GS-WGAN:
A Gradient-Sanitized Approach for Learning Differentially Private Generators



Dingfan Chen[1] Tribhuvanesh Orekondy[2] Mario Fritz[1]

[1] CISPA Helmholtz Center for Information Security
[2] Max Planck Institute for Informatics

In a Nutshell

In a Nutshell

- **Problem**
    - High-dimensional data generation with differential privacy guarantees

In a Nutshell

- **Problem**
  - High-dimensional data generation with differential privacy guarantees

- **Method:** Gradient Sanitization Approach for GANs
  - Key:
    - Sanitize gradients w.r.t. the generated samples
    - Exploit the Lipschitz property of Wasserstein GANs
  - Many benefits:
    - Avoids intensive hyper-parameters search
    - Allows stable training with complex model architectures
    - Applies seamlessly to centralized/ decentralized(federated) setting

In a Nutshell

- **Problem**
    - High-dimensional data generation with differential privacy guarantees
- **Method:** Gradient Sanitization Approach for GANs
    - Key:
        - Sanitize gradients w.r.t. the generated samples
        - Exploit the Lipschitz property of Wasserstein GANs
    - Many benefits:
        - Avoids intensive hyper-parameters search
        - Allows stable training with complex model architectures
        - Applies seamlessly to centralized/ decentralized(federated) setting
- **Results**
    - Extensive evaluation: 2 settings, 3 datasets, 5 baselines …
    - Promising results: Consistent improvement over baselines across different datasets, settings and metrics

Problem

[1] Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014
[2] Dwork et al., “The Algorithmic Foundations of Differential Privacy”, Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., “Deep Learning with Differential Privacy”, CCS 2016

Problem

- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task
  - Rigorous privacy guarantee

$\}\longrightarrow$ Generative Adversarial Networks (GANs)[1]

$\longrightarrow$ Differential Privacy (DP)[2]

[1] Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014
[2] Dwork et al., “The Algorithmic Foundations of Differential Privacy”, Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., “Deep Learning with Differential Privacy”, CCS 2016

Problem

- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task
  - Rigorous privacy guarantee

Generative Adversarial Networks (GANs)[1]

Differential Privacy (DP)[2]

- Existing Approach
  - Differentially private stochastic gradient descent (DP-SGD)[3]

[1] Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014
[2] Dwork et al., “The Algorithmic Foundations of Differential Privacy”, Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., “Deep Learning with Differential Privacy”, CCS 2016

Problem

- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task
  - Rigorous privacy guarantee

Generative Adversarial Networks (GANs)[1]

Differential Privacy (DP)[2]

- Existing Approach
  - Differentially private stochastic gradient descent (DP-SGD)[3]
    - Gradient
      $\boldsymbol{g}^{(t)} := \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G)$
    - Gradient descent step
      $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)} - \eta \cdot \boldsymbol{g}^{(t)}$

non-private sensitive (ε,δ)-private

[1] Goodfellow et al., "Generative Adversarial Nets", NIPS 2014
[2] Dwork et al., "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., "Deep Learning with Differential Privacy", CCS 2016

Problem

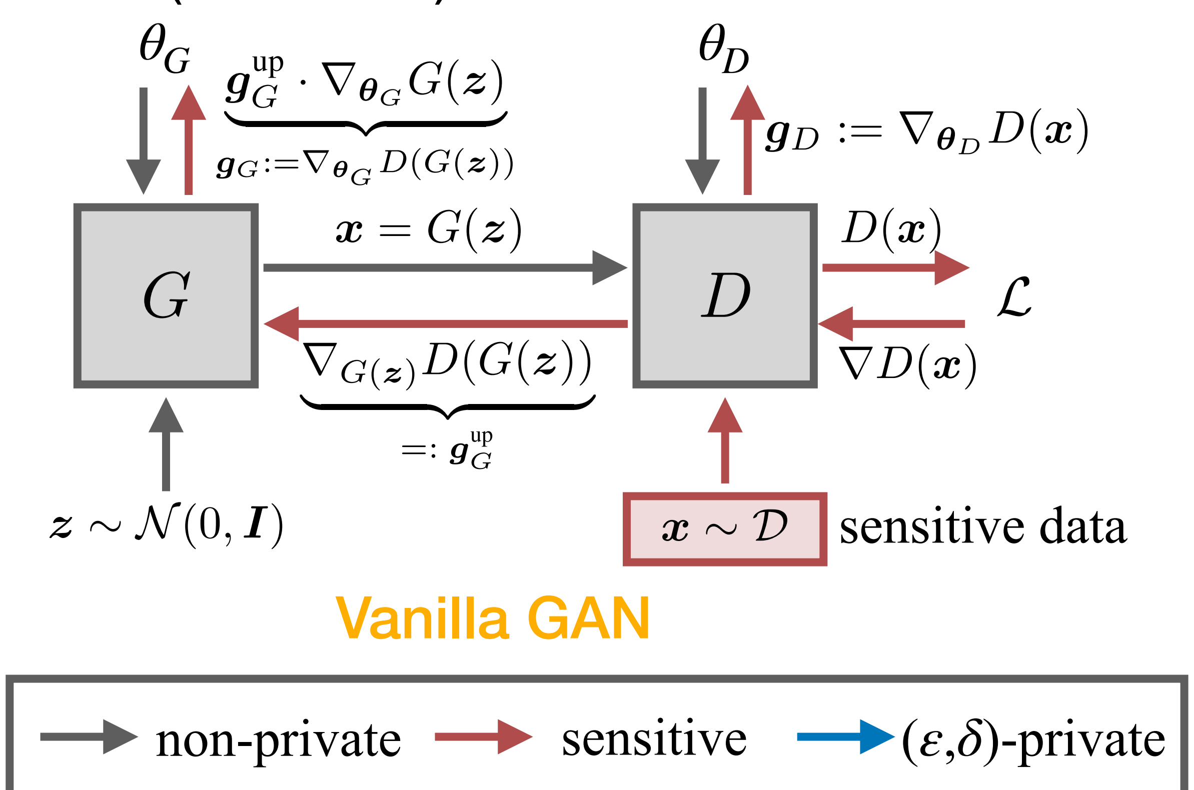
- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task

  → Generative Adversarial Networks (GANs)[1]

  - Rigorous privacy guarantee

  → Differential Privacy (DP)[2]

- Existing Approach
  - Differentially private stochastic gradient descent (DP-SGD)[3]
    - Gradient
      $\boldsymbol{g}^{(t)} := \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G)$
    - Gradient descent step
      $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)} - \eta \cdot \boldsymbol{g}^{(t)}$



Vanilla GAN

[1] Goodfellow et al., "Generative Adversarial Nets", NIPS 2014
[2] Dwork et al., "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., "Deep Learning with Differential Privacy", CCS 2016

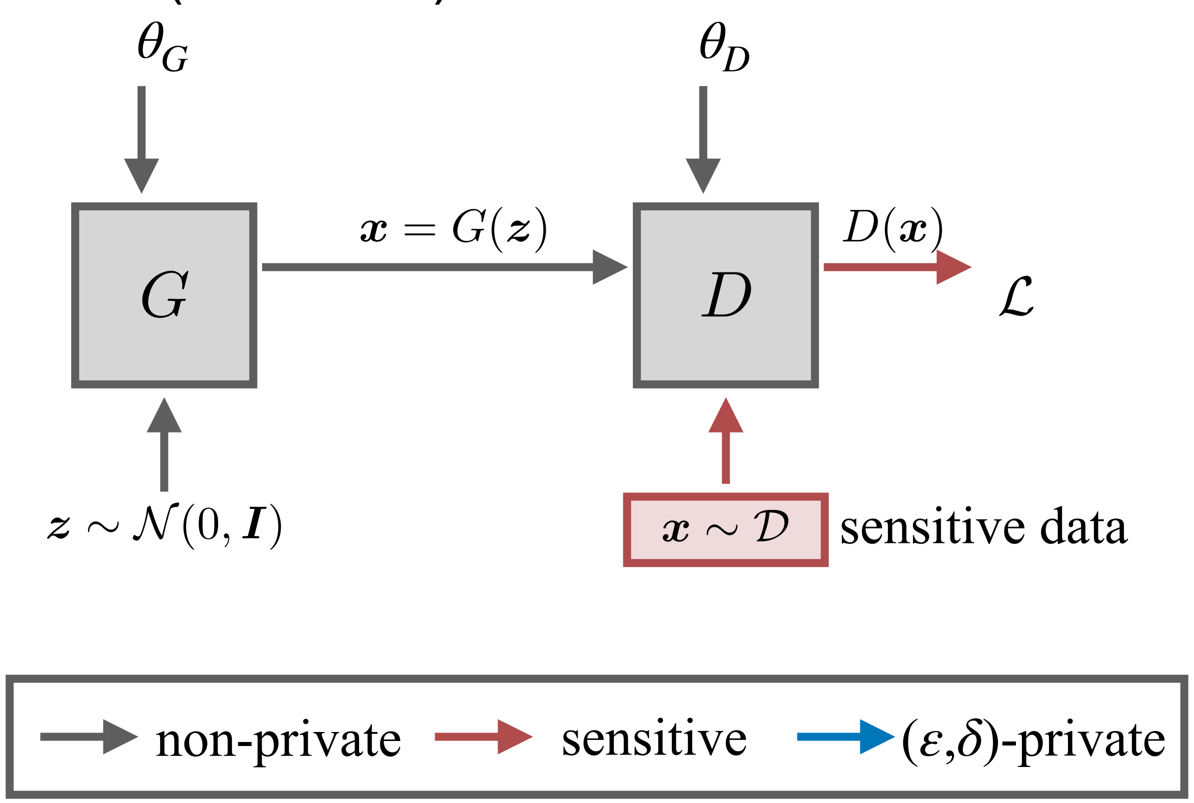
Problem

- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task
  - Rigorous privacy guarantee

Generative Adversarial Networks (GANs)[1]

Differential Privacy (DP)[2]

- Existing Approach
  - Differentially private stochastic gradient descent (DP-SGD)[3]
    - Gradient
      $\boldsymbol{g}^{(t)} := \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G)$
    - Sanitization mechanism
      $\hat{\boldsymbol{g}}^{(t)} := \mathcal{M}_{\sigma,C}(\boldsymbol{g}^{(t)}) = \mathrm{clip}(\boldsymbol{g}^{(t)}, C) + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})$
    - Gradient descent step
      $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)} - \eta \cdot \hat{\boldsymbol{g}}^{(t)}$



[1] Goodfellow et al., "Generative Adversarial Nets", NIPS 2014
[2] Dwork et al., "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., "Deep Learning with Differential Privacy", CCS 2016

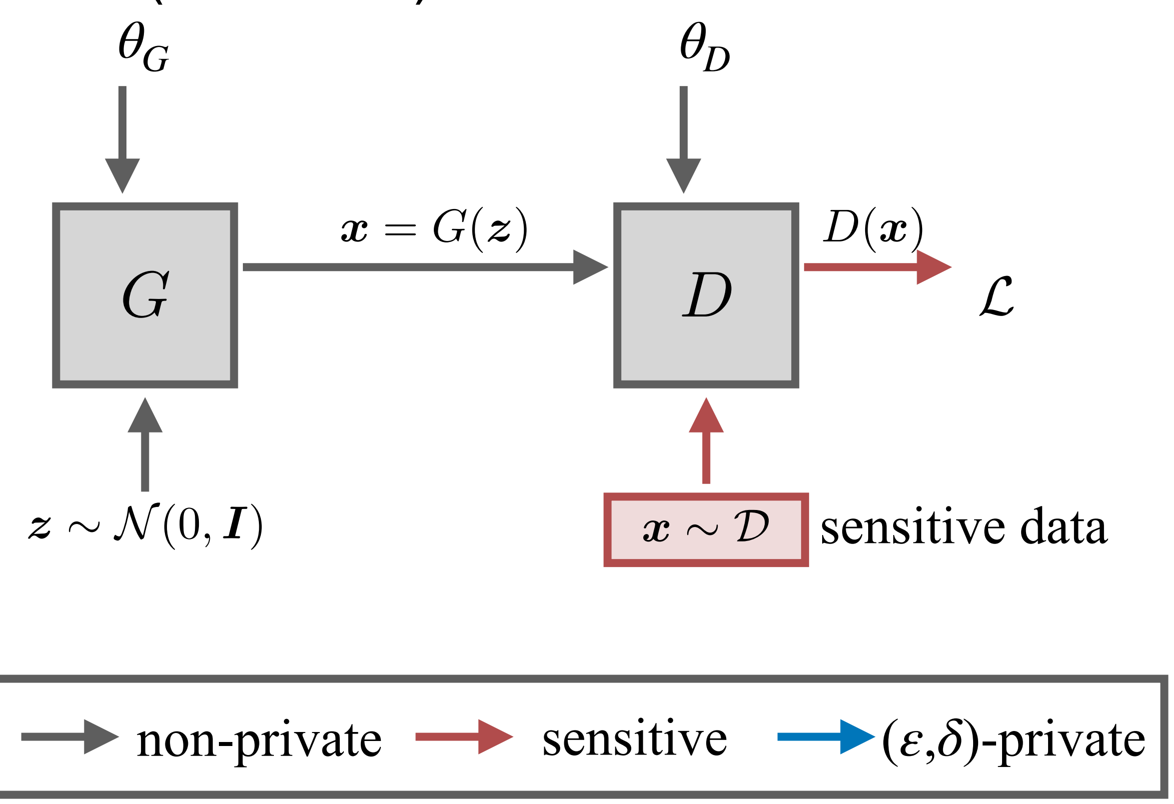
Problem

- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task
  - Rigorous privacy guarantee

High-dimensional data, Arbitrary downstream task → Generative Adversarial Networks (GANs)[1]

Rigorous privacy guarantee → Differential Privacy (DP)[2]

- Existing Approach
  - Differentially private stochastic gradient descent (DP-SGD)[3]
    - Gradient
      $\boldsymbol{g}^{(t)} := \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G)$
    - Sanitization mechanism
      $\hat{\boldsymbol{g}}^{(t)} := \mathcal{M}_{\sigma,C}(\boldsymbol{g}^{(t)}) = \mathrm{clip}(\boldsymbol{g}^{(t)}, C) + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})$
    - Gradient descent step
      $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)} - \eta \cdot \hat{\boldsymbol{g}}^{(t)}$



[1] Goodfellow et al., "Generative Adversarial Nets", NIPS 2014
[2] Dwork et al., "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., "Deep Learning with Differential Privacy", CCS 2016

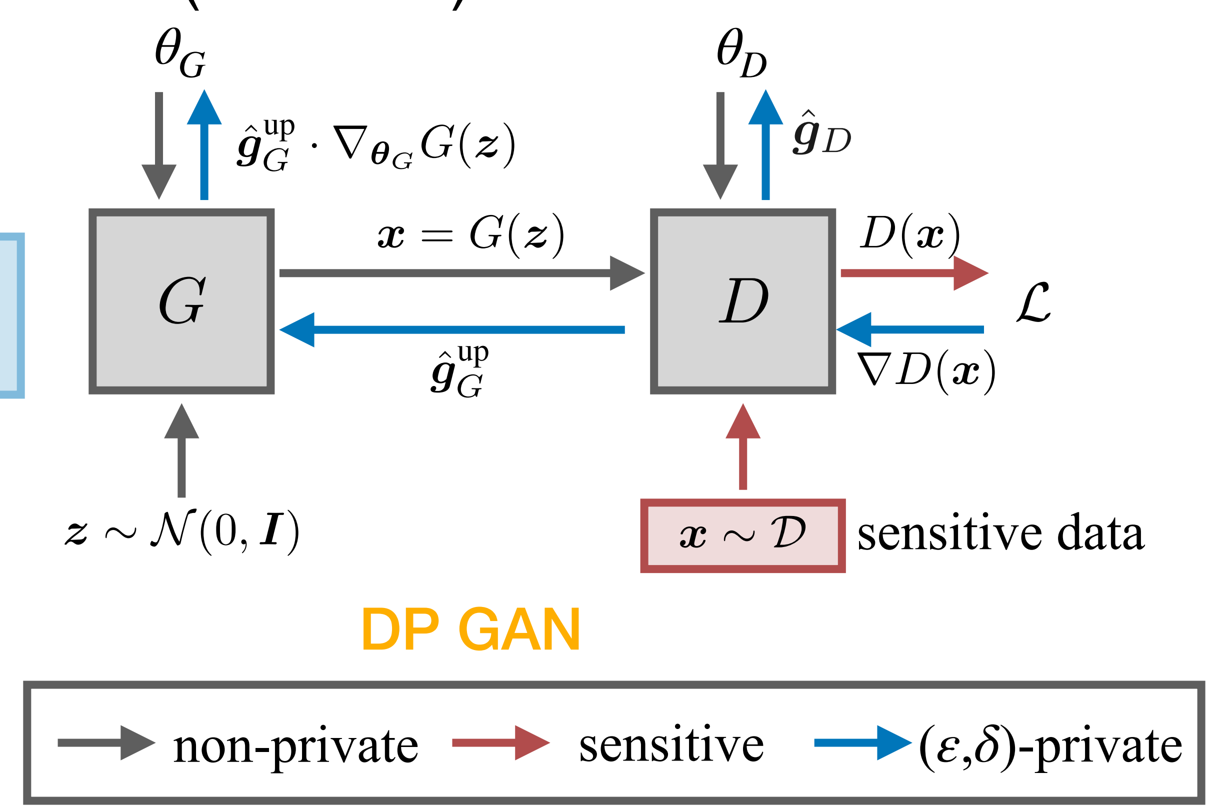
Problem

- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task

  } → Generative Adversarial Networks (GANs)[1]

  - Rigorous privacy guarantee → Differential Privacy (DP)[2]

- Existing Approach
  - Differentially private stochastic gradient descent (DP-SGD)[3]
    - Gradient
      $\boldsymbol{g}^{(t)} := \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G)$
    - Sanitization mechanism
      $\hat{\boldsymbol{g}}^{(t)} := \mathcal{M}_{\sigma,C}(\boldsymbol{g}^{(t)}) = \text{clip}(\boldsymbol{g}^{(t)}, C) + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})$
    - Gradient descent step
      $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)} - \eta \cdot \hat{\boldsymbol{g}}^{(t)}$



DP GAN

[1] Goodfellow et al., "Generative Adversarial Nets", NIPS 2014
[2] Dwork et al., "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., "Deep Learning with Differential Privacy", CCS 2016

Problem

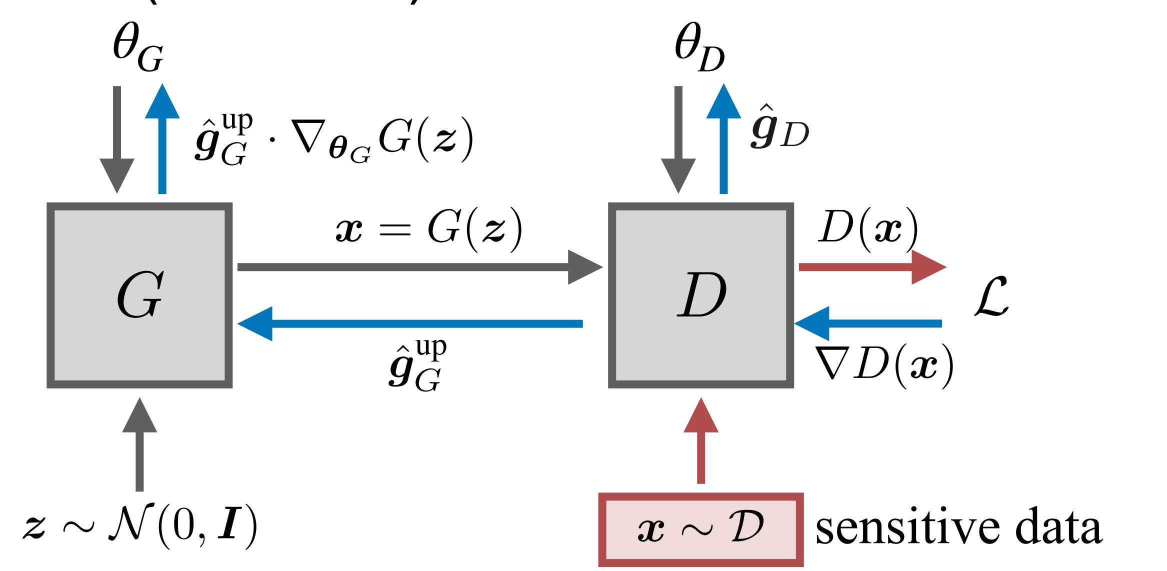
- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task
  - Rigorous privacy guarantee

High-dimensional data, Arbitrary downstream task → Generative Adversarial Networks (GANs)[1]

Rigorous privacy guarantee → Differential Privacy (DP)[2]

- Existing Approach
  - Differentially private stochastic gradient descent (DP-SGD)[3]
    - Gradient
      $\boldsymbol{g}^{(t)} := \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_D, \boldsymbol{\theta}_G)$
    - Sanitization mechanism
      $\hat{\boldsymbol{g}}^{(t)} := \mathcal{M}_{\sigma,C}(\boldsymbol{g}^{(t)}) = \mathrm{clip}(\boldsymbol{g}^{(t)}, \boxed{C}) + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})$
    - Gradient descent step
      $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)} - \eta \cdot \hat{\boldsymbol{g}}^{(t)}$

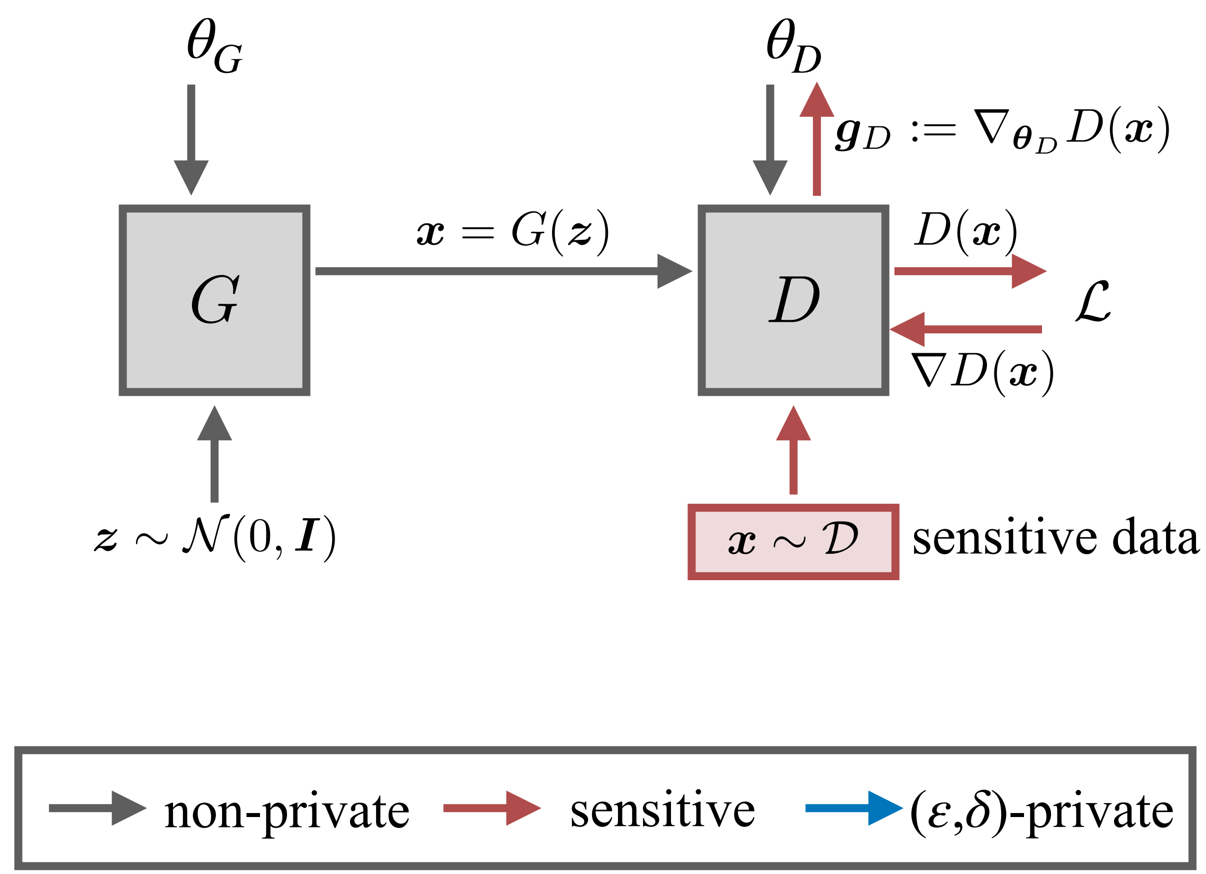
clipping bound



DP GAN

[1] Goodfellow et al., "Generative Adversarial Nets", NIPS 2014
[2] Dwork et al., "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science
[3] Abadi et al., "Deep Learning with Differential Privacy", CCS 2016

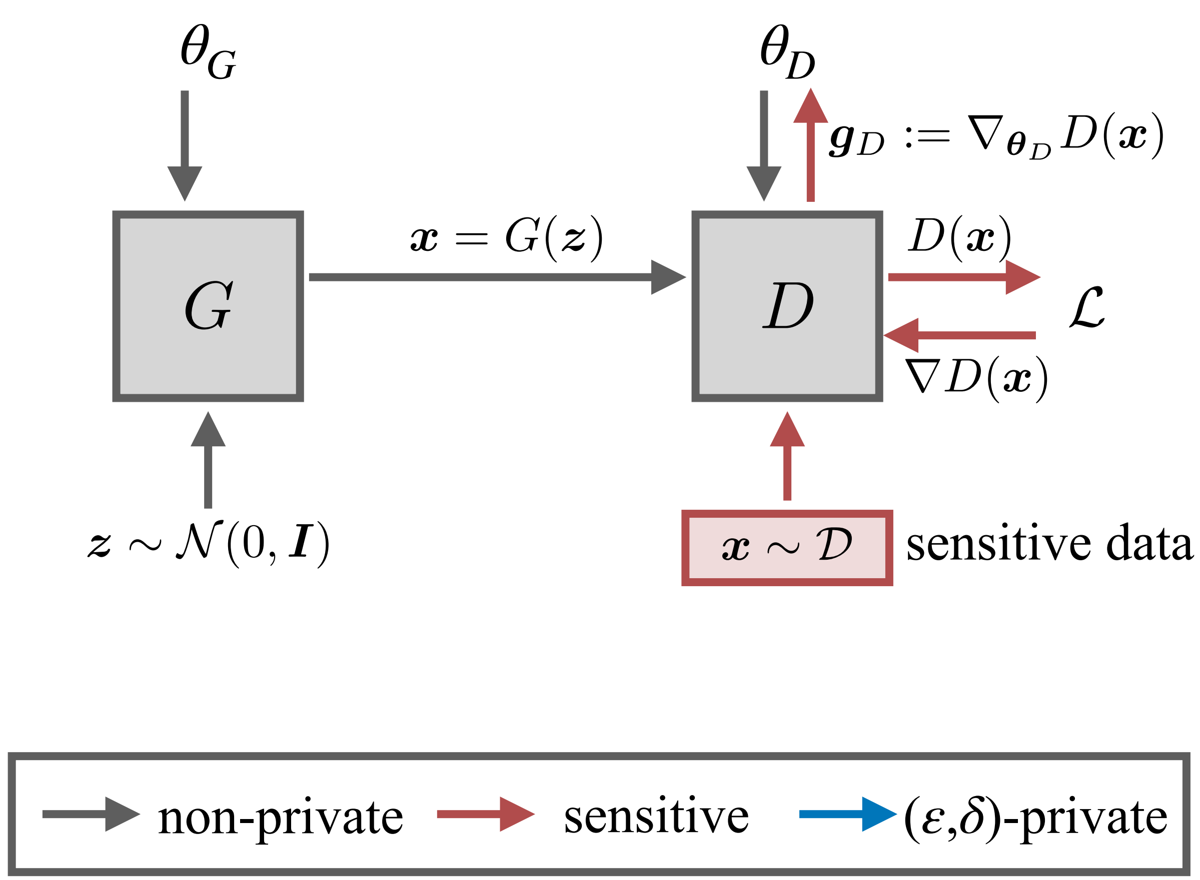
Approach



[1] Arjovsky et al., "Wasserstein Generative Adversarial Network", ICML 2017
[2] Gulrajani et al., "Improved Training of Wasserstein GANs", NIPS 2017

Approach

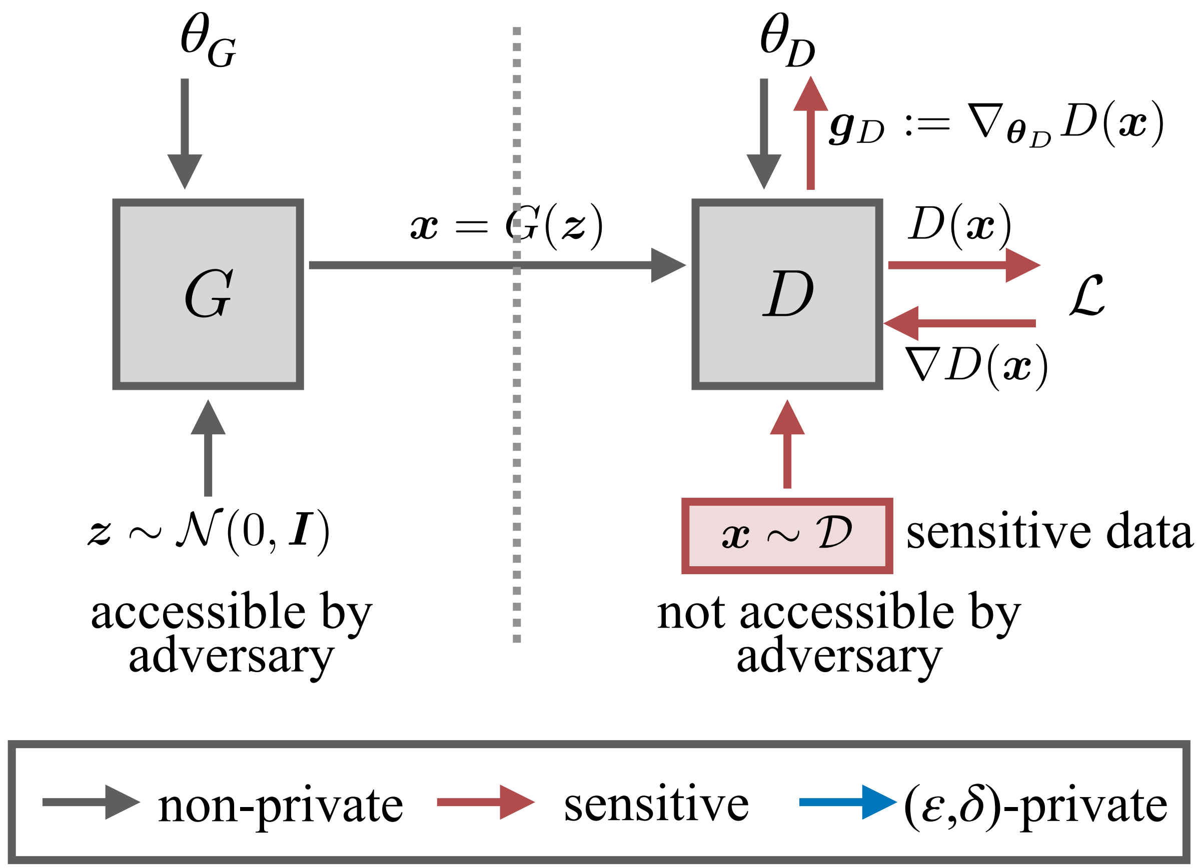
- Insight:
  - Only the *generator* need to be publicly-released



[1] Arjovsky et al., “Wasserstein Generative Adversarial Network”, ICML 2017
[2] Gulrajani et al., “Improved Training of Wasserstein GANs”, NIPS 2017

Approach

- Insight:
    - Only the *generator* need to be publicly-released

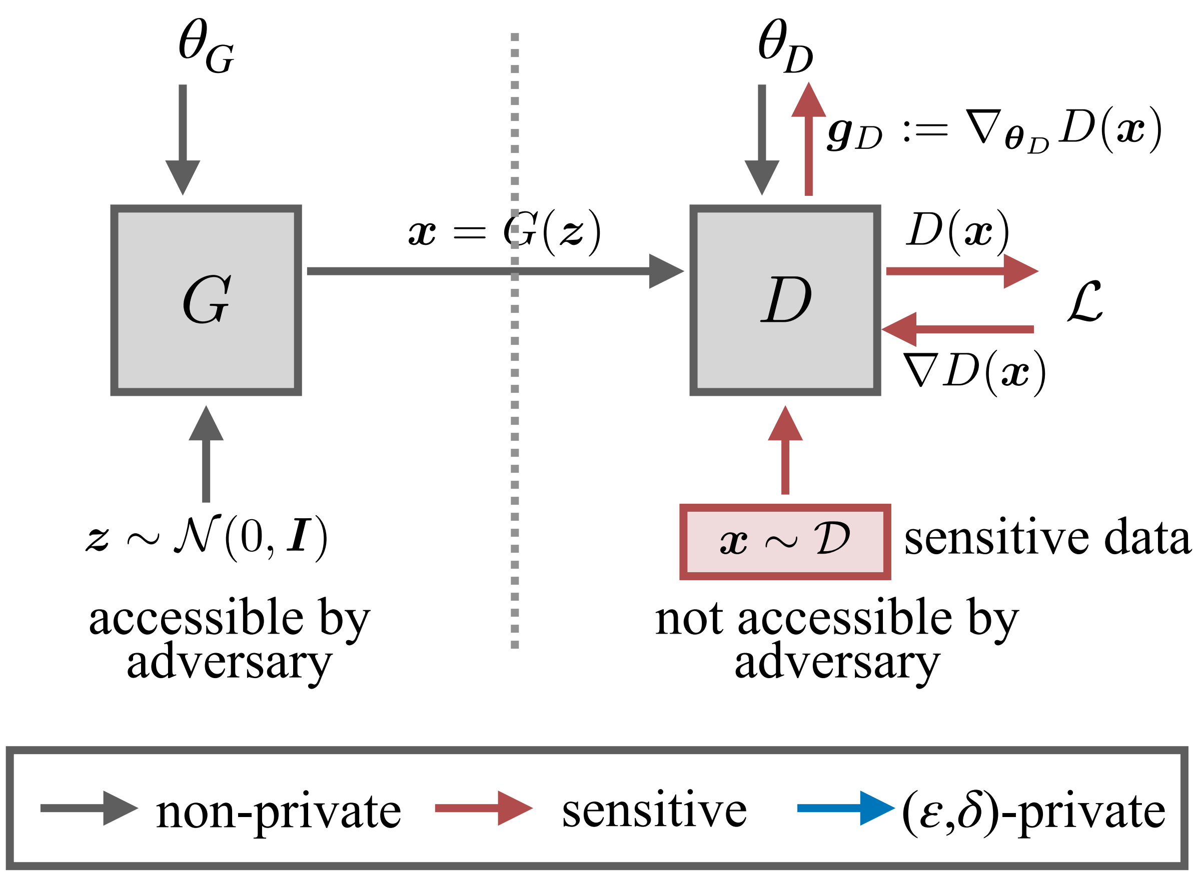


[1] Arjovsky et al., "Wasserstein Generative Adversarial Network", ICML 2017
[2] Gulrajani et al., "Improved Training of Wasserstein GANs", NIPS 2017

Approach

- Insight:
  - Only the *generator* need to be publicly-released

- Our framework:
  1. Selectively applying sanitization mechanism

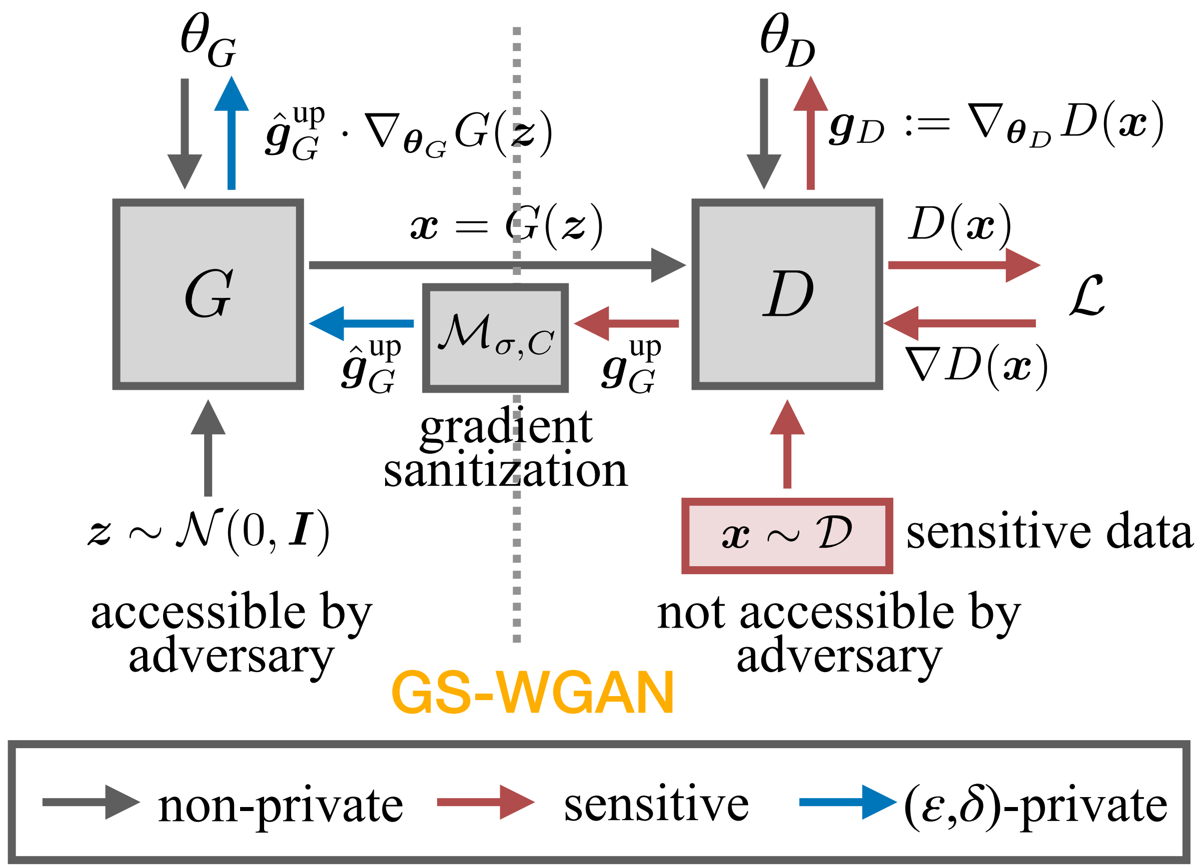


[1] Arjovsky et al., "Wasserstein Generative Adversarial Network", ICML 2017
[2] Gulrajani et al., "Improved Training of Wasserstein GANs", NIPS 2017

Approach

- Insight:
  - Only the *generator* need to be publicly-released

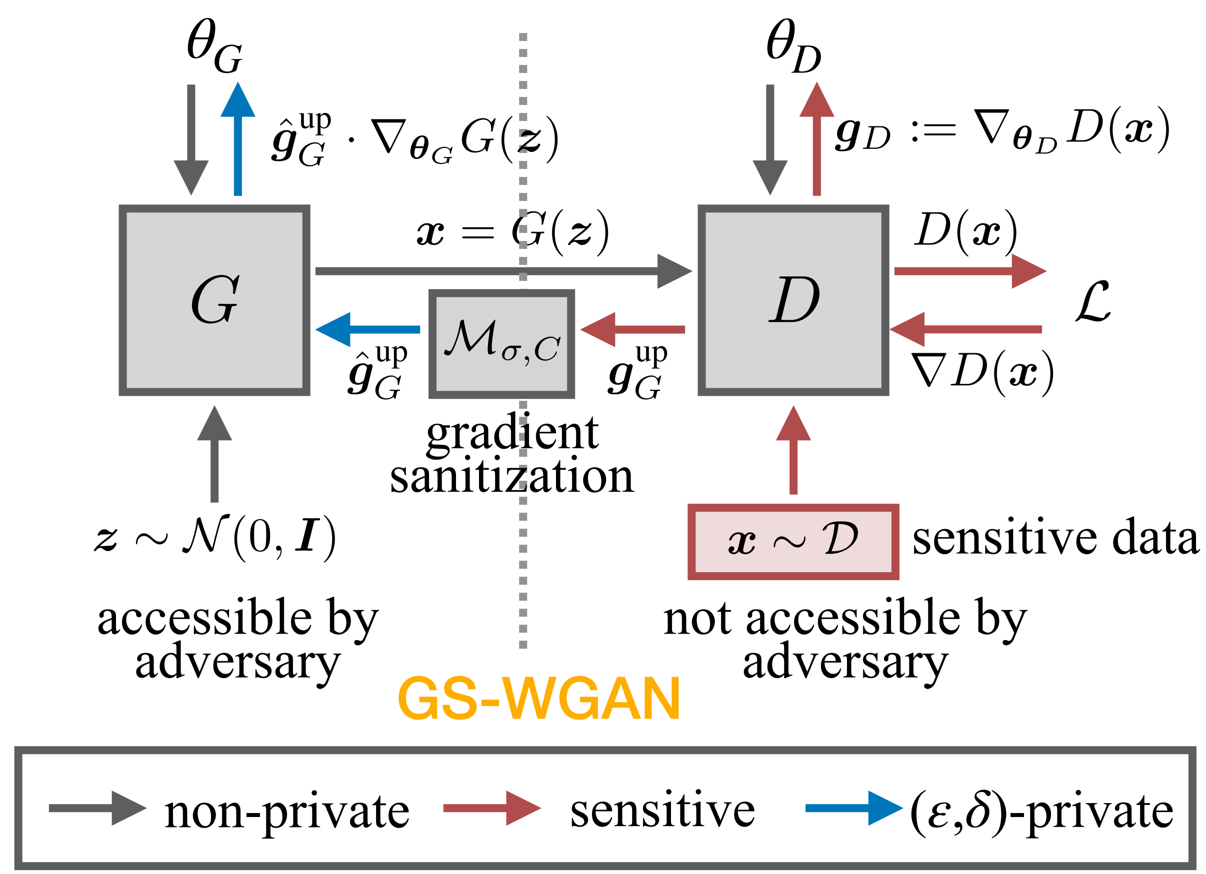
- Our framework:
  1. Selectively applying sanitization mechanism



[1] Arjovsky et al., "Wasserstein Generative Adversarial Network", ICML 2017
[2] Gulrajani et al., "Improved Training of Wasserstein GANs", NIPS 2017

Approach

- Insight:
  - Only the *generator* need to be publicly-released
- Our framework:
  1. Selectively applying sanitization mechanism



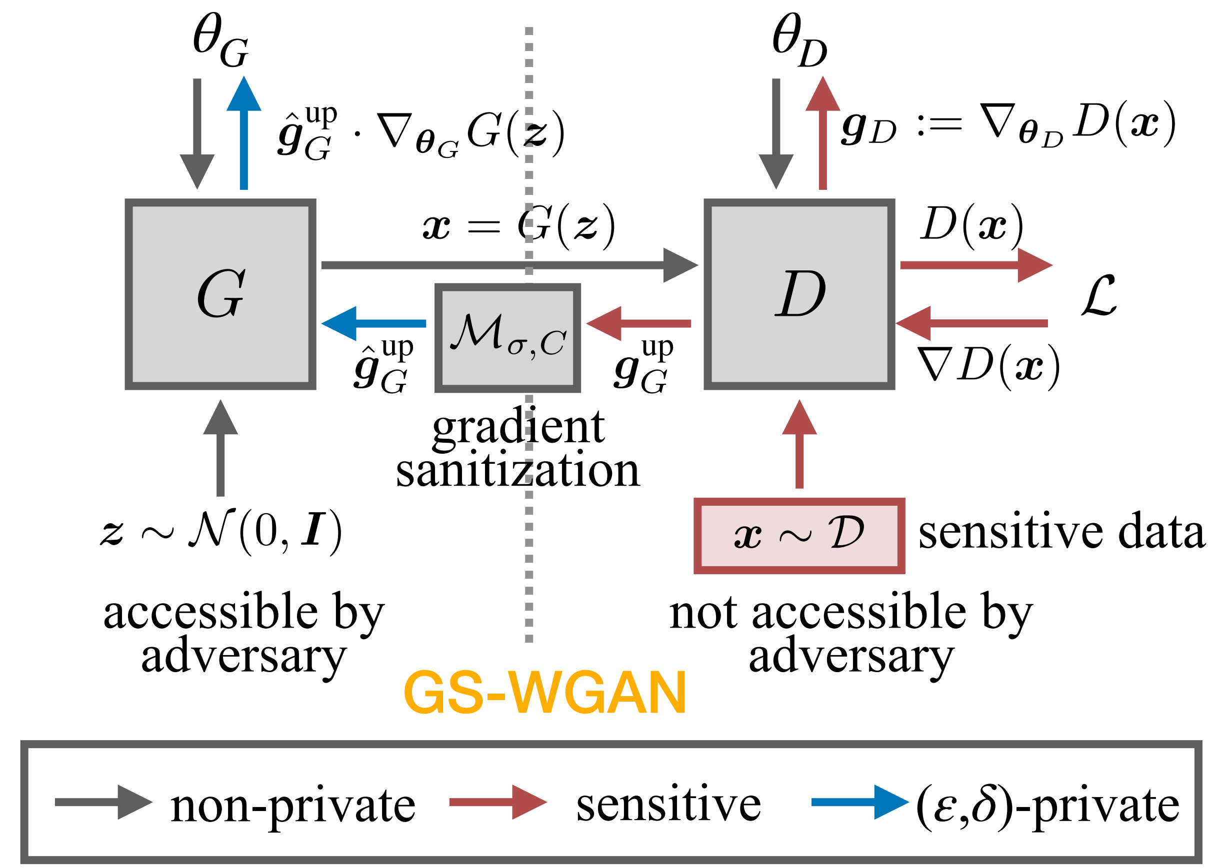
[1] Arjovsky et al., "Wasserstein Generative Adversarial Network", ICML 2017
[2] Gulrajani et al., "Improved Training of Wasserstein GANs", NIPS 2017

Approach

- Insight:
  - Only the *generator* need to be publicly-released

- Our framework:
  1. Selectively applying sanitization mechanism

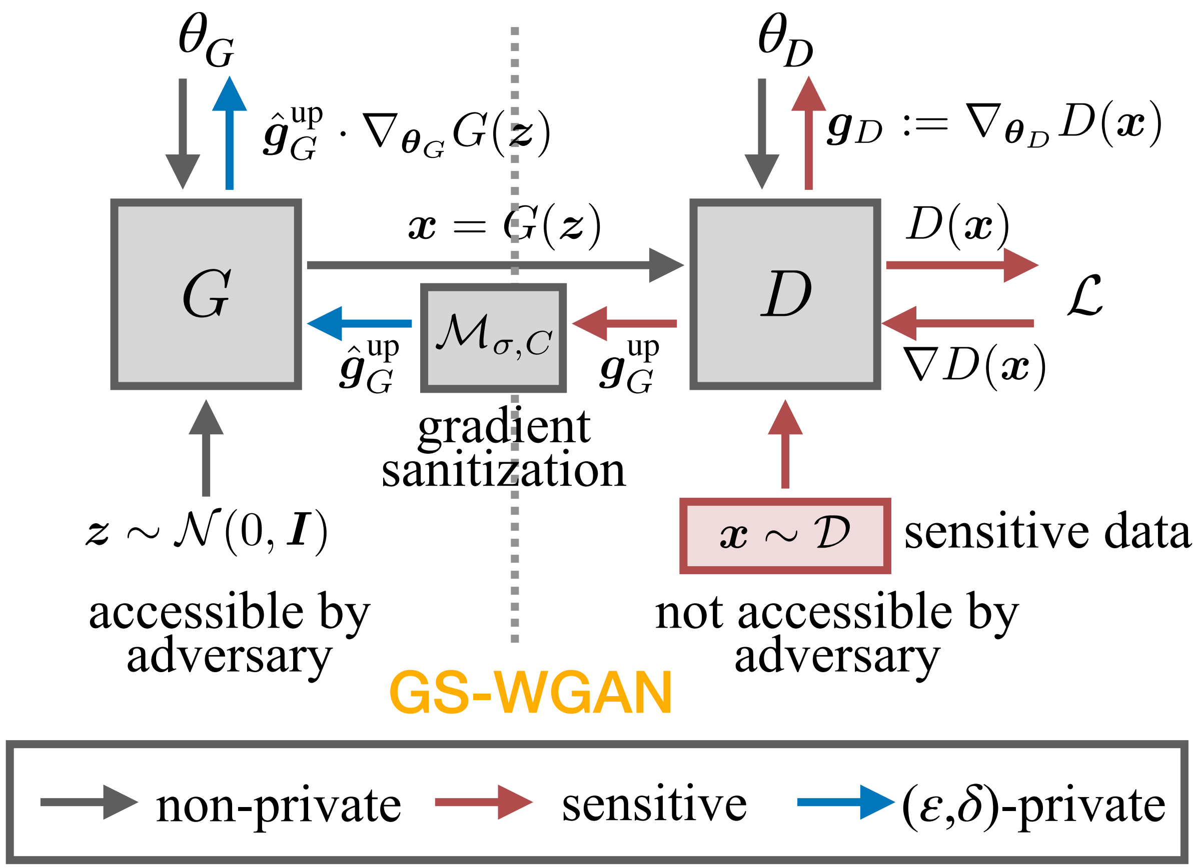
- Advantages:
  1. Maximally preserve the true gradient direction



[1] Arjovsky et al., “Wasserstein Generative Adversarial Network”, ICML 2017
[2] Gulrajani et al., “Improved Training of Wasserstein GANs”, NIPS 2017

Approach

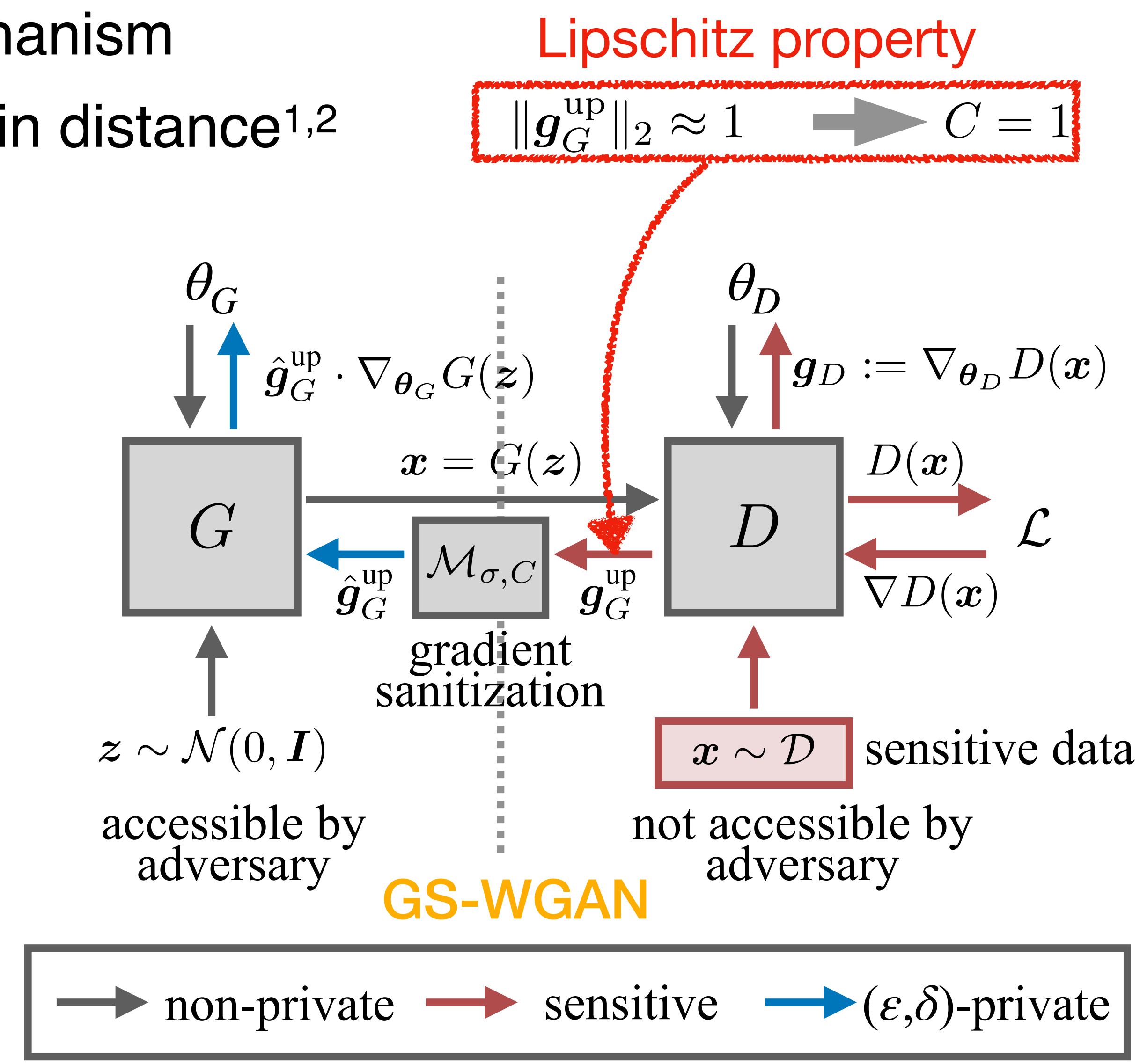
- Insight:
  - Only the *generator* need to be publicly-released
- Our framework:
  1. Selectively applying sanitization mechanism
  2. Bounding sensitivity using Wasserstein distance[1,2]
- Advantages:
  1. Maximally preserve the true gradient direction



[1] Arjovsky et al., "Wasserstein Generative Adversarial Network", ICML 2017
[2] Gulrajani et al., "Improved Training of Wasserstein GANs", NIPS 2017

Approach

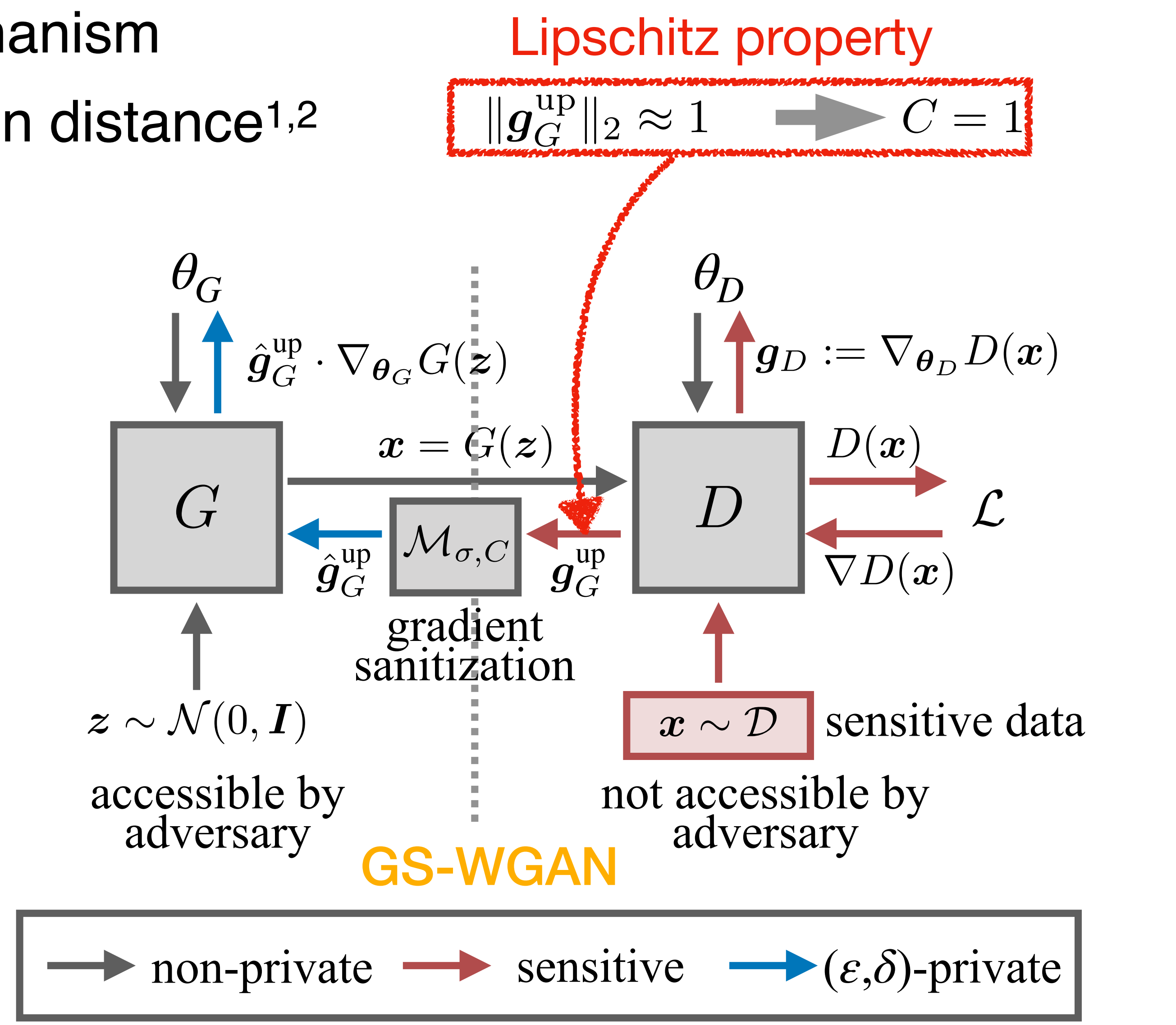
- Insight:
  - Only the *generator* need to be publicly-released
- Our framework:
  1. Selectively applying sanitization mechanism
  2. Bounding sensitivity using Wasserstein distance[1,2]
- Advantages:
  1. Maximally preserve the true gradient direction



[1] Arjovsky et al., "Wasserstein Generative Adversarial Network", ICML 2017
[2] Gulrajani et al., "Improved Training of Wasserstein GANs", NIPS 2017

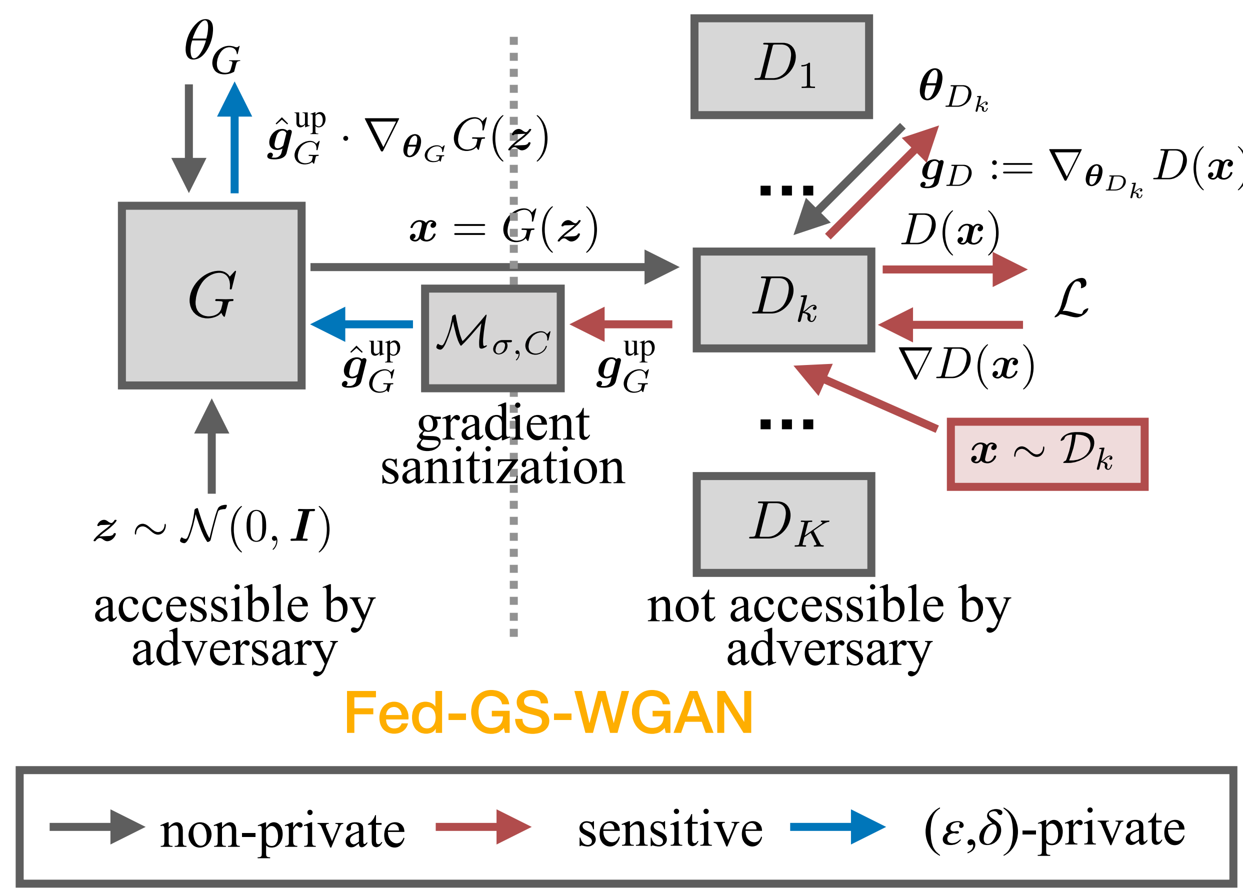
Approach

- Insight:
  - Only the *generator* need to be publicly-released
- Our framework:
  1. Selectively applying sanitization mechanism
  2. Bounding sensitivity using Wasserstein distance[1,2]
- Advantages:
  1. Maximally preserve the true gradient direction
  2. Bypass an intensive and fragile hyper-parameter search for clipping value
  3. Small clipping bias



[1] Arjovsky et al., “Wasserstein Generative Adversarial Network”, ICML 2017
[2] Gulrajani et al., “Improved Training of Wasserstein GANs”, NIPS 2017

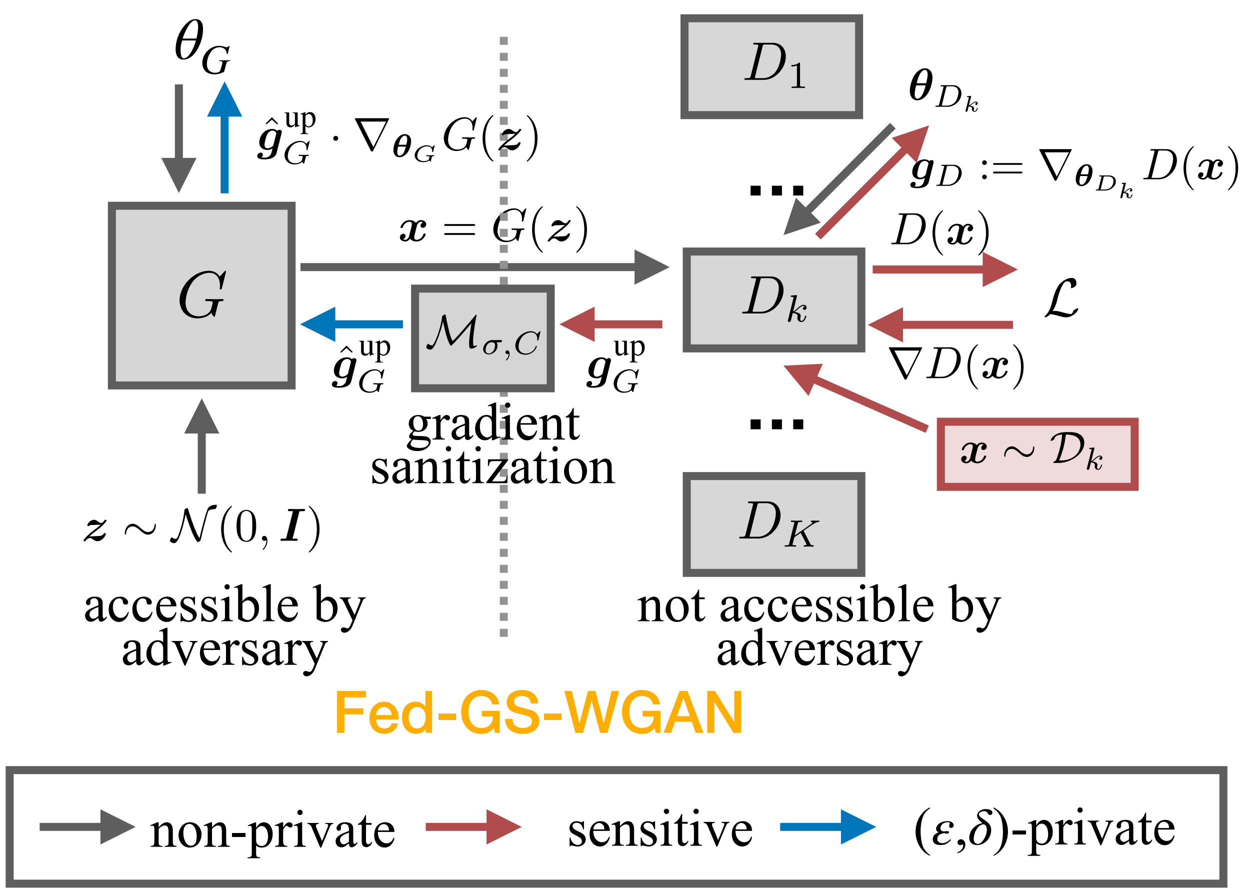
Approach



[1] Augenstein et al., "Generative Models for Effective ML on Private, Decentralized Datasets", ICLR 2020

Approach

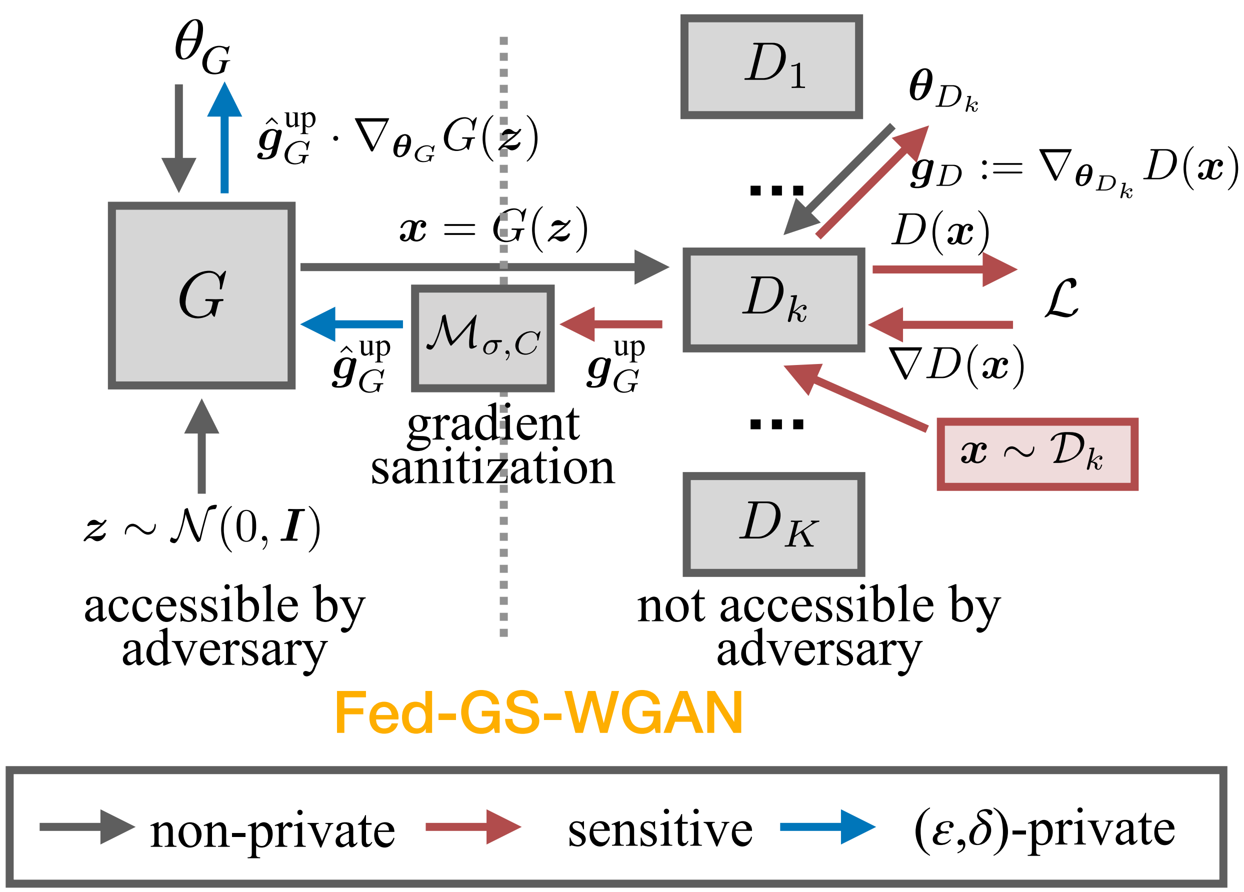
- Decentralized (Federated) setting
  - Each user train a discriminator on its sensitive dataset locally



[1] Augenstein et al., "Generative Models for Effective ML on Private, Decentralized Datasets", ICLR 2020

Approach

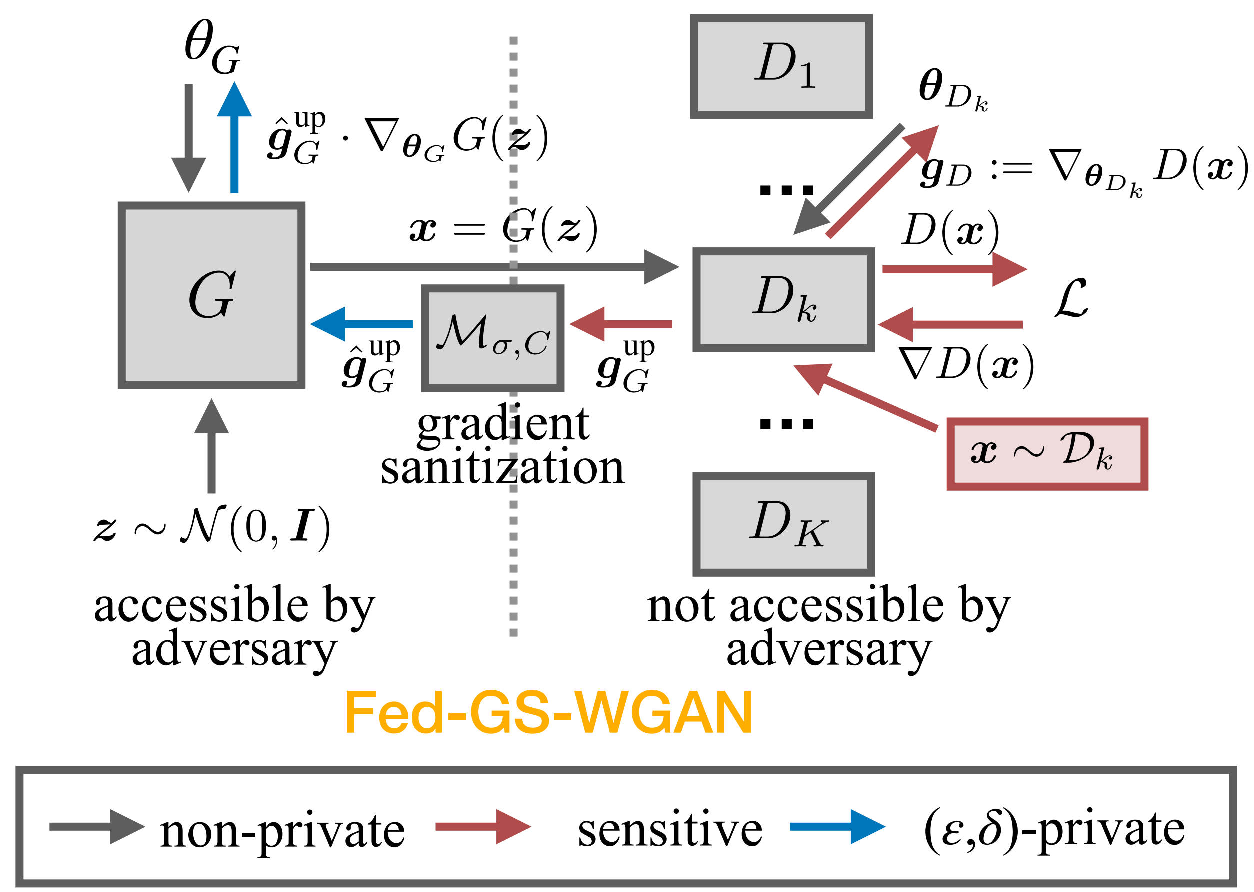
- Decentralized (Federated) setting
  - Each user train a discriminator on its sensitive dataset locally



[1] Augenstein et al., "Generative Models for Effective ML on Private, Decentralized Datasets", ICLR 2020

Approach

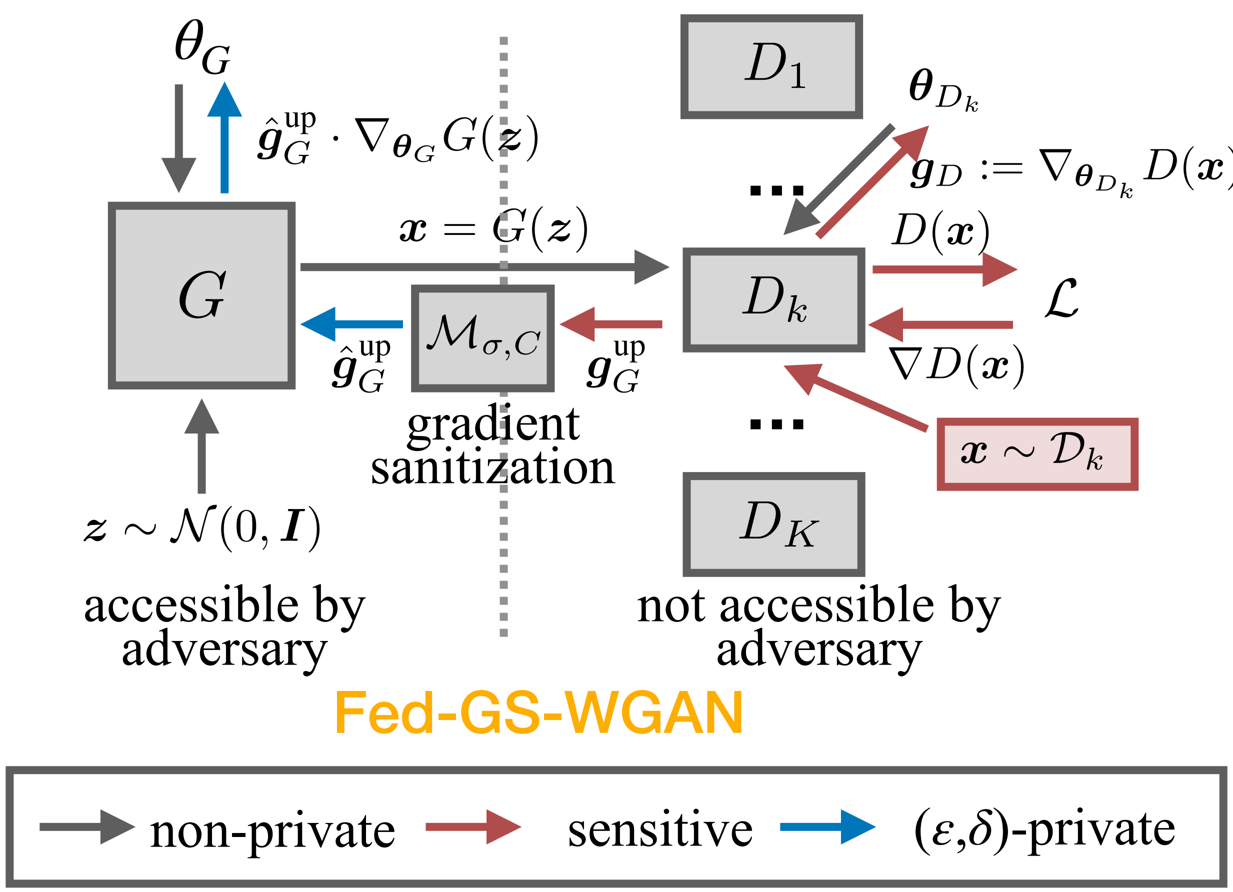
- Decentralized (Federated) setting
  - Each user train a discriminator on its sensitive dataset locally
  - Communicate the sanitized gradient



[1] Augenstein et al., “Generative Models for Effective ML on Private, Decentralized Datasets”, ICLR 2020

Approach

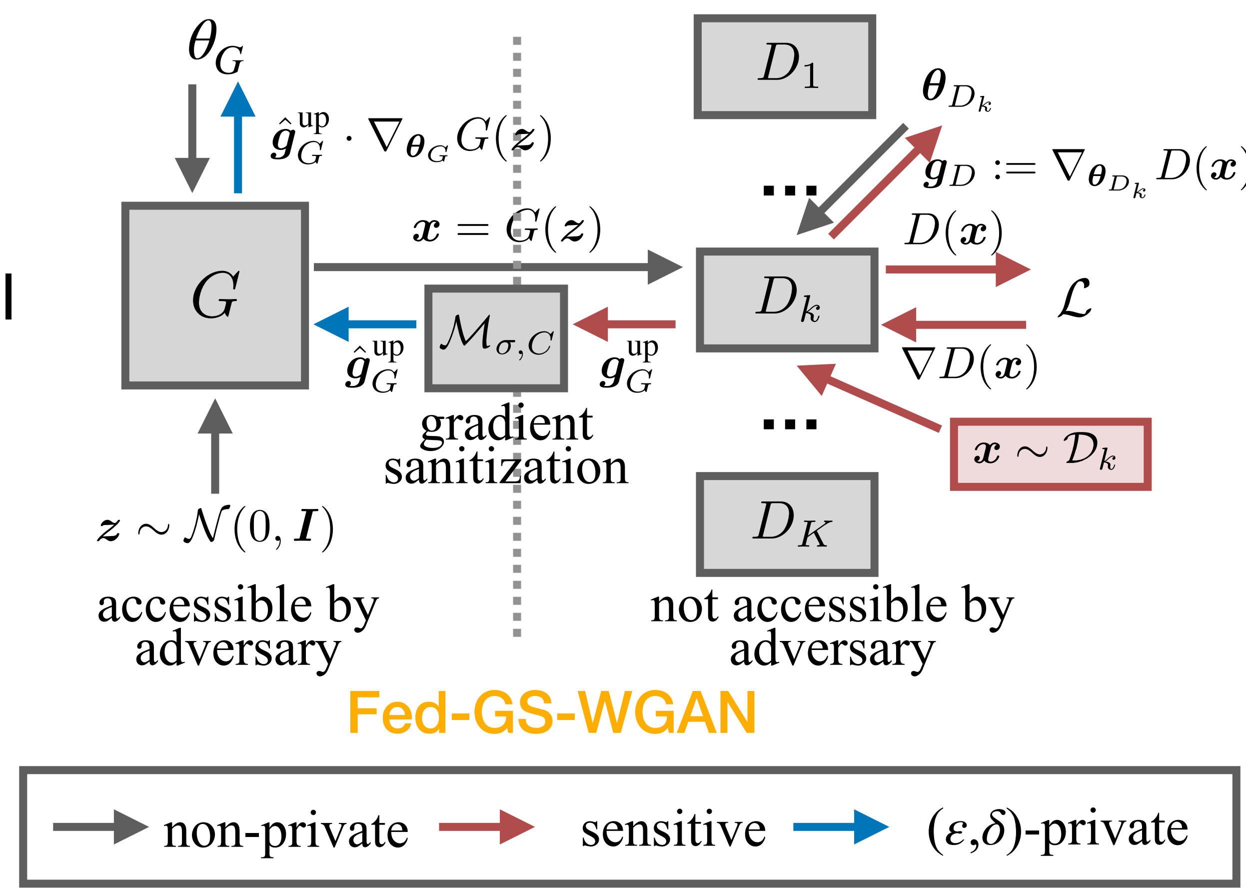
- Decentralized (Federated) setting
  - Each user train a discriminator on its sensitive dataset locally
  - Communicate the sanitized gradient



[1] Augenstein et al., "Generative Models for Effective ML on Private, Decentralized Datasets", ICLR 2020

Approach

- Decentralized (Federated) setting
  - Each user train a discriminator on its sensitive dataset locally
  - Communicate the sanitized gradient
- Advantages:
  - User-level DP guarantee under an *untrusted* server
  - Communication-efficient (gradients w.r.t. generated samples are *more compact* than gradients w.r.t model parameters[1])



[1] Augenstein et al., "Generative Models for Effective ML on Private, Decentralized Datasets", ICLR 2020

Evaluation

- Datasets
  - Images (MNIST, Fashion-MNIST, Fed-EMNIST)

- Evaluation metrics
  - **Privacy**: Determined by ε with fixed δ
  - **Utility**:
    - Sample quality: realism of the generated samples
      - Inception score (IS)[1,2], Frechet Inception Distance (FID)[3]
    - Usefulness for downstream tasks:
      - Classification accuracy: MLP Acc, CNN Acc, Avg Acc, Calibrated Acc (trained on generated data and test on real data)

[1] Li et al., "Alice: Towards Understanding Adversarial Learning for Joint Distribution Matching", NIPS 2017
[2] Salimans et al., "Improved Techniques for Training GANs", NIPS 2016
[3] Heusel et al., "GANs Trained by a Two Time-scale Update Rule Converge to a Local Nash Equilibrium", NIPS 2017

Results

		IS↑	FID ↓	MLP ↑ Acc	CNN ↑ Acc	Avg ↑ Acc	Calibrated ↑ Acc
MNIST	Real	9.80	1.02	0.98	0.99	0.88	100 %
	G-PATE [1]	3.85	177.16	0.25	0.51	0.34	40%
	DP-SGD GAN	4.76	179.16	0.60	0.63	0.52	59%
	DP-Merf	2.91	247.53	0.63	0.63	0.57	66%
	DP-Merf AE	3.06	161.11	0.54	0.68	0.42	47%
	Ours	**9.23**	**61.34**	**0.79**	**0.80**	**0.60**	**69%**
Fashion-MNIST	Real	8.98	1.49	0.88	0.91	0.79	100%
	G-PATE	3.35	205.78	0.30	0.50	0.40	54%
	DP-SGD GAN	3.55	243.80	0.50	0.46	0.43	53%
	DP-Merf	2.32	267.78	0.56	0.62	0.51	65%
	DP-Merf AE	3.68	213.59	0.56	0.62	0.45	55%
	Ours	**5.32**	**131.34**	**0.65**	**0.65**	**0.53**	**67%**

Table 1: Quantitative Results on MNIST and Fashion-MNIST ($\varepsilon = 10, \delta = 10^{-5}$)

	IS ↑	FID ↓	epsilon ↓	CT (byte) ↓
Fed Avg GAN	10.88	218.24	9.99×10^6	$\sim 3.94 \times 10^7$
Ours	**11.25**	**60.76**	$\mathbf{5.99 \times 10^2}$	$\mathbf{\sim 1.50 \times 10^5}$

Table 4: Quantitative Results on Federated EMNIST ($\delta = 1.15 \times 10^{-3}$)

Results

- Centralized setting

		IS↑	FID ↓	MLP ↑ Acc	CNN ↑ Acc	Avg ↑ Acc	Calibrated ↑ Acc
MNIST	Real	9.80	1.02	0.98	0.99	0.88	100 %
	G-PATE [1]	3.85	177.16	0.25	0.51	0.34	40%
	DP-SGD GAN	4.76	179.16	0.60	0.63	0.52	59%
	DP-Merf	2.91	247.53	0.63	0.63	0.57	66%
	DP-Merf AE	3.06	161.11	0.54	0.68	0.42	47%
	Ours	**9.23**	**61.34**	**0.79**	**0.80**	**0.60**	**69%**
Fashion-MNIST	Real	8.98	1.49	0.88	0.91	0.79	100%
	G-PATE	3.35	205.78	0.30	0.50	0.40	54%
	DP-SGD GAN	3.55	243.80	0.50	0.46	0.43	53%
	DP-Merf	2.32	267.78	0.56	0.62	0.51	65%
	DP-Merf AE	3.68	213.59	0.56	0.62	0.45	55%
	Ours	**5.32**	**131.34**	**0.65**	**0.65**	**0.53**	**67%**

Table 1: Quantitative Results on MNIST and Fashion-MNIST ($\varepsilon = 10, \delta = 10^{-5}$)

- Decentralized (Federated) setting

	IS ↑	FID ↓	epsilon ↓	CT (byte) ↓
Fed Avg GAN	10.88	218.24	9.99×10^6	$\sim 3.94 \times 10^7$
Ours	**11.25**	**60.76**	$\mathbf{5.99 \times 10^2}$	$\mathbf{\sim 1.50 \times 10^5}$

Table 4: Quantitative Results on Federated EMNIST ($\delta = 1.15 \times 10^{-3}$)

Results

- Centralized setting

  Improves the IS by:

  - 94% on MNIST
  - 45% on Fashion-MNIST

  Improves the MLP Acc by:

  - 25% on MNIST
  - 16% on Fashion-MNIST

		IS↑	FID ↓	MLP ↑ Acc	CNN ↑ Acc	Avg ↑ Acc	Calibrated ↑ Acc
MNIST	Real	9.80	1.02	0.98	0.99	0.88	100 %
	G-PATE [1]	3.85	177.16	0.25	0.51	0.34	40%
	DP-SGD GAN	4.76	179.16	0.60	0.63	0.52	59%
	DP-Merf	2.91	247.53	0.63	0.63	0.57	66%
	DP-Merf AE	3.06	161.11	0.54	0.68	0.42	47%
	Ours	**9.23**	**61.34**	**0.79**	**0.80**	**0.60**	**69%**
Fashion-MNIST	Real	8.98	1.49	0.88	0.91	0.79	100%
	G-PATE	3.35	205.78	0.30	0.50	0.40	54%
	DP-SGD GAN	3.55	243.80	0.50	0.46	0.43	53%
	DP-Merf	2.32	267.78	0.56	0.62	0.51	65%
	DP-Merf AE	3.68	213.59	0.56	0.62	0.45	55%
	Ours	**5.32**	**131.34**	**0.65**	**0.65**	**0.53**	**67%**

Table 1: Quantitative Results on MNIST and Fashion-MNIST ($\varepsilon = 10, \delta = 10^{-5}$)

- Decentralized (Federated) setting

  Better *sample quality*:

  - 0.28x smaller FID

  Lower *privacy cost*:

  - 10^4× smaller epsilon

	IS ↑	FID ↓	epsilon ↓	CT (byte) ↓
Fed Avg GAN	10.88	218.24	9.99×10^6	$\sim 3.94 \times 10^7$
Ours	**11.25**	**60.76**	$\mathbf{5.99 \times 10^2}$	$\mathbf{\sim 1.50 \times 10^5}$

Table 4: Quantitative Results on Federated EMNIST ($\delta = 1.15 \times 10^{-3}$)

Results

- Centralized setting

  Improves the IS by:
  - 94% on MNIST
  - 45% on Fashion-MNIST

  Improves the MLP Acc by:
  - 25% on MNIST
  - 16% on Fashion-MNIST

		IS↑	FID ↓	MLP ↑ Acc	CNN ↑ Acc	Avg ↑ Acc	Calibrated ↑ Acc
MNIST	Real	9.80	1.02	0.98	0.99	0.88	100 %
	G-PATE [1]	3.85	177.16	0.25	0.51	0.34	40%
	DP-SGD GAN	4.76	179.16	0.60	0.63	0.52	59%
	DP-Merf	2.91	247.53	0.63	0.63	0.57	66%
	DP-Merf AE	3.06	161.11	0.54	0.68	0.42	47%
	Ours	**9.23**	**61.34**	**0.79**	**0.80**	**0.60**	**69%**
Fashion-MNIST	Real	8.98	1.49	0.88	0.91	0.79	100%
	G-PATE	3.35	205.78	0.30	0.50	0.40	54%
	DP-SGD GAN	3.55	243.80	0.50	0.46	0.43	53%
	DP-Merf	2.32	267.78	0.56	0.62	0.51	65%
	DP-Merf AE	3.68	213.59	0.56	0.62	0.45	55%
	Ours	**5.32**	**131.34**	**0.65**	**0.65**	**0.53**	**67%**

Table 1: Quantitative Results on MNIST and Fashion-MNIST ($\varepsilon = 10, \delta = 10^{-5}$)

- Decentralized (Federated) setting

  Better *sample quality*:
  - 0.28x smaller FID

  Lower *privacy cost*:
  - 10^4x smaller epsilon

	IS ↑	FID ↓	epsilon ↓	CT (byte) ↓
Fed Avg GAN	10.88	218.24	9.99×10^6	$\sim 3.94 \times 10^7$
Ours	**11.25**	**60.76**	$\mathbf{5.99 \times 10^2}$	$\mathbf{\sim 1.50 \times 10^5}$

Table 4: Quantitative Results on Federated EMNIST ($\delta = 1.15 \times 10^{-3}$)

Consistent improvement over baselines across different datasets, settings and metrics

Results

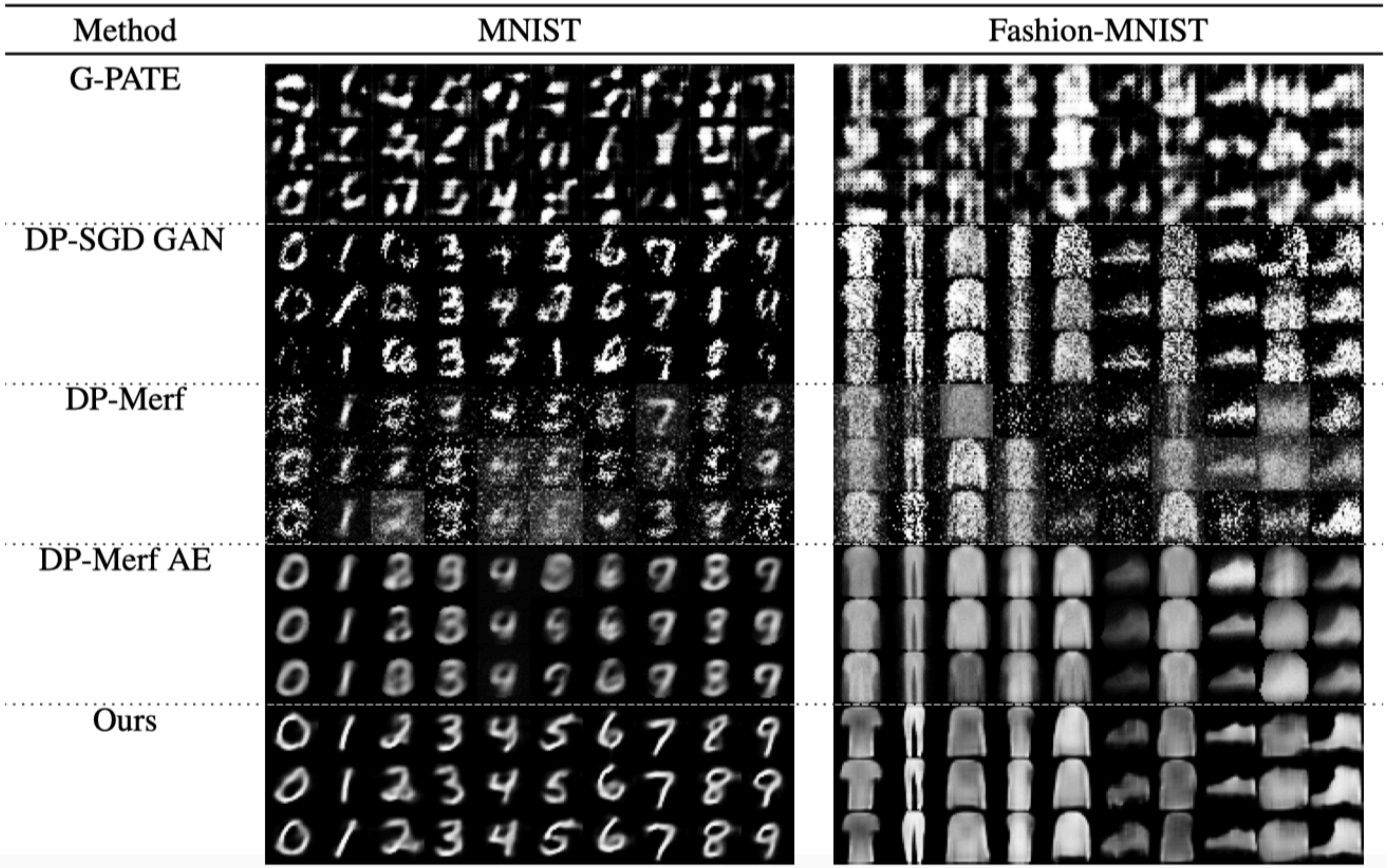


Figure 3: Generated samples with $(\varepsilon, \delta) = (10, 10^{-5})$

Results

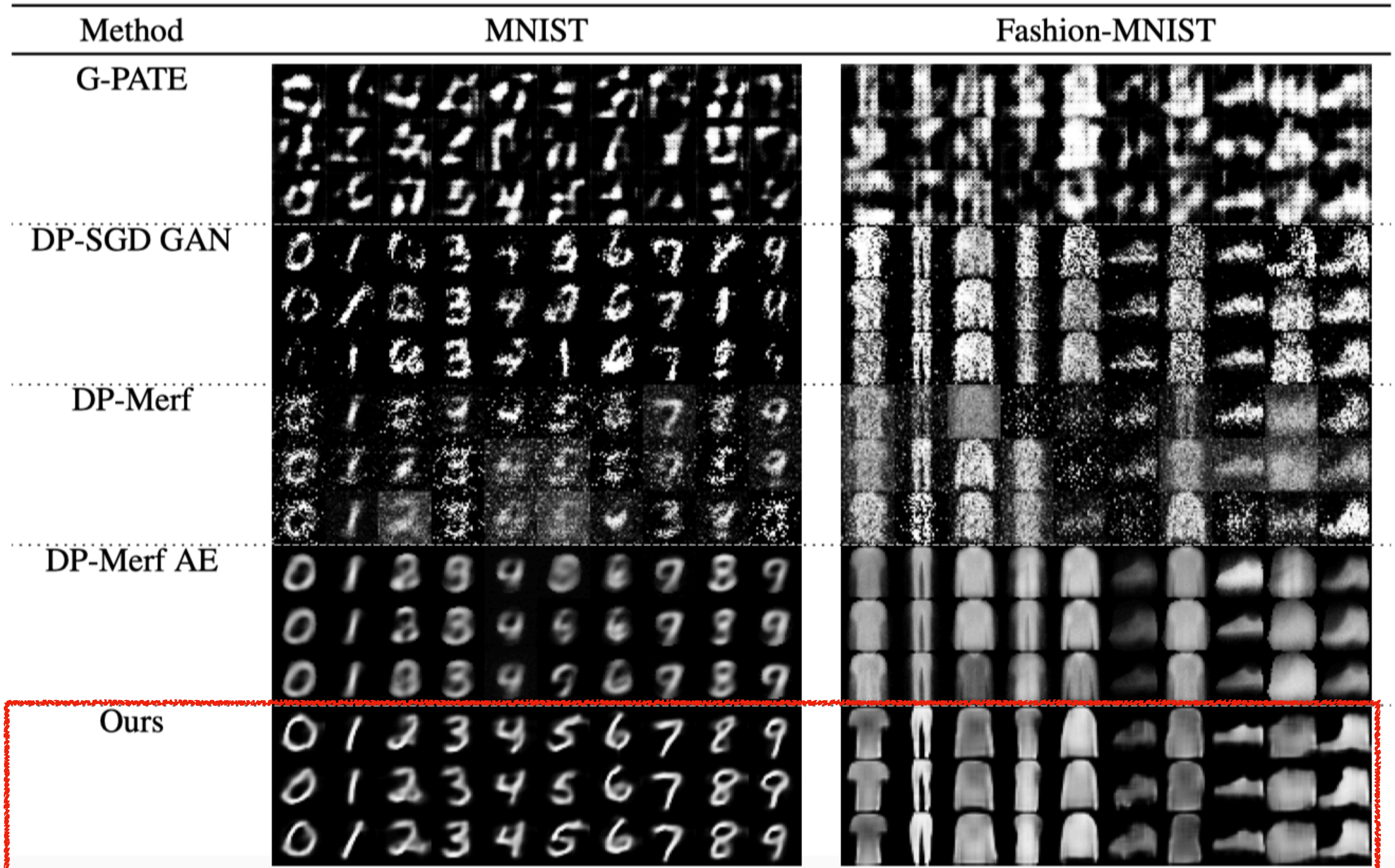


Figure 3: Generated samples with $(\varepsilon, \delta) = (10, 10^{-5})$

More details in the paper

GS-WGAN: A Gradient-Sanitized Approach for Learning Differentially Private Generators

Dingfan Chen[1] Tribhuvanesh Orekondy[2] Mario Fritz[1]

Code and Models are available on Github

https://github.com/DingfanChen/GS-WGAN

[1] CISPA Helmholtz Center for Information Security
[2] Max Planck Institute for Informatics