HITACHI
Inspire the Next

Influence Estimation for Generative Adversarial Networks

Naoyuki Terashita, Hiroki Ohashi, Yuichi Nonaka, and Takashi Kanemaru
Hitachi, Ltd.

Motivation and Contribution

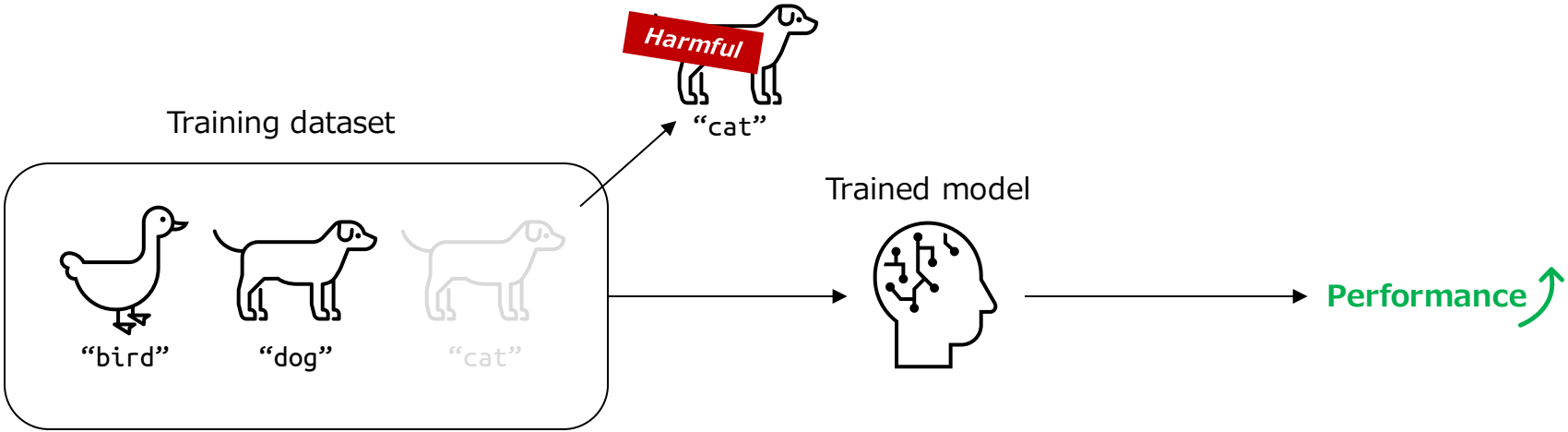
HITACHI
Inspire the Next

- Motivation
  - Identifying ***harmful training instances*** of GANs by ***influence estimation***
  - Solving 2 issues to bring approaches for supervised learning to influence estimation for GANs
    i. A training instance only takes an indirect role in the generator's training
    ii. Losses do not always represent the model performance

- Contribution
  - We proposed an estimator of influence on the GAN parameters using ***Jacobian of mini-batches*** to consider the indirect role of a training instance.
  - We proposed to evaluate the instance based on ***influence on GAN evaluation metric*** and proposed its estimator.
  - We evaluated estimated influence on GAN evaluation metric by two experiments (1. Estimation accuracy, 2. Data cleansing)

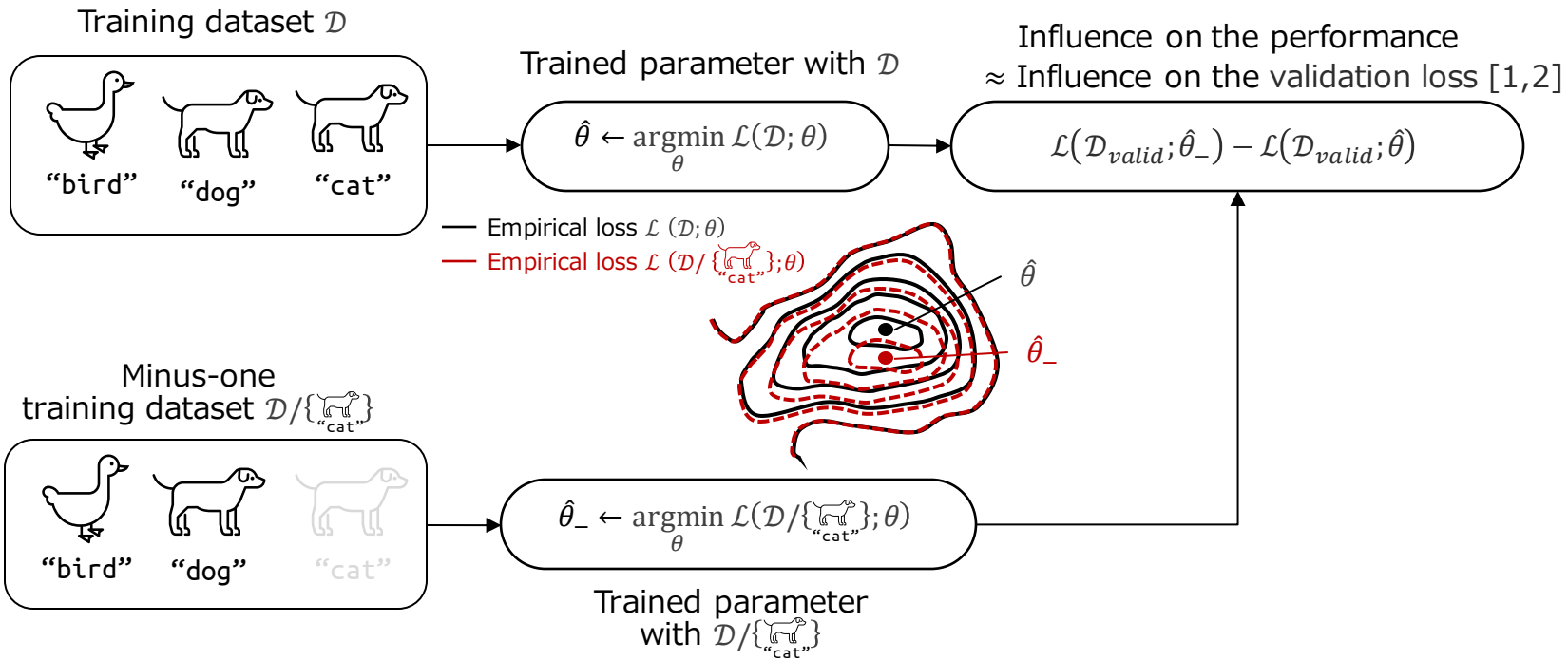
Influence estimation and harmful instance

- ***Influence*** of a training instance: scalar or vector of how absence of the instance changes the performance (parameters, or predictions) of the trained model.
- ***Influence Estimation***: Approximating influence without performing actual removal of instance and retraining
- ***Harmful instance***: A training instance whose absence has ***positive influence on the performance***



Influence on the performance in supervised learning

In supervised learning settings,



[1] Khanna, Rajiv, et al. "Interpreting black box predictions using fisher kernels." The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019.
[2] Hara, Satoshi, et al. "Data Cleansing for Models Trained with SGD.", Advances in Neural Information Processing Systems, 2019

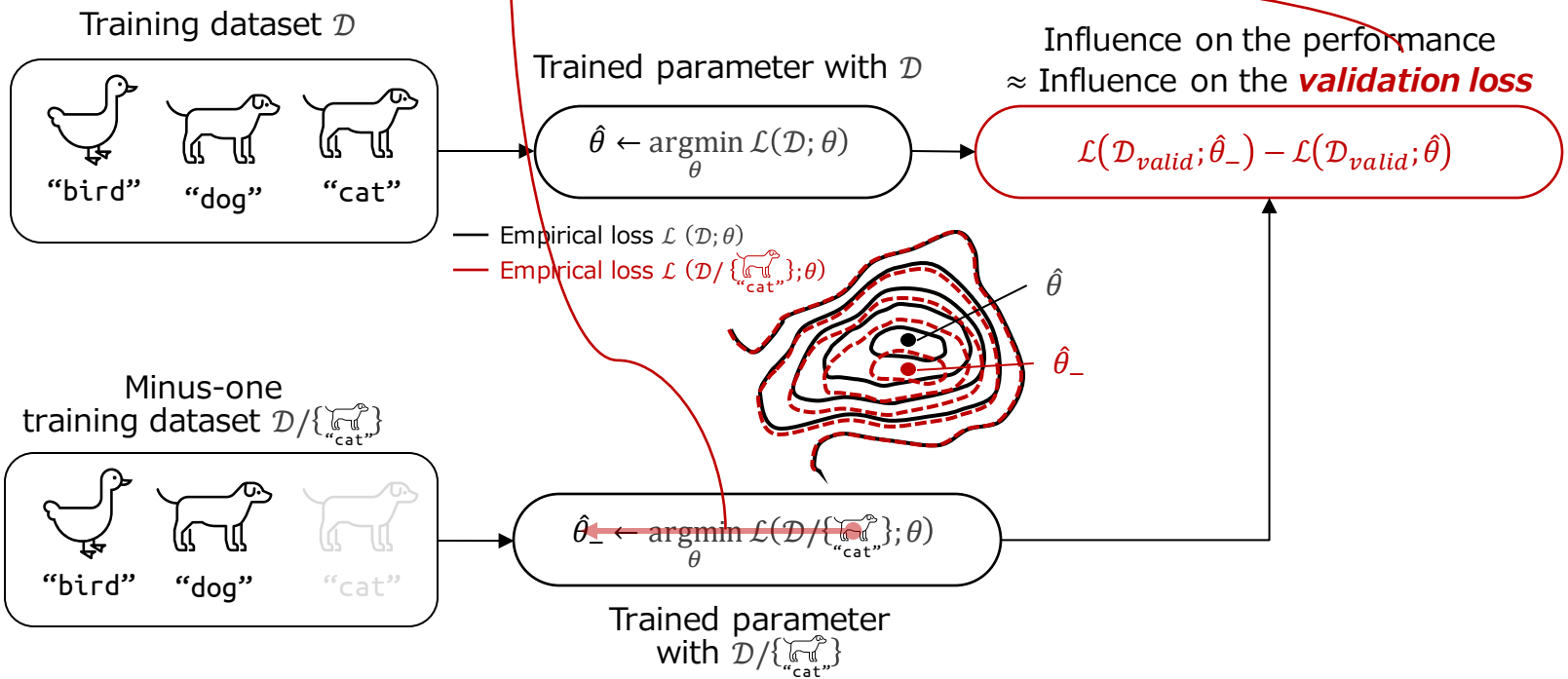
Assumptions in influence estimation for supervised learning

HITACHI
Inspire the Next

Existing approaches for supervised learning [1,2] put 2 assumptions:

i. Absence of an instance ***directly*** changes the training

ii. ***Validation loss represents the performance***



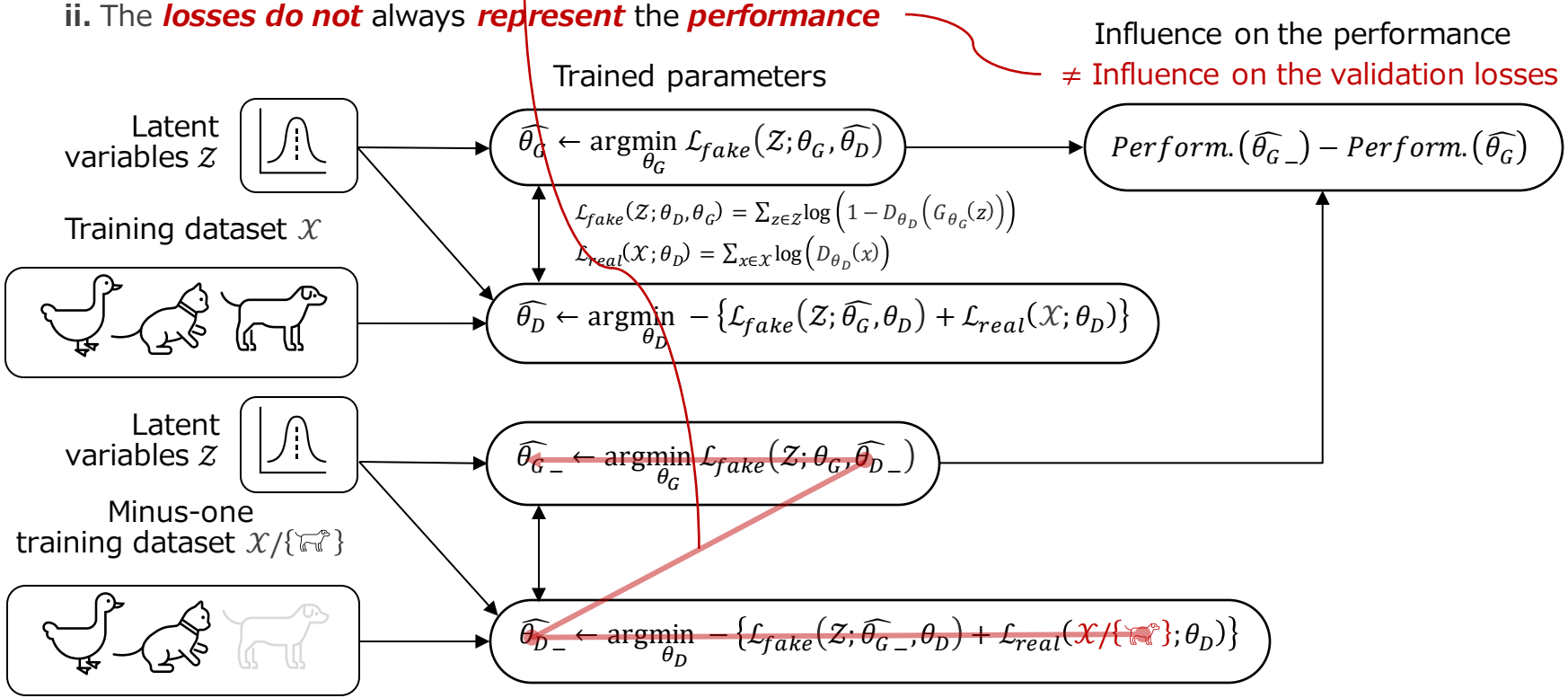
[1] Khanna, Rajiv, et al. "Interpreting black box predictions using fisher kernels." The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019.
[2] Hara, Satoshi, et al. "Data Cleansing for Models Trained with SGD.", Advances in Neural Information Processing Systems, 2019

Influence on the performance in GANs

In GAN's two models (Generator G_{θ_G}, Discriminator D_{θ_D}) setting,

i. Absence of an instance ***only indirectly*** changes the generator's training

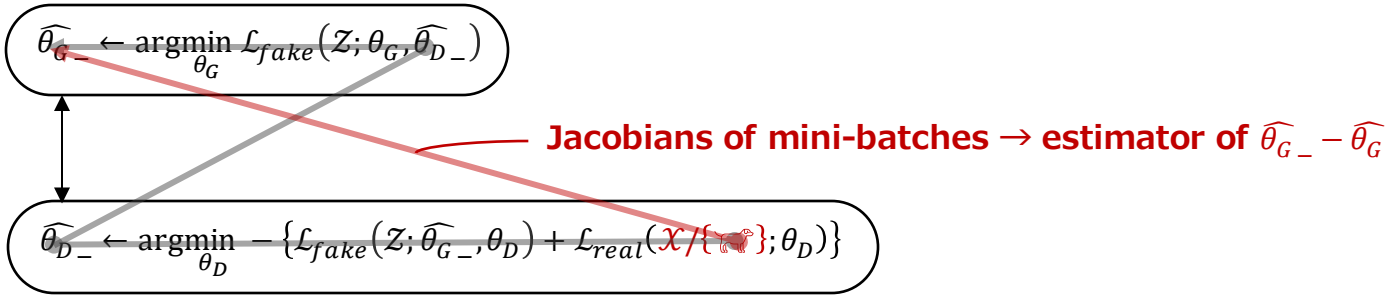
ii. The ***losses do not*** always ***represent*** the ***performance***



Our contribution

i. Absence of an instance ***only indirectly*** changes the generator's training

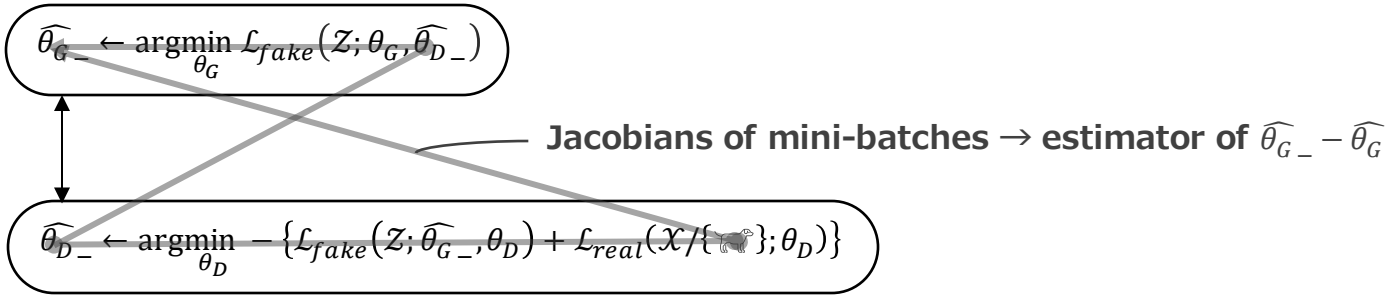
→ ***Jacobians of mini-batches*** can be used to obtain the **estimator of influence on the parameters**



Our contribution

i. Absence of an instance ***only indirectly*** changes the generator's training

→ ***Jacobians of mini-batches*** can be used to obtain the **estimator of influence on the parameters**



ii. The ***losses do not*** always ***represent*** the ***performance***

→ ***Influence on GAN evaluation metrics*** (e.g., Inception Score, FID) and its **estimator**.

Influence on GAN evaluation metrics

$$GAN\_Metric\left(G\left(\widehat{\theta_{G\_}}\right)\right) - GAN\_Metric\left(G\left(\widehat{\theta_G}\right)\right)$$

≈ Linear approximation

Estimator

$$\left\langle \nabla_{\theta_G} GAN\_Metric\left(G\left(\widehat{\theta_G}\right)\right), \widehat{\theta_{G\_}} - \widehat{\theta_G} \right\rangle$$

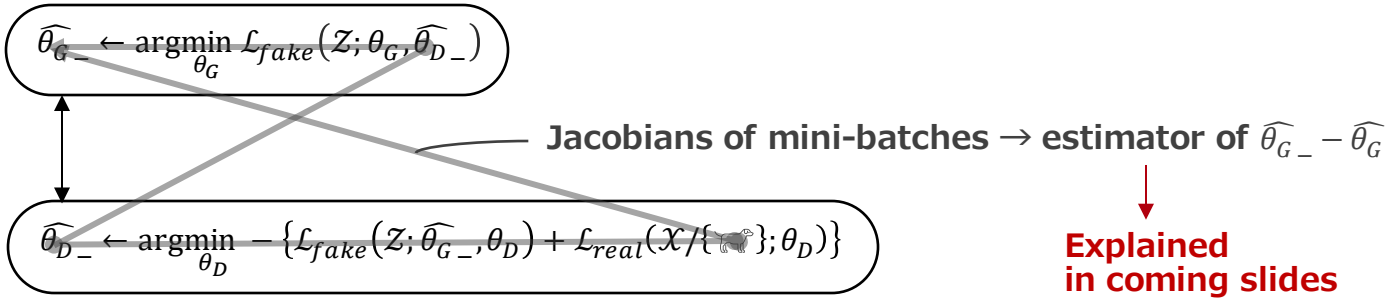
Computable when differentiable

Estimated by **i.**

Our contribution

i. Absence of an instance ***only indirectly*** changes the generator's training

→ ***Jacobians of mini-batches*** can be used to obtain the **estimator of influence on the parameters**



ii. The ***losses do not*** always ***represent*** the ***performance***

→ ***Influence on GAN evaluation metrics*** (e.g., Inception Score, FID) and its **estimator**.

Influence on GAN evaluation metrics

$$GAN\_Metric\left(G\left(\widehat{\theta_{G\_}}\right)\right) - GAN\_Metric\left(G\left(\widehat{\theta_G}\right)\right)$$

≈ Linear approximation

Estimator

$$\left\langle \nabla_{\theta_G} GAN\_Metric\left(G\left(\widehat{\theta_G}\right)\right), \widehat{\theta_{G\_}} - \widehat{\theta_G} \right\rangle$$

Computable when differentiable

Estimated by **i.**

Proposed estimator of influence on $[\theta_G, \theta_D]^\top$ with Jacobian

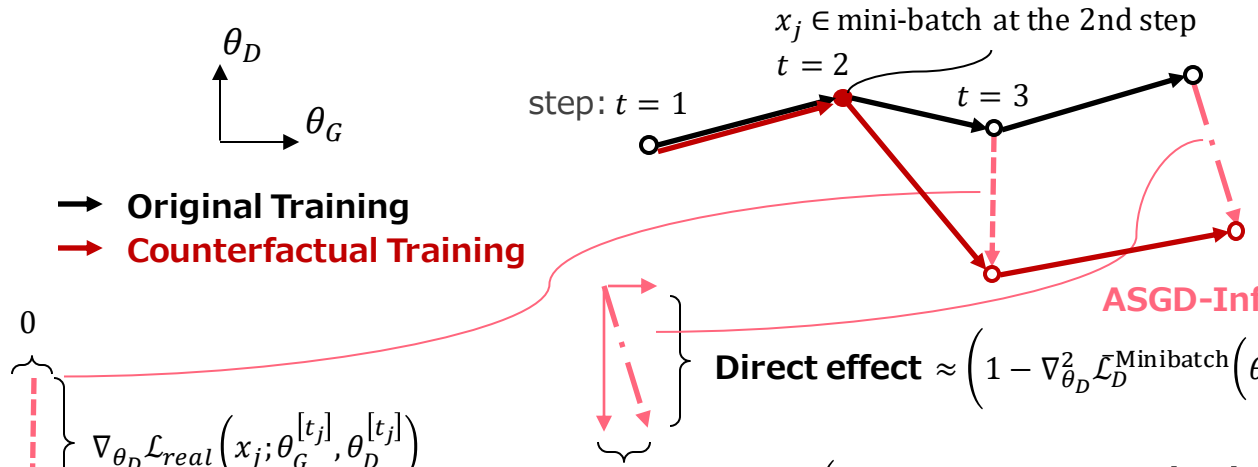
We suppose SGD training, where …

- Parameter space $[\theta_G, \theta_D]^\top$ is only 2-dimensional
- θ_G and θ_D are updated simultaneously
- Both learning rates are 1
- 3 update steps
- We want to know influence of j-th instance $x_j \in \mathcal{X}$

} Only for simplicity

$$\mathcal{L}_G^{\text{Minibatch}} := \sum_{z \in Z_{minibatch}} \mathcal{L}_{fake}(z; \theta_G, \theta_D)$$

$$\mathcal{L}_D^{\text{Minibatch}} := -\sum_{z \in Z_{minibatch}} \mathcal{L}_{fake}(z; \theta_G, \theta_D) - \sum_{x \in X_{minibatch}} \mathcal{L}_{real}(x; \theta_D)$$

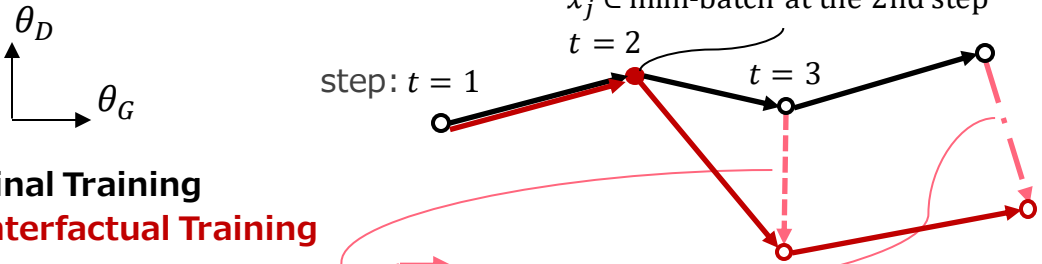


Proposed estimator of influence on $[\theta_G, \theta_D]^\top$ with Jacobian

We suppose SGD training, where …

- Parameter space $[\theta_G, \theta_D]^\top$ is only 2-dimensional
- θ_G and θ_D are updated simultaneously
- Both learning rates are 1
- 3 update steps
- We want to know influence of j-th instance $x_j \in \mathcal{X}$

} Only for simplicity



See our paper for more general estimator

Original Training

Counterfactual Training

ASGD-Influence

0

$\nabla_{\theta_D} \mathcal{L}_{real}\left(x_j; \theta_G^{[t_j]}, \theta_D^{[t_j]}\right)$

Direct effect

Direct effect $\approx \left(1 - \nabla_{\theta_D}^2 \mathcal{L}_D^{\text{Minibatch}}\left(\theta_G^{[t_j+1]}, \theta_D^{[t_j+1]}\right)\right) \cdot \nabla_{\theta_D} \mathcal{L}_{real}\left(x_j; \theta_G^{[t_j]}, \theta_D^{[t_j]}\right)$

Indirect effect *from* θ_D to θ_G

$\approx \left(1 - \nabla_{\theta_D} \nabla_{\theta_G} \mathcal{L}_G^{\text{Minibatch}}\left(\theta_G^{[t_j+1]}, \theta_D^{[t_j+1]}\right)\right) \cdot \nabla_{\theta_D} \mathcal{L}_{real}\left(x_j; \theta_G^{[t_j]}, \theta_D^{[t_j]}\right)$

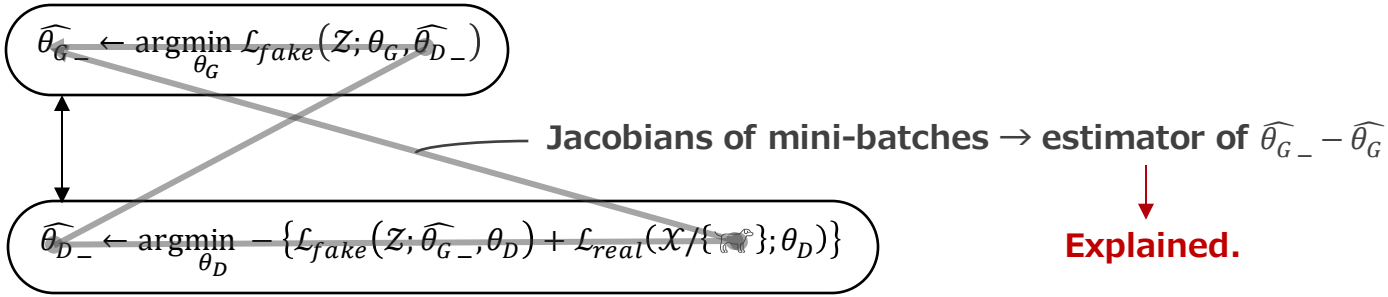
↑

Jacobian block of a mini-batch

Our contribution

i. Absence of an instance ***only indirectly*** changes the generator's training

→ ***Jacobians of mini-batches*** can be used to obtain the **estimator of influence on the parameters**



ii. The ***losses do not*** always ***represent*** the ***performance***

→ ***Influence on GAN evaluation metrics*** (e.g., Inception Score, FID) and its **estimator**.

Influence on GAN evaluation metrics

$$GAN\_Metric\left(G(\widehat{\theta_{G\_}})\right) - GAN\_Metric\left(G(\widehat{\theta_G})\right)$$

≈ Linear approximation

Estimator

$$\left\langle \nabla_{\theta_G} GAN\_Metric\left(G(\widehat{\theta_G})\right), \widehat{\theta_{G\_}} - \widehat{\theta_G} \right\rangle$$

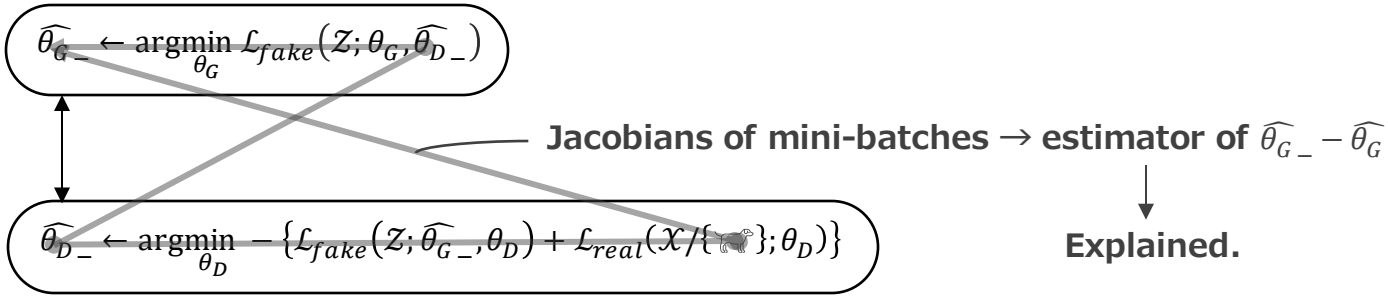
Computable when differentiable

Estimated by **i.**

Our contribution

i. Absence of an instance ***only indirectly*** changes the generator's training

→ ***Jacobians of mini-batches*** can be used to obtain the **estimator of influence on the parameters**



ii. The ***losses do not*** always ***represent*** the ***performance***

→ ***Influence on GAN evaluation metrics*** (e.g., Inception Score, FID) and its **estimator**.

Evaluated in the experiments

Influence on GAN evaluation metrics

$$GAN\_Metric\left(G(\widehat{\theta_{G\_}})\right) - GAN\_Metric\left(G(\widehat{\theta_G})\right)$$

≈ Linear approximation

Estimator

$$\left\langle \nabla_{\theta_G} GAN\_Metric\left(G(\widehat{\theta_G})\right), \widehat{\theta_{G\_}} - \widehat{\theta_G} \right\rangle$$

Computable when differentiable

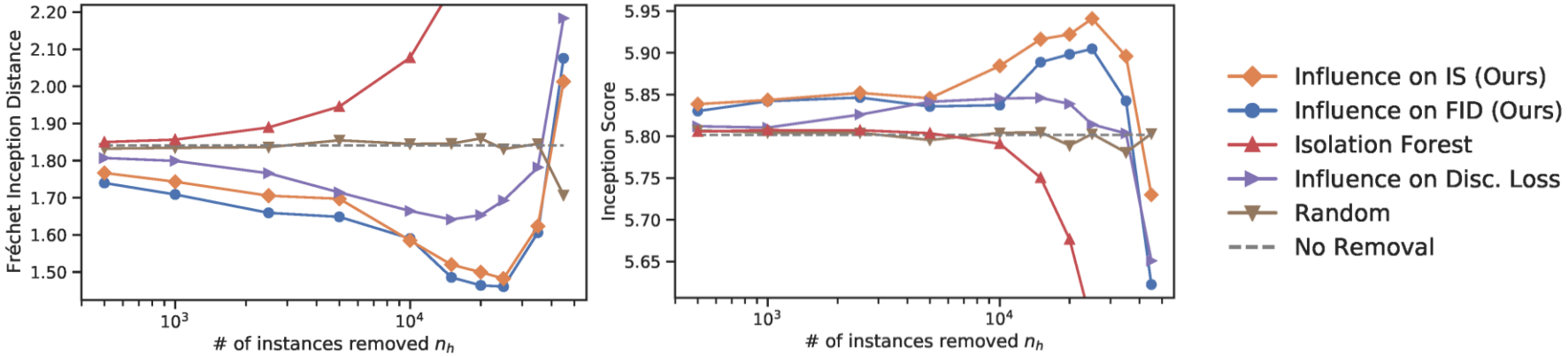
Estimated by **i.**

Experiments

- **Exp. 1: Estimation Accuracy**
- **Exp. 2: Data Cleansing**
  - How does the generative performance improve when we remove suggested harmful instances?

Experiments (Data Cleansing)

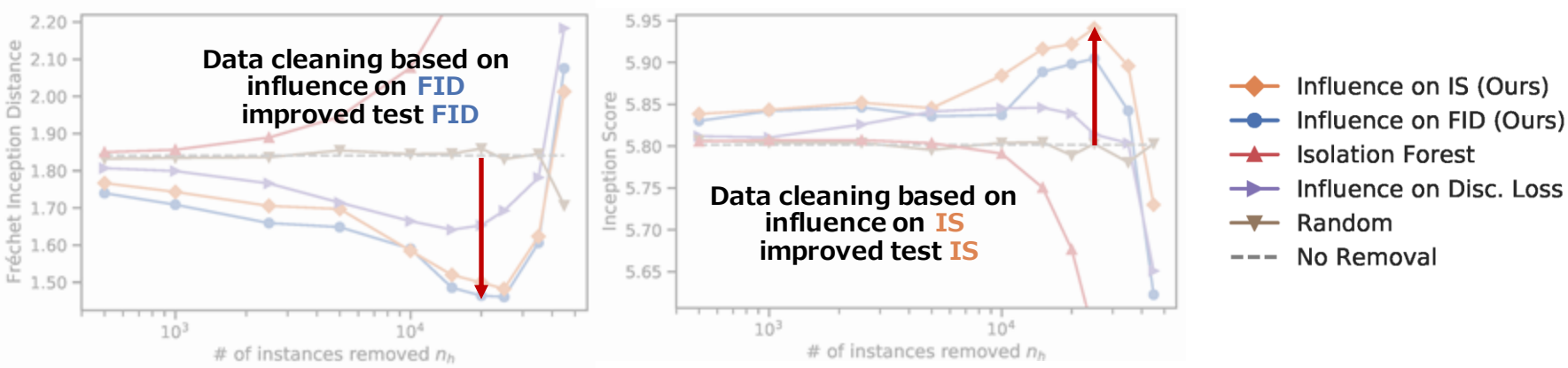
- Setup
  - Dataset: MNIST, Architecture: Deep Convolutional GAN
  - Select top n_h **highly-harmful instances** based on,
    - **influence on GAN evaluation metrics** (Inception Score (IS) and Fréchet Inception Score (FID))
    - baselines (Isolation Forest, Random, Discriminator Loss)
  - Retrain the model from the last epoch without the highly-harmful instances (**Data cleansing**)
  - Evaluate improvements in test IS/FID on retrained models from the original model (No Removal).



Experiments (Data Cleansing)

- Setup
  - Dataset: MNIST, Architecture: Deep Convolutional GAN
  - Select top n_h **highly-harmful instances** based on,
    - **influence on GAN evaluation metrics** (Inception Score (IS) and Fréchet Inception Score (FID))
    - baselines (Isolation Forest, Random, Discriminator Loss)
  - Retrain the model from the last epoch without the highly-harmful instances (**Data cleansing**)
  - Evaluate improvements in test IS/FID on retrained models from the original model (No Removal).

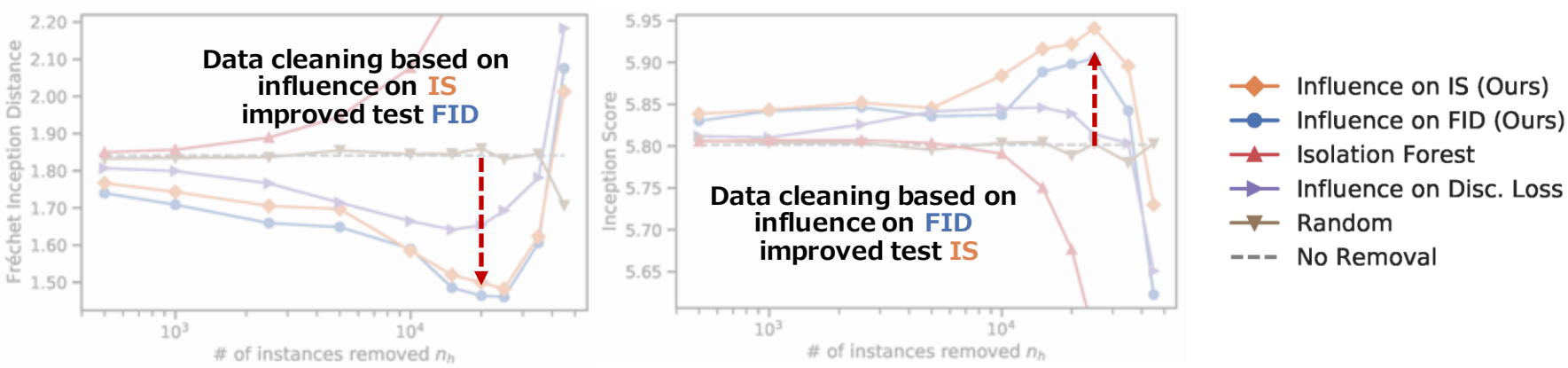
- Quantitative Results
  - GAN evaluation metrics were **statistically significantly improved**.



Experiments (Data Cleansing)

- Setup
  - Dataset: MNIST, Architecture: Deep Convolutional GAN
  - Select top n_h **highly-harmful instances** based on,
    - **influence on GAN evaluation metrics** (Inception Score (IS) and Fréchet Inception Score (FID))
    - baselines (Isolation Forest, Random, Discriminator Loss)
  - Retrain the model from the last epoch without the highly-harmful instances (**Data cleansing**)
  - Evaluate improvements in test IS/FID on retrained models from the original model (No Removal).

- Quantitative Results
  - GAN evaluation metrics were **statistically significantly improved**.
  - **Data cleansing** based on the **influence on FID** also **improved IS**, and vice versa.



Conclusion

HITACHI
Inspire the Next

- Existing influence estimation for supervised learning implicitly requires …
  i. Absence of a training instance to directly change the training
  ii. Validation loss to represent the performance

- However, in GAN's training,
  i. Absence of a training instance only indirectly changes the generator's training
  ii. The losses do not always represent the performance

- Our contribution:
  i. We proposed estimator of influence on the parameters using Jacobians of mini-batches.
  ii. We proposed to evaluate by influence on GAN evaluation metric and proposed its estimator.
  iii. We conducted experiments (1. Estimation accuracy, 2. Data cleansing) to evaluate our estimator.

- Experiment (Data Cleansing)
  - Removing a set of the highly-harmful instances improved GAN evaluation metrics,
    - including a GAN evaluation metric not used for the influence estimation
  - Data cleansing improved the visual diversity in the generated samples.