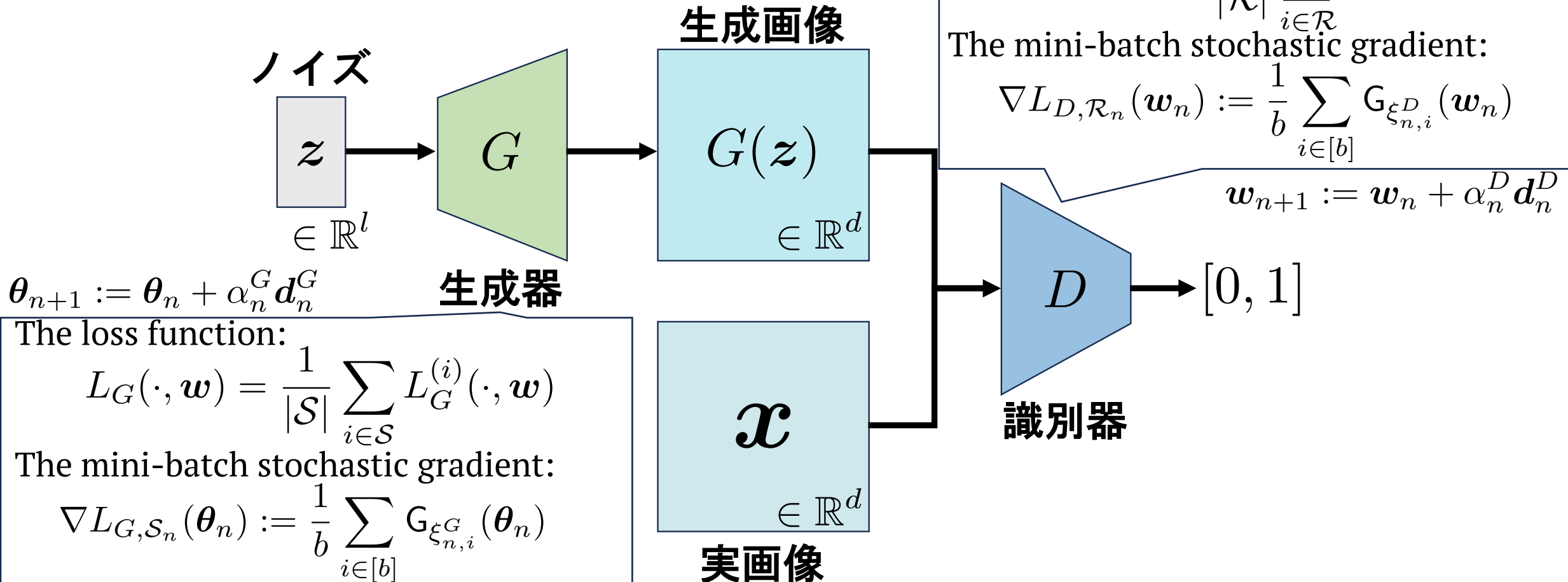
ξ^G : A random variable that does not depend on $\mathbf{w} \in \mathbb{R}^W$ and $\boldsymbol{\theta} \in \mathbb{R}^\Theta$.

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

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

Mathematical Preliminaries

【Definition】



Mathematical Preliminaries

【Assumptions】

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

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

(i) For each iteration n ,

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

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

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

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 .

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

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

Mathematical Preliminaries

【Problems】

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

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



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

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

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

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

Mathematical Preliminaries

【Problem 1】

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

satisfying

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

$$\langle \theta^* - \theta, \nabla_\theta L_G(\theta^*, w^*) \rangle \leq 0,$$

$$\langle w^* - w, \nabla_w L_D(\theta^*, w^*) \rangle \leq 0.$$

【Algorithms】

▷ We use Adam, AdaBelief and RMSProp.

Algorithm 1 Adaptive Method for Solving Problem (1)

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

1: $n \leftarrow 0, (\theta_0, w_0) \in \mathbb{R}^\Theta \times \mathbb{R}^W, m_{-1}^G = \mathbf{0} \in \mathbb{R}^\Theta, m_{-1}^D = \mathbf{0} \in \mathbb{R}^W$

2: **loop**

3: **loop**

4: $m_n^G := \beta_1^G m_{n-1}^G + (1 - \beta_1^G) \nabla L_{G, S_n}(\theta_n)$

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

6: $H_n^G \in \mathbb{S}_{++}^\Theta \cap \mathbb{D}^\Theta$

7: Find $d_n^G \in \mathbb{R}^\Theta$ that solves $H_n^G d = -\hat{m}_n^G$

8: $\theta_{n+1} := \theta_n + \alpha_n^G d_n^G$

9: **end loop**

10: **loop**

11: $m_n^D := \beta_1^D m_{n-1}^D + (1 - \beta_1^D) \nabla L_{D, \mathcal{R}_n}(w_n)$

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

13: $H_n^D \in \mathbb{S}_{++}^W \cap \mathbb{D}^W$

14: Find $d_n^D \in \mathbb{R}^W$ that solves $H_n^D d = -\hat{m}_n^D$

15: $w_{n+1} := w_n + \alpha_n^D d_n^D$

16: **end loop**

17: $n \leftarrow n + 1$

18: **end loop**

Theoretical Analysis

【Convergence Analysis】

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

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \mathbb{E} [\langle \boldsymbol{\theta}_n - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n) \rangle] &\leq \underbrace{\frac{\Theta \text{Dist}(\boldsymbol{\theta}) H^G}{2\alpha^G \tilde{\beta}_1^G}}_{A_G} \frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G}}_{B_G} \frac{1}{b} + \underbrace{\frac{M_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G} + \frac{\beta_1^G}{\tilde{\beta}_1^G} \sqrt{\Theta \text{Dist}(\boldsymbol{\theta}) (\sigma_G^2 + M_G^2)}}_{C_G} \\ \frac{1}{N} \sum_{n=1}^N \mathbb{E} [\langle \boldsymbol{w}_n - \boldsymbol{w}, \nabla_{\boldsymbol{w}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n) \rangle] &\leq \underbrace{\frac{W \text{Dist}(\boldsymbol{w}) H^D}{2\alpha^D \tilde{\beta}_1^D}}_{A_D} \frac{1}{N} + \underbrace{\frac{\sigma_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D}}_{B_D} \frac{1}{b} + \underbrace{\frac{M_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D} + \frac{\beta_1^D}{\tilde{\beta}_1^D} \sqrt{W \text{Dist}(\boldsymbol{w}) (\sigma_D^2 + M_D^2)}}_{C_D} \end{aligned}$$

Theoretical Analysis

【Relationship between b and N 】

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

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \mathbb{E} [\langle \boldsymbol{\theta}_n - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n) \rangle] &\leq \underbrace{\frac{\Theta \text{Dist}(\boldsymbol{\theta}) H^G}{2\alpha^G \tilde{\beta}_1^G}}_{A_G} \frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G}}_{B_G} \frac{1}{b} + \underbrace{\frac{M_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G} + \frac{\beta_1^G}{\tilde{\beta}_1^G} \sqrt{\Theta \text{Dist}(\boldsymbol{\theta}) (\sigma_G^2 + M_G^2)}}_{C_G} \\ &\rightarrow \text{small} \\ \frac{1}{N} \sum_{n=1}^N \mathbb{E} [\langle \boldsymbol{w}_n - \boldsymbol{w}, \nabla_{\boldsymbol{w}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n) \rangle] &\leq \underbrace{\frac{W \text{Dist}(\boldsymbol{w}) H^D}{2\alpha^D \tilde{\beta}_1^D}}_{A_D} \frac{1}{N} + \underbrace{\frac{\sigma_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D}}_{B_D} \frac{1}{b} + \underbrace{\frac{M_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D} + \frac{\beta_1^D}{\tilde{\beta}_1^D} \sqrt{W \text{Dist}(\boldsymbol{w}) (\sigma_D^2 + M_D^2)}}_{C_D} \\ &\rightarrow \text{small} \end{aligned}$$

Theoretical Analysis

【Relationship between b and N 】

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

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \mathbb{E} [\langle \boldsymbol{\theta}_n - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n) \rangle] &\leq \underbrace{\frac{\Theta \text{Dist}(\boldsymbol{\theta}) H^G}{2\alpha^G \tilde{\beta}_1^G}}_{A_G} \frac{1}{N} + \underbrace{\frac{\sigma_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G}}_{B_G} \frac{1}{b} + \underbrace{\frac{M_G^2 \alpha^G}{2\tilde{\beta}_1^G \tilde{\gamma}^{G^2} h_{0,*}^G} + \frac{\beta_1^G}{\tilde{\beta}_1^G} \sqrt{\Theta \text{Dist}(\boldsymbol{\theta}) (\sigma_G^2 + M_G^2)}}_{C_G} \\ &=: \epsilon_G^2 \\ \frac{1}{N} \sum_{n=1}^N \mathbb{E} [\langle \boldsymbol{w}_n - \boldsymbol{w}, \nabla_{\boldsymbol{w}} L_G(\boldsymbol{\theta}_n, \boldsymbol{w}_n) \rangle] &\leq \underbrace{\frac{W \text{Dist}(\boldsymbol{w}) H^D}{2\alpha^D \tilde{\beta}_1^D}}_{A_D} \frac{1}{N} + \underbrace{\frac{\sigma_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D}}_{B_D} \frac{1}{b} + \underbrace{\frac{M_D^2 \alpha^D}{2\tilde{\beta}_1^D \tilde{\gamma}^{D^2} h_{0,*}^D} + \frac{\beta_1^D}{\tilde{\beta}_1^D} \sqrt{W \text{Dist}(\boldsymbol{w}) (\sigma_D^2 + M_D^2)}}_{C_D} \\ &=: \epsilon_D^2 \end{aligned}$$

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,

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

i.e.

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

- ▷ 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^* minimizing $N(b)b$,

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

Theoretical Analysis

【Estimation of the Critical Batch Size】

▷ The lower bound of b^* can be expressed as follows

(i) for Adam,

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

(ii) for AdaBelief,

$$b_G^* \geq \frac{\sigma_G^2}{\epsilon_G^3} \frac{\alpha^G}{(1 - \beta_1^G)^3 \sqrt{\frac{4\Theta}{1 - \beta_2^G} \frac{1}{|S|^2}}}$$

(iii) for RMSProp,

$$b_G^* \geq \frac{\sigma_G^2}{\epsilon_G^3} \frac{\alpha^G}{\sqrt{\frac{\Theta}{|S|^2}}}$$

Theoretical Analysis

【Estimation of the Critical Batch Size】

▷ The lower bound of b^* can be expressed as follows

(i) for Adam,

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

【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^* can be expressed as follows

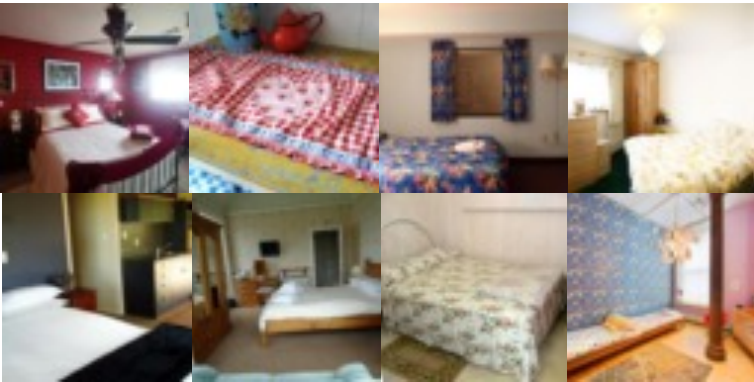
(i) for Adam,

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

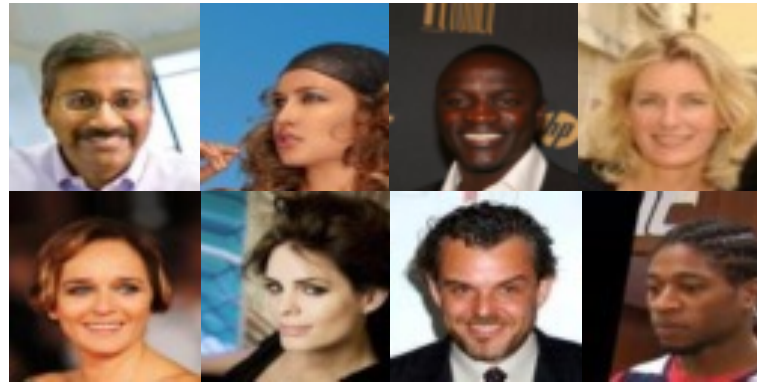
▷ This indicates that it is possible to estimate the critical batch size specific to the **model-dataset-optimizer** combination.

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 → $ S $	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**.

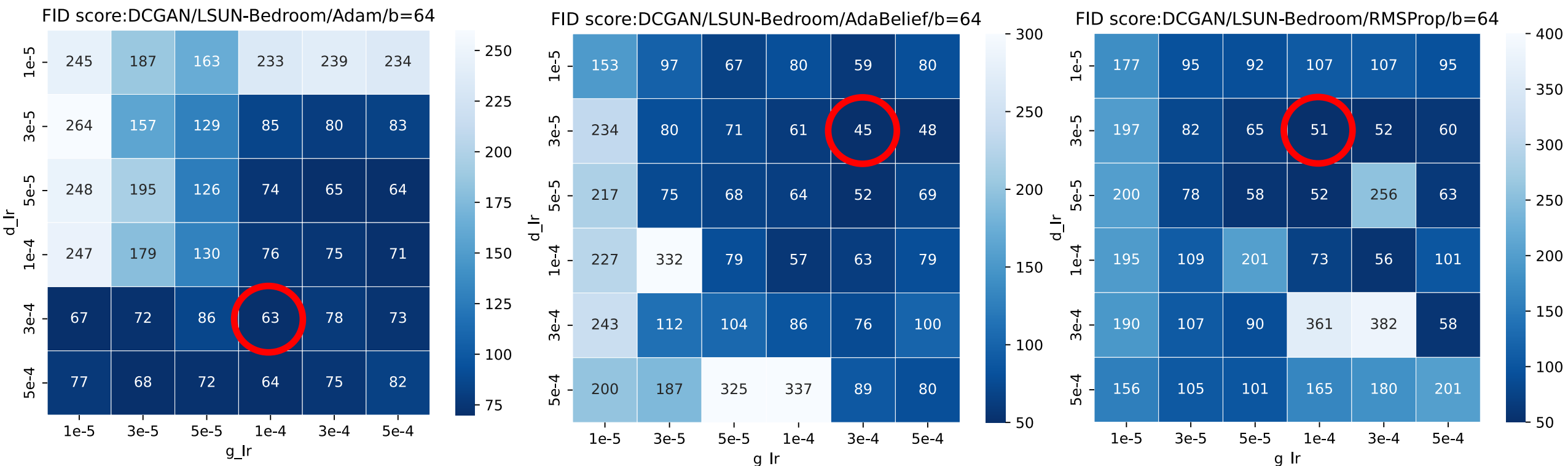
⋈

$N_G(b)$ 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

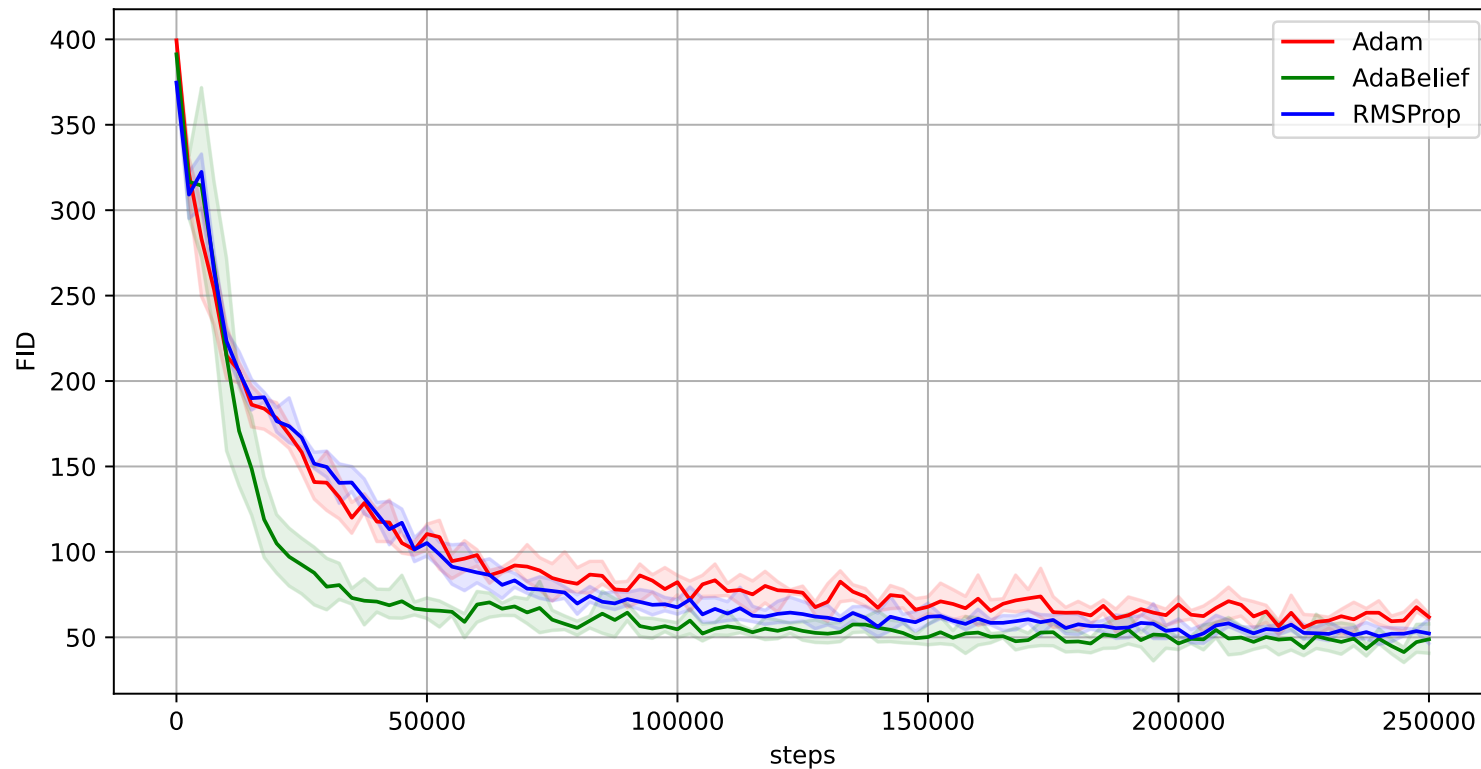


Numerical Results

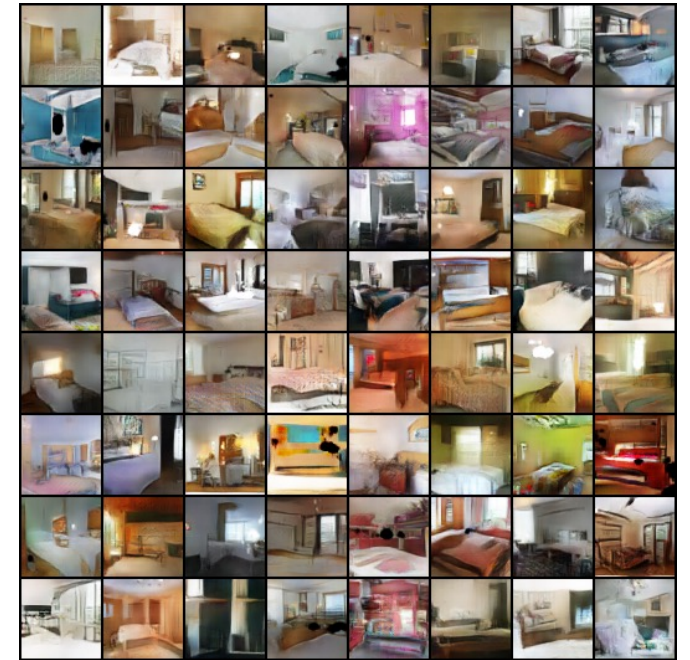
【Relationship between b and N 】

▷ Training DCGAN on the Bedroom dataset

DCGAN on the LSUN-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**.

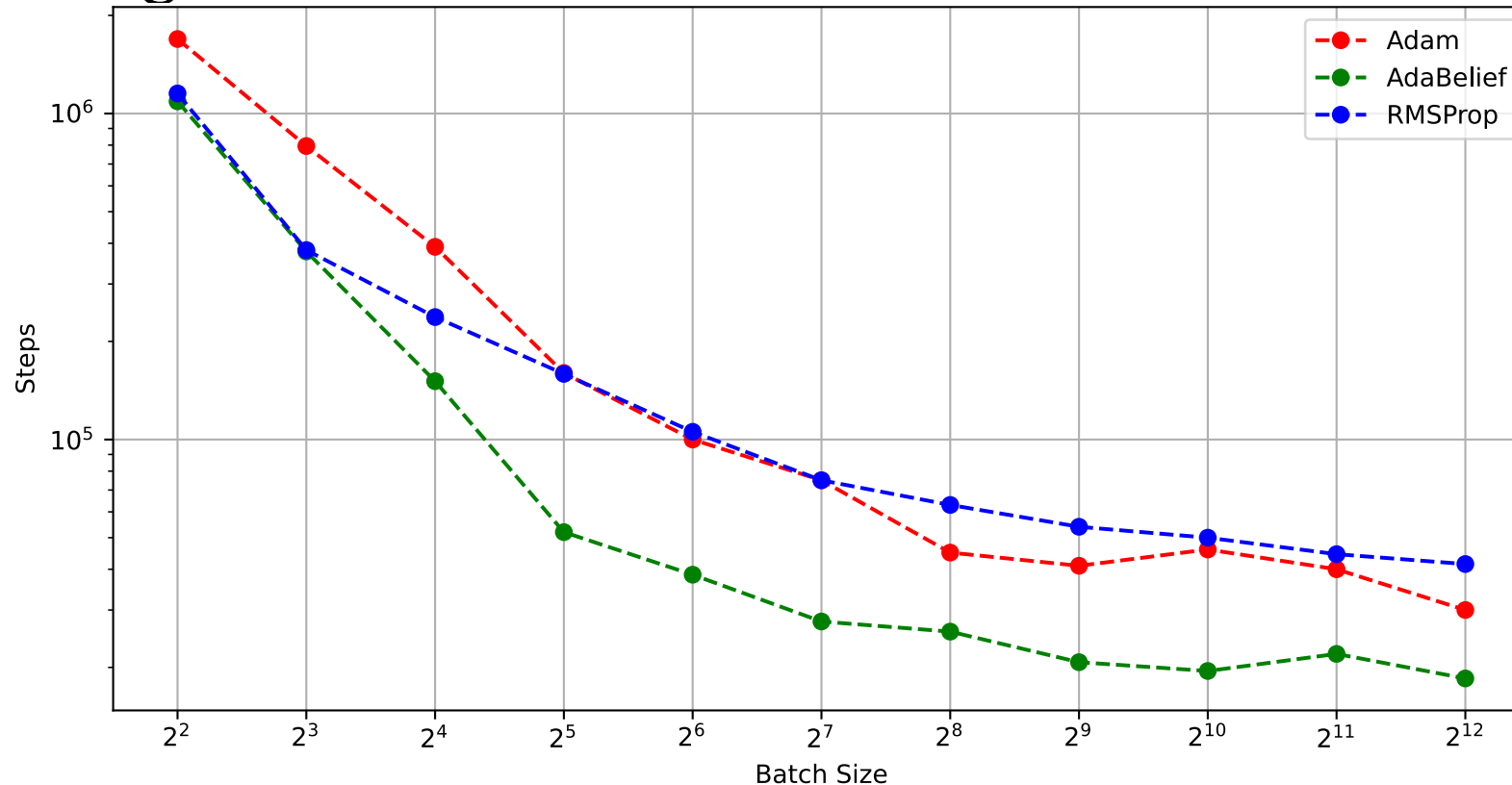
}}

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

Numerical Results

【Relationship between b and N 】

▷ Training DCGAN on the LSUN-Bedroom dataset

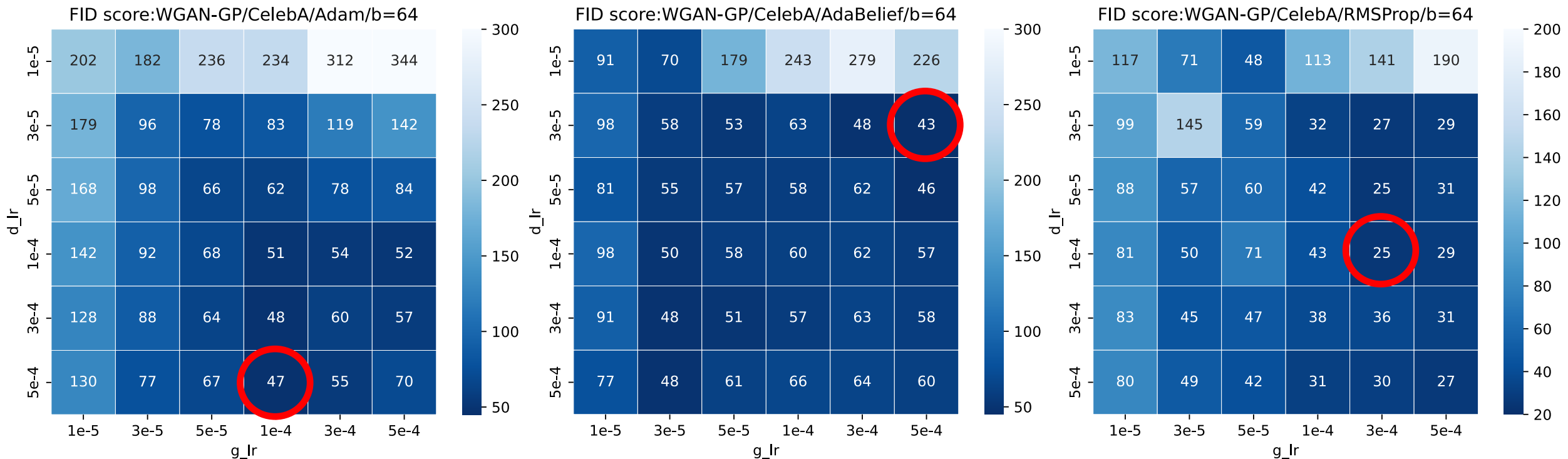


N vs b graph

Numerical Results

【Relationship between b and N 】

▷ Training WGAN-GP on the CelebA dataset

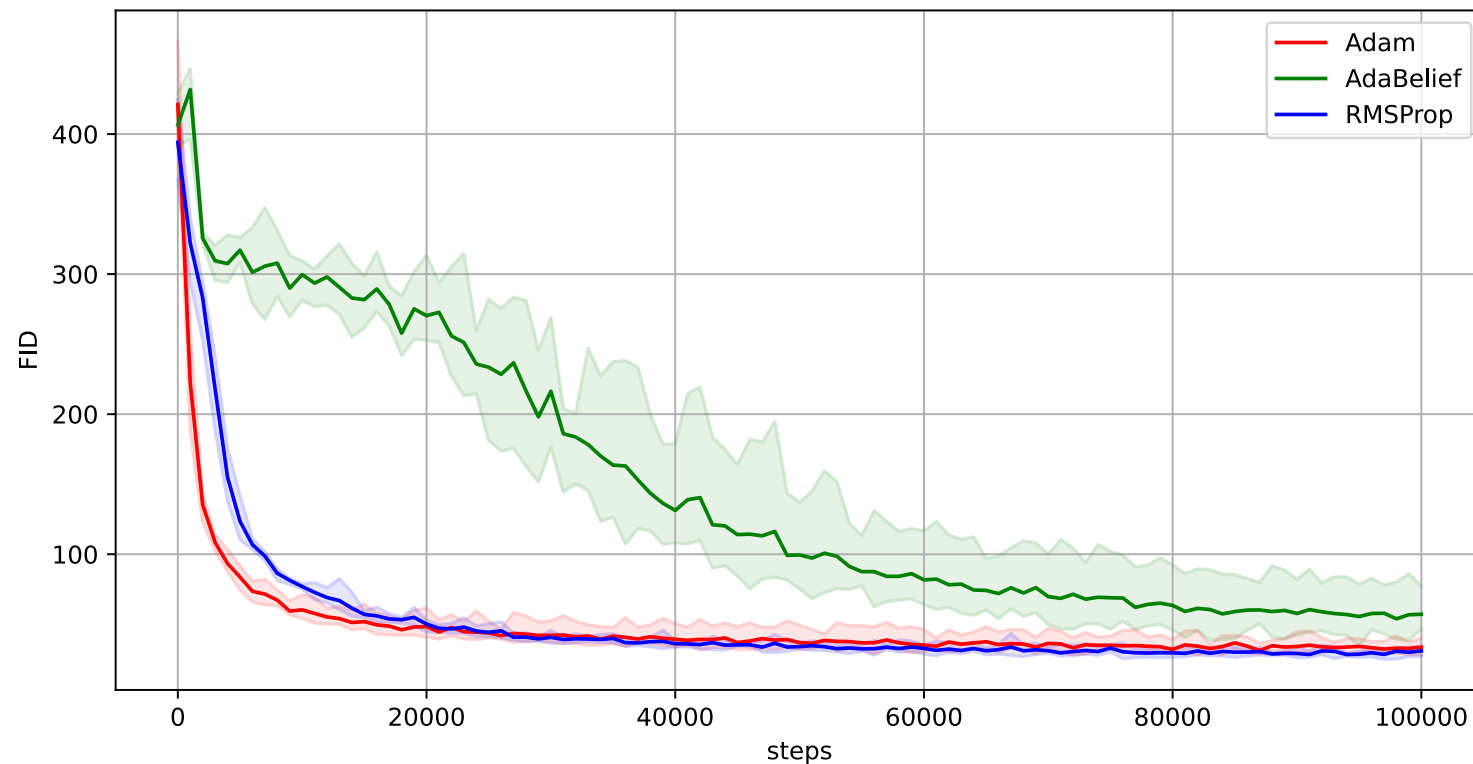


Numerical Results

【Relationship between b and N 】

▷ Training WGAN-GP on the CelebA dataset

WGAN-GP on the CelebA dataset



FID vs N graph

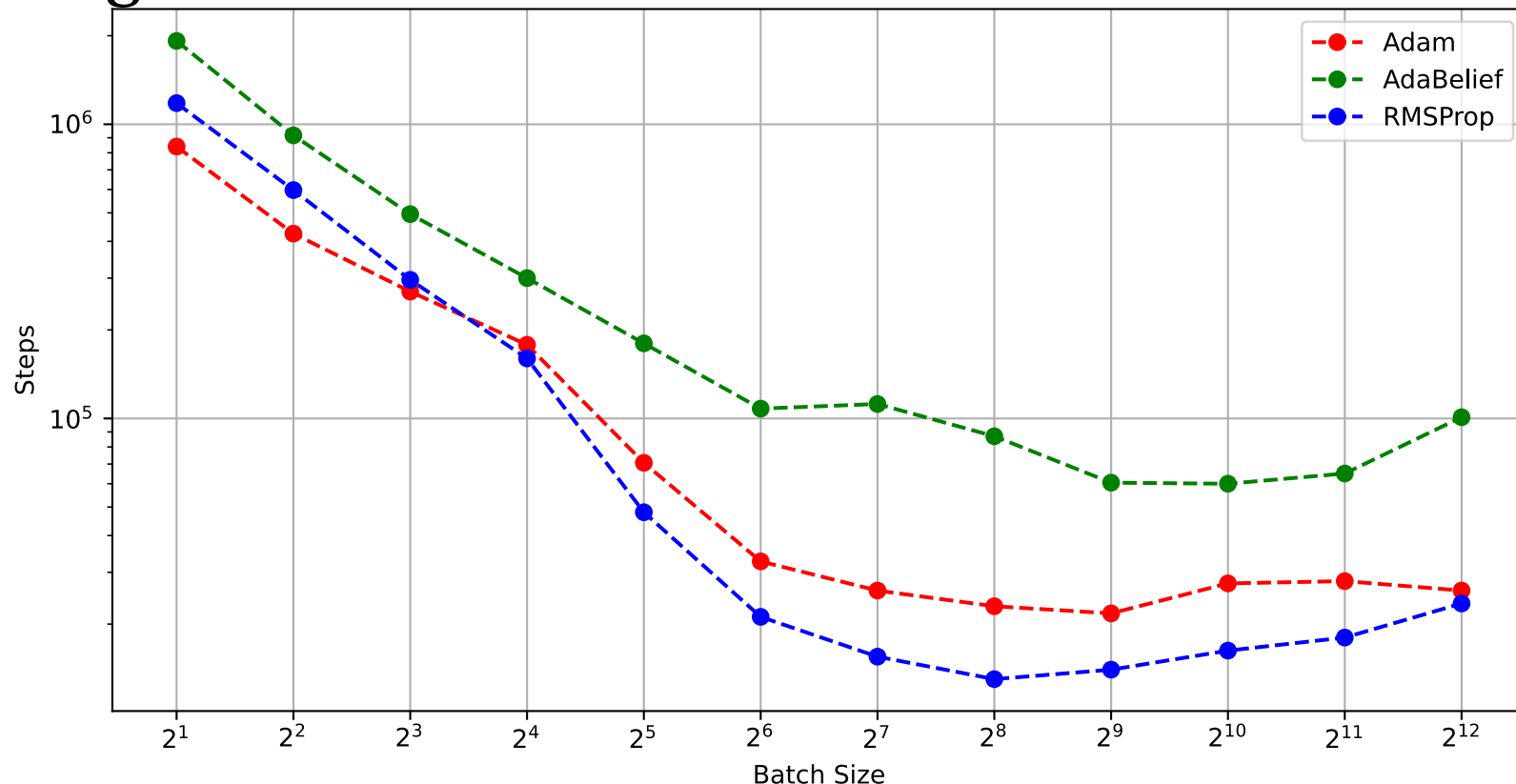


Generated images by Adam
(FID=40)

Numerical Results

【Relationship between b and N 】

▷ Training WGAN-GP on the CelebA dataset

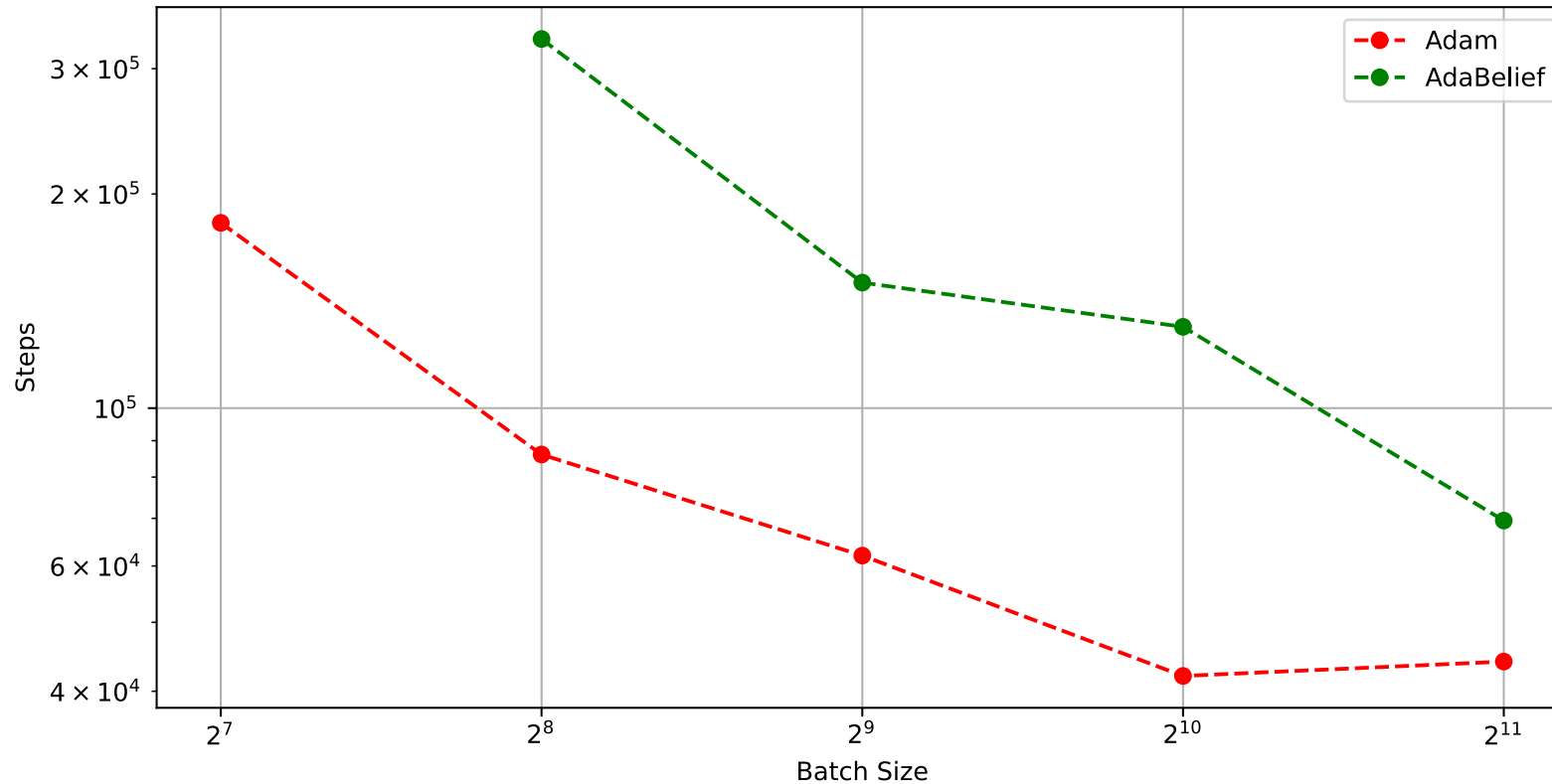


N vs b graph

Numerical Results

【Relationship between b and N 】

▷ Training BigGAN on the ImageNet dataset



N vs b graph

Numerical Results

【 Existence of a Critical Batch Size 】

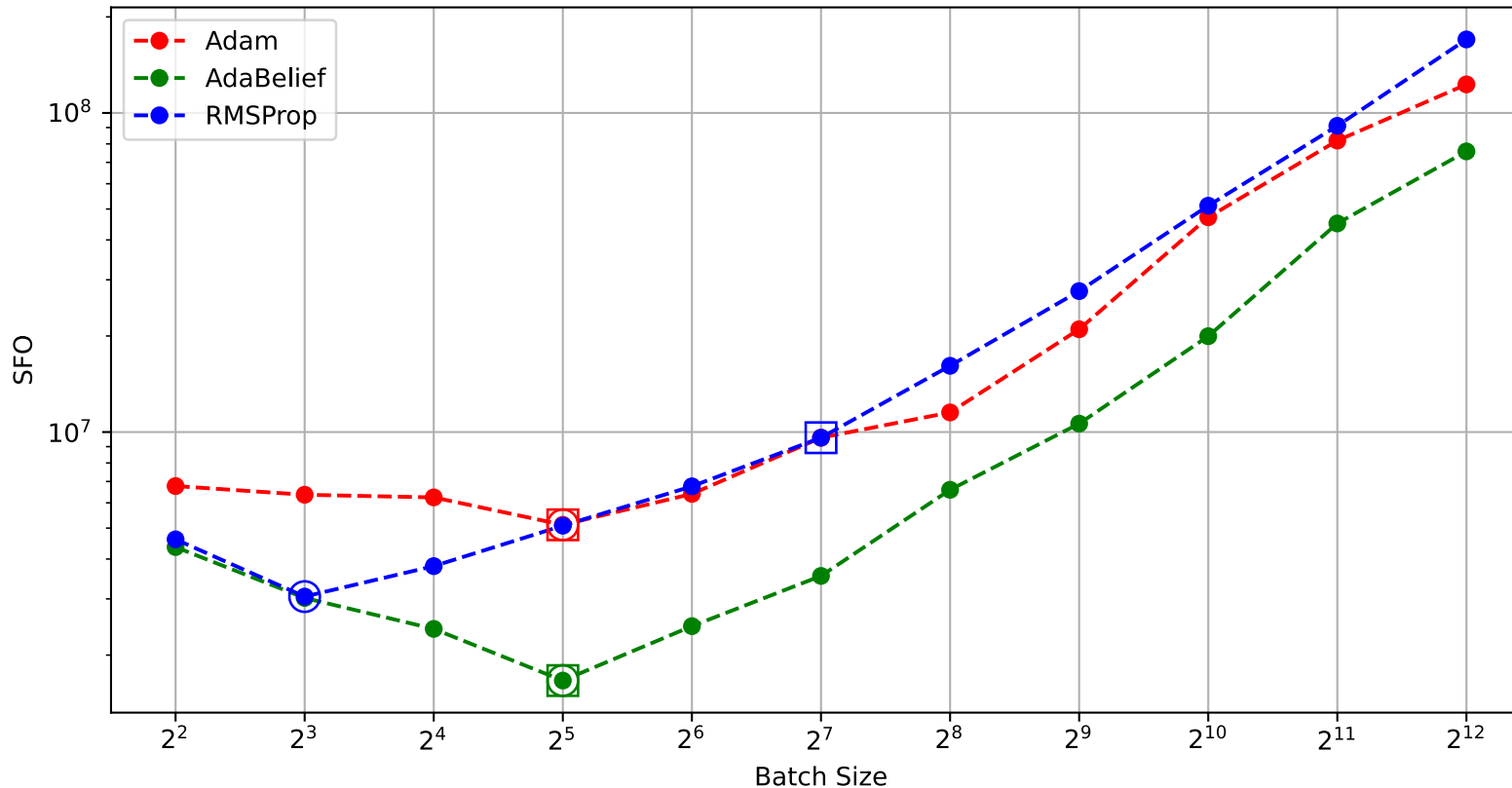
$$N_G(b)b = \frac{A_G b^2}{(\epsilon_G^2 - C_G)b - B_G}$$

- ▷ 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

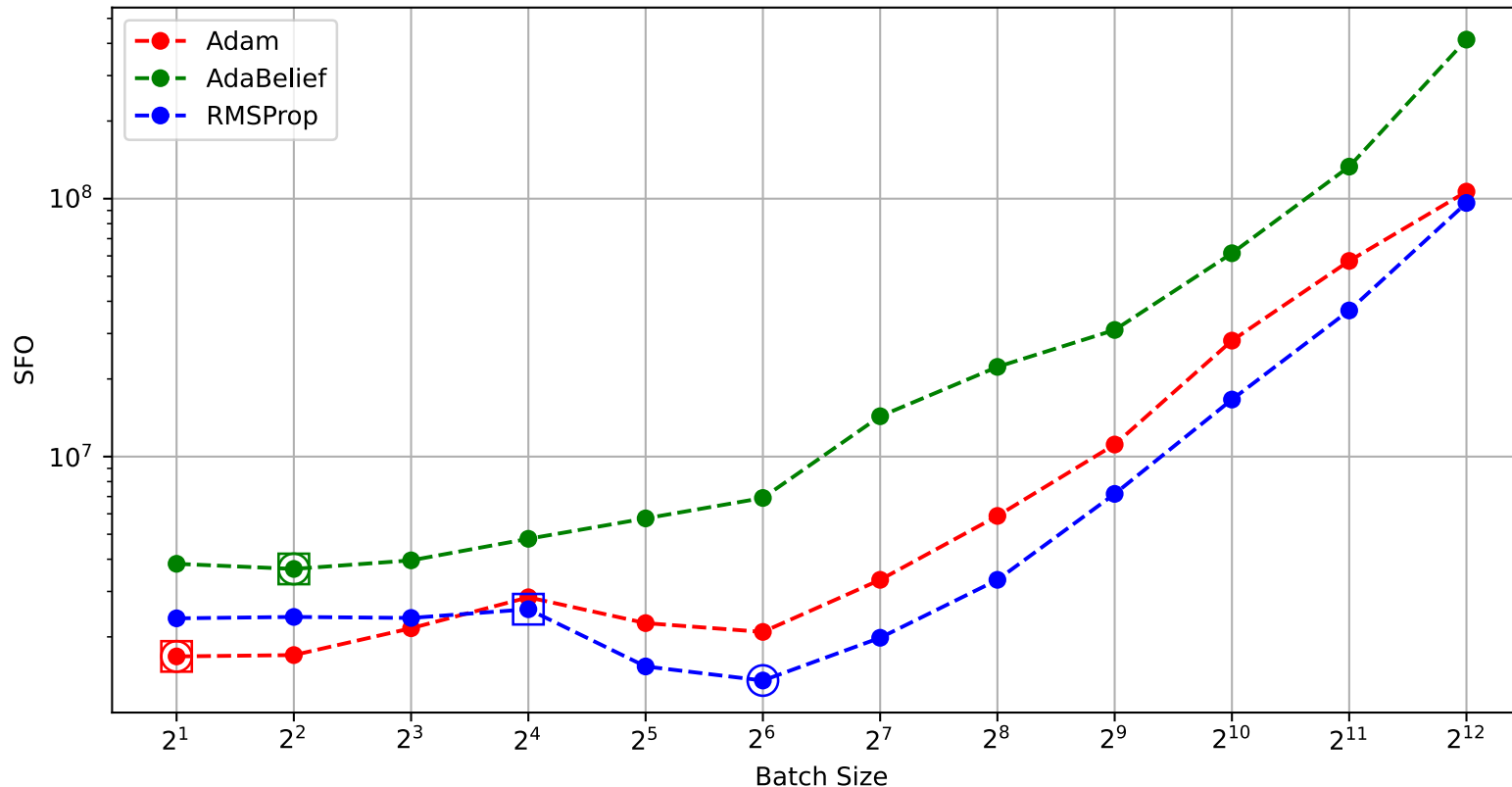
▷ RMSProp: 2^3

Nb vs b graph

Numerical Results

【 Existence of a Critical Batch Size 】

▷ Training WGAN-GP on the CelebA dataset



▷ Measured

▷ Adam: 2^1

▷ AdaBelief: 2^2

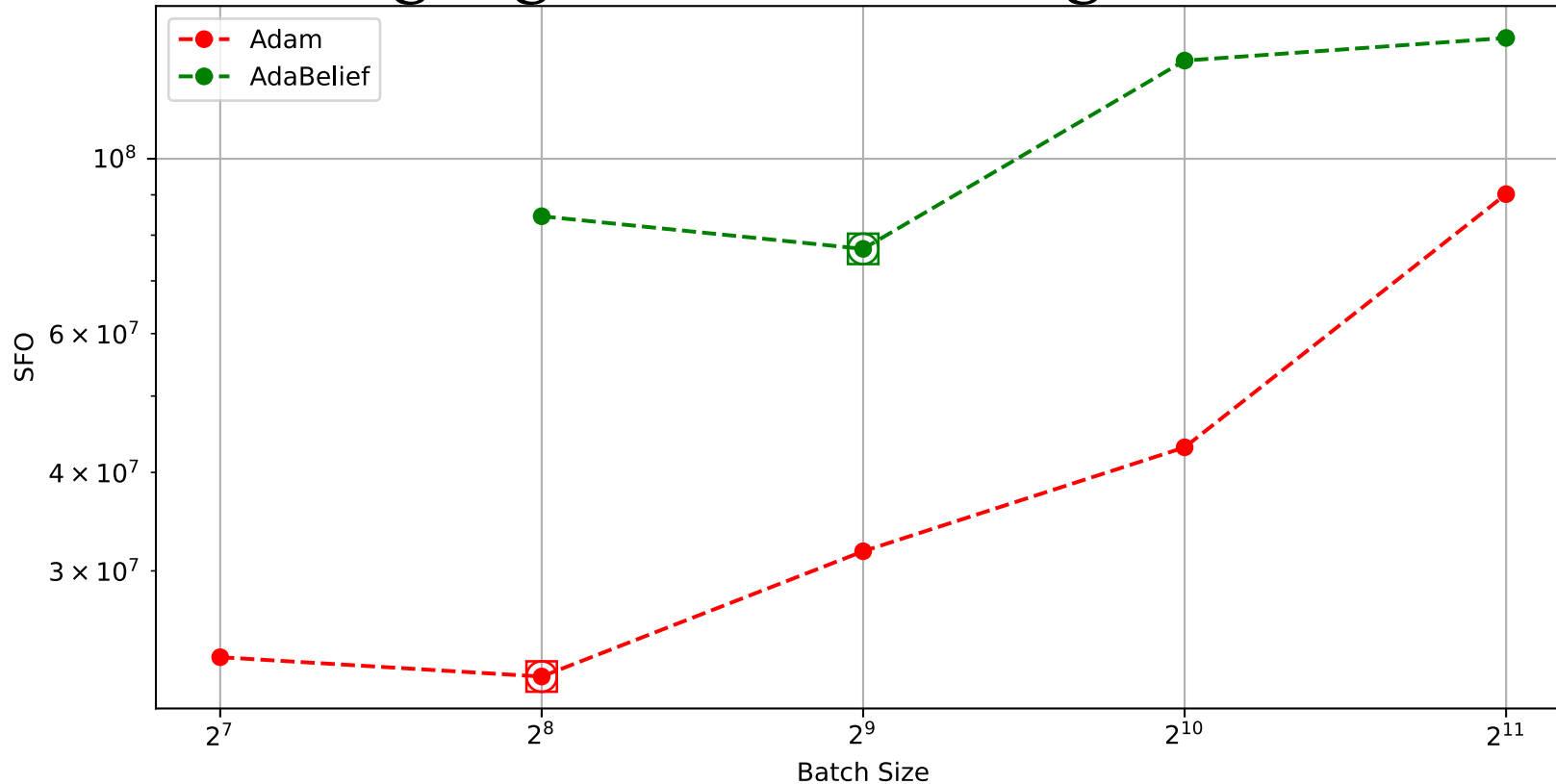
▷ RMSProp: 2^6

Nb vs b graph

Numerical Results

【 Existence of a Critical Batch Size 】

▷ Training BigGAN on the ImageNet dataset



▷ Measured

▷ Adam: 2^8

▷ AdaBelief: 2^9

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

$$\begin{aligned} b_G^* &\geq \frac{\sigma_G^2}{\epsilon_G^3} \frac{\alpha^G}{(1 - \beta_1^G)^3 \sqrt{\frac{\Theta}{1 - \beta_2^G} \frac{1}{|\mathcal{S}|^2}}} \\ &= \frac{\sigma_G^2}{\epsilon_G^3} \times \frac{0.0001}{(1 - 0.5)^3 \times \sqrt{\frac{3576704}{1 - 0.999} \times \frac{1}{3033042}}} \end{aligned}$$

- ▷ 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,

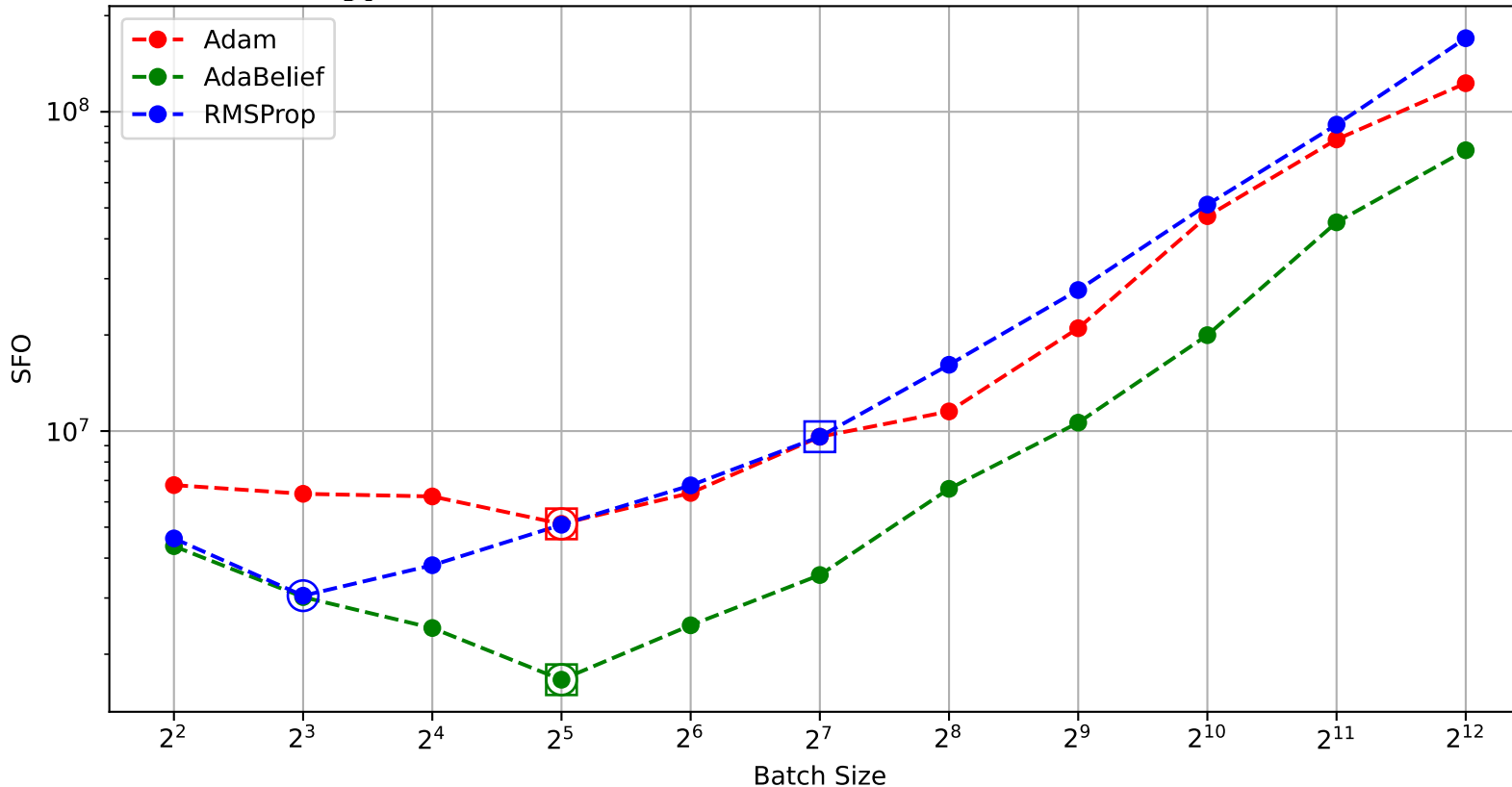
$$\begin{aligned} b_G^* &\geq 788.7 \times \frac{0.0003}{(1 - 0.5)^3 \times \sqrt{\frac{4 \times 3576704}{1 - 0.999}} \times \frac{1}{3033042}} \\ &\approx 47.9 \end{aligned}$$

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 】

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

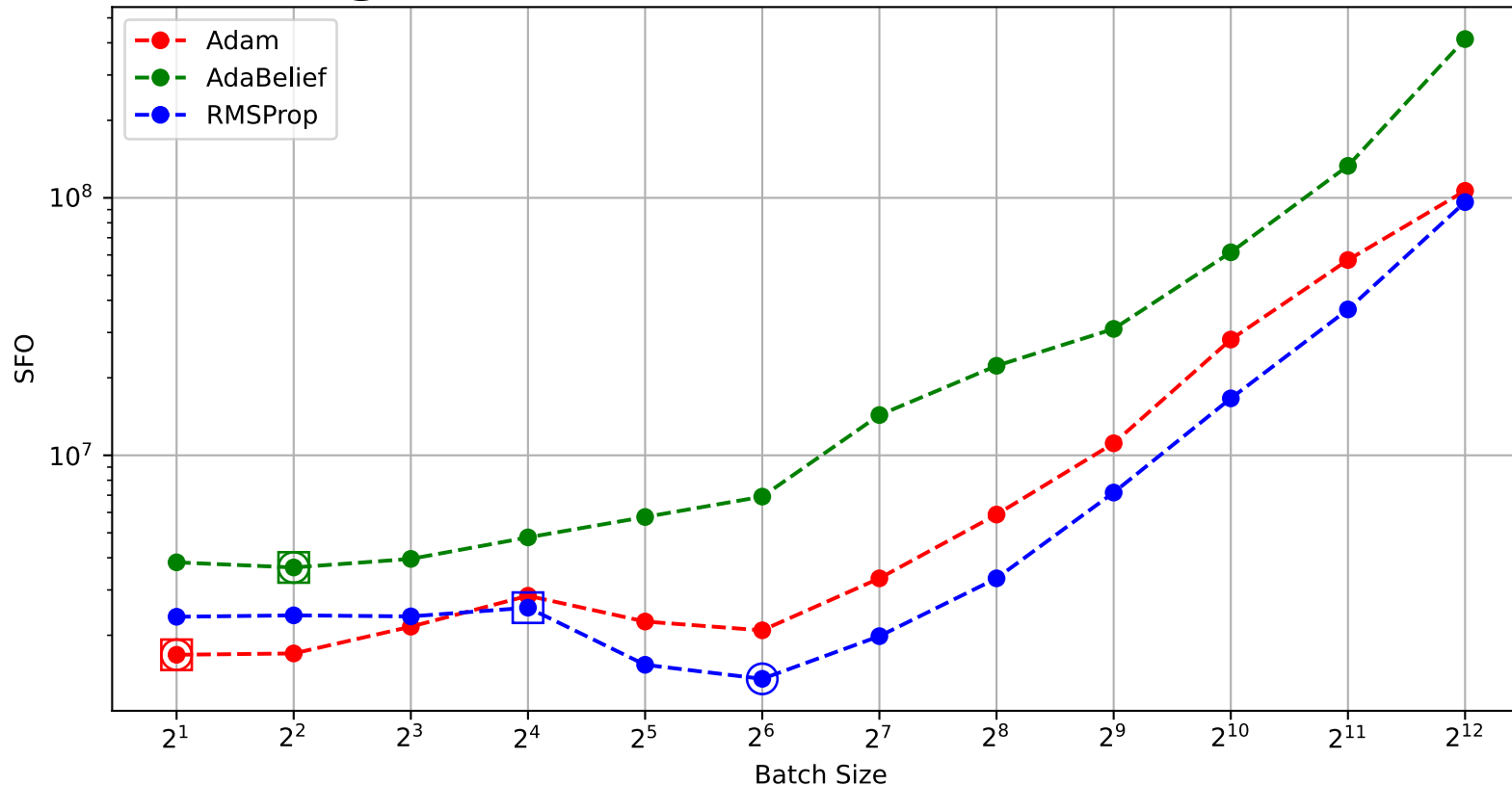
- ▷ Using this ratio, we can estimate other optimizer's critical batch size.
- ▷ Moreover, this ratio can be appropriated for **another GAN's training**, as long as the model used is the same.
- ▷ For example, for Adam in training WGAN-GP on the CelebA dataset,

$$b_G^* \geq 788.7 \times \frac{0.0001}{(1 - 0.5)^3 \times \sqrt{\frac{3576704}{1-0.999} \times \frac{1}{3033042}}} \\ \approx 1.7$$

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

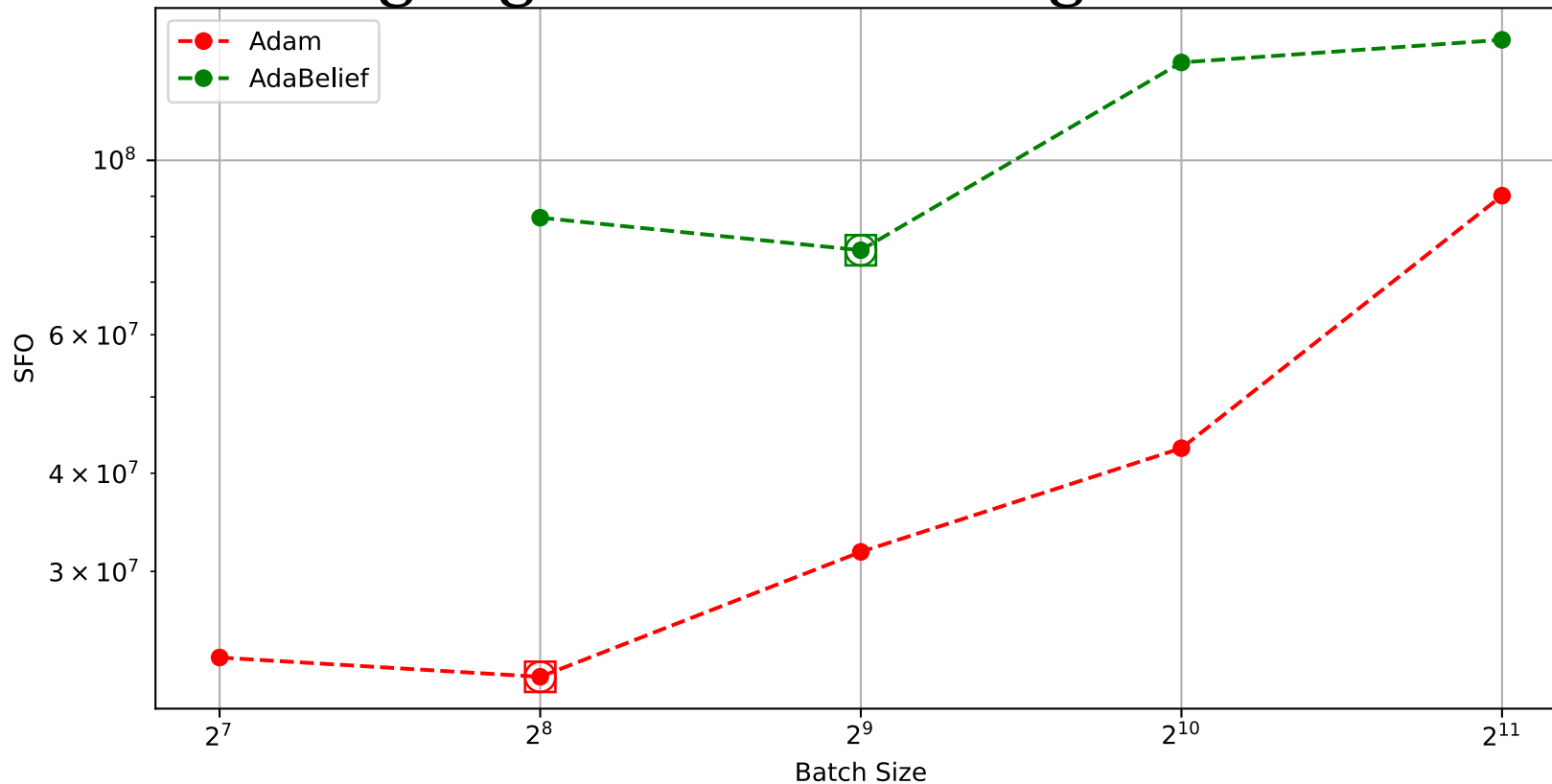
▷ RMSProp: 2^4

Numerical Results

【 Estimation of the Critical Batch Size 】

Nb vs b graph

▷ Training BigGAN on the ImageNet dataset



▷ 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**.