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

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

max planck institut informatik

NEURAL INFORMATION PROCESSING SYSTEMS

Motivation

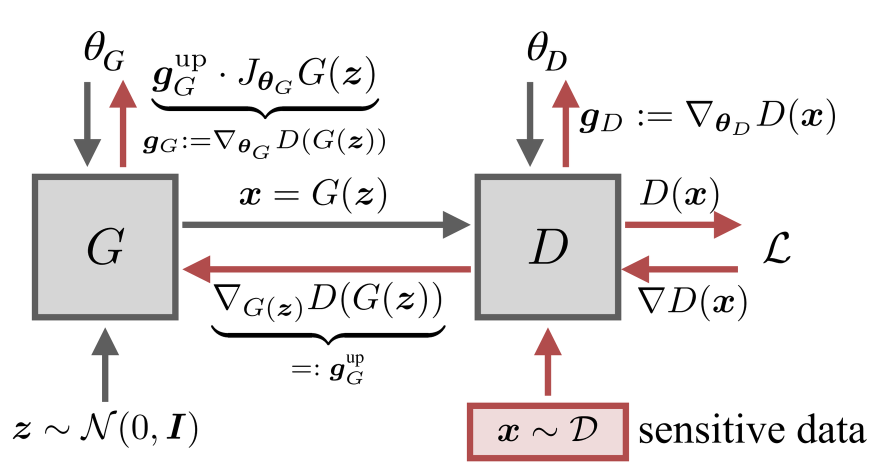
- Progress in training ML models in sensitive domains (e.g., healthcare) is impeded by scarcity of dataset
- Can we release synthetic datasets with rigorous privacy guarantees?

Task

- Privacy-preserving data generation
  - High-dimensional data
  - Arbitrary downstream task
    - } → Generative Adversarial Networks (GANs)[1]
  - Rigorous privacy guarantee → Differential Privacy (DP)[2]

Problem

- Existing Approach: Differentially private stochastic gradient descent (**DP-SGD**)[3]
  - Sanitize gradients before performing descent step
  - **Sanitization** includes:
    - *Clipping* the gradients
    - Adding calibrated *random noise*
  - However, selecting a proper clipping bound is difficult in practice:
    - Require intensive hyper-parameters search
    - Introduce high clipping bias



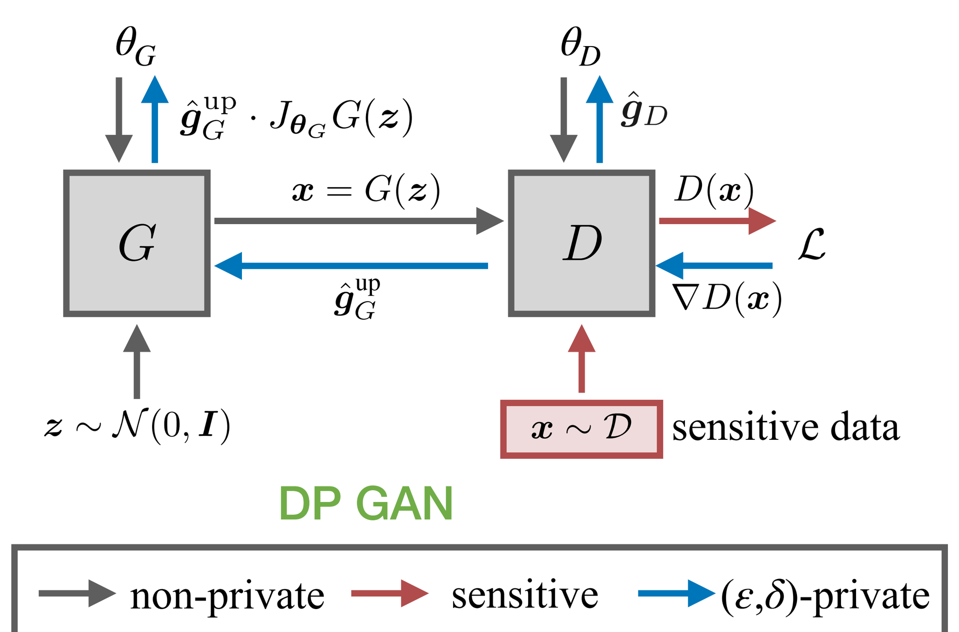
Vanilla GAN

Vanilla GAN

- Gradient
  $g^{(t)} := \nabla_{\theta} \mathcal{L}(\theta_D, \theta_G)$
- Gradient descent step
  $\theta^{(t+1)} := \theta^{(t)} - \eta \cdot g^{(t)}$

DPGAN

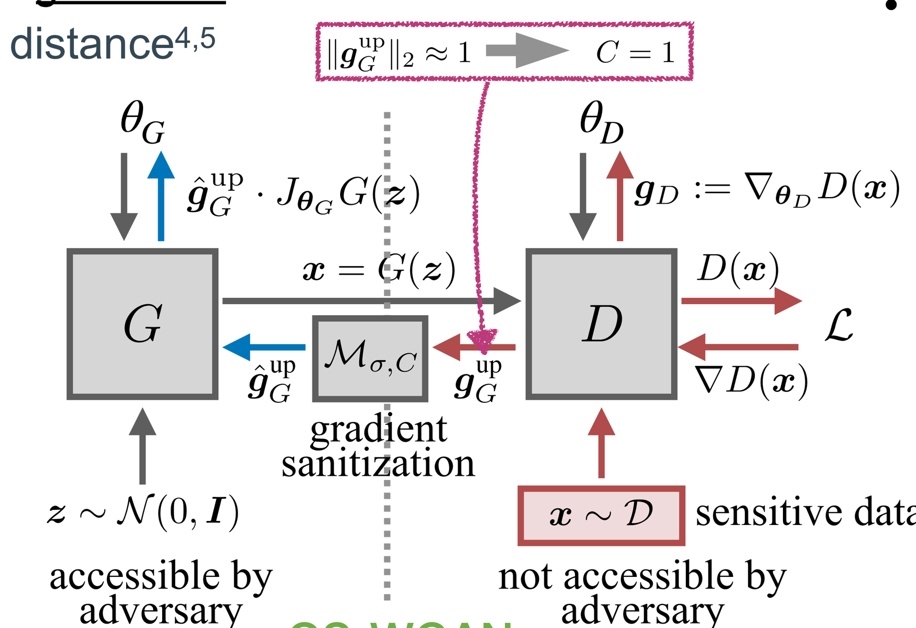
- Gradient
  $g^{(t)} := \nabla_{\theta} \mathcal{L}(\theta_D, \theta_G)$
- **Sanitization mechanism** (clipping bound)
  $\hat{g}^{(t)} := \mathcal{M}_{\sigma,C}(g^{(t)}) = \text{clip}(g^{(t)}, \boxed{C}) + \mathcal{N}(0, \sigma^2 C^2 I)$
- Gradient descent step
  $\theta^{(t+1)} := \theta^{(t)} - \eta \cdot \hat{g}^{(t)}$



DP GAN

Approach GS-WGAN (Gradient-sanitized Wasserstein GAN)

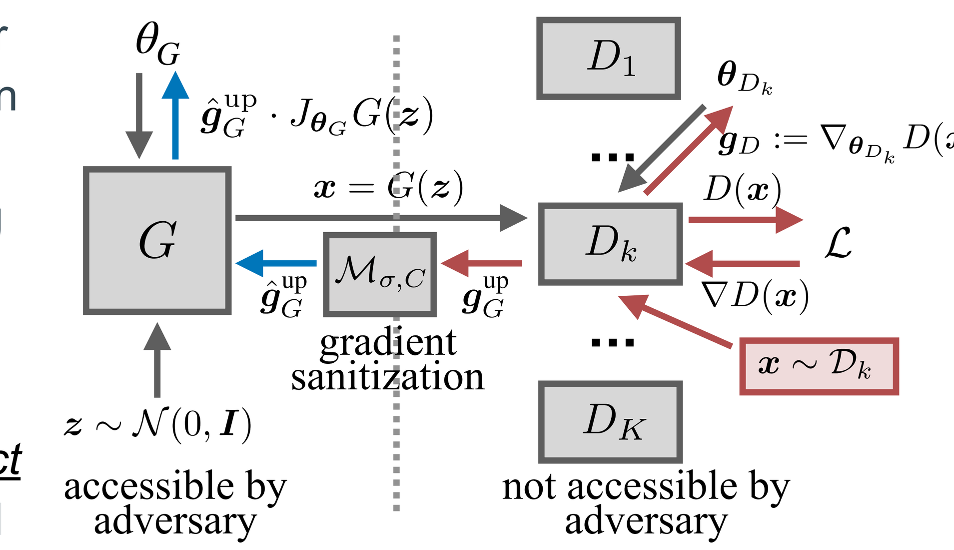
- **Insight:**
  - Only the *generator* need to be publicly-released
- **Our framework:**
  1. Selectively applying sanitization mechanism: $\hat{g}_G = \mathcal{M}_{\sigma,C}(\underbrace{\nabla_{G(z)} \mathcal{L}_G(\theta_G)}_{g_G^{\text{up}}}) \cdot \underbrace{J_{\theta_G} G(z; \theta_G)}_{J_G^{\text{local}}}$
     - Train the *discriminator* non-privately
     - Sanitize gradients transferred to the *generator*
  2. Bounding sensitivity using Wasserstein distance[4,5]
     - Lipschitz property
- **Advantage:**
  1. Maximally preserve the true gradient direction
  2. Bypass an intensive and fragile hyper-parameter search for the clipping bound
  3. Small clipping bias



GS-WGAN

Decentralized (Federated) setting: Fed-GS-WGAN

- **Our framework:**
  - Each user trains a discriminator on its sensitive dataset locally
  - The server maintains a generator trained with DP guarantee
  - Users send the sanitized gradients to the server, while receiving generated samples from the server
- **Advantage:**
  1. User-level DP guarantee under an *untrusted* server assumption
     - Gradients are sanitized at each client *before* sending to the server
  2. Communication-efficient
     - Gradients w.r.t. generated samples are *more compact* than gradients w.r.t. model parameters[6]



Fed-GS-WGAN

Evaluation

- **Datasets: Images** (MNIST, Fashion-MNIST, Fed-EMNIST)
- **Metrics:**
  - **Privacy**: Determined by ε with fixed δ
  - **Utility**:
    - Sample quality: realism of the generated samples
      - Inception score (IS), Frechet Inception Distance (FID)
    - Usefulness for downstream tasks:
      - Classification accuracy: (trained on generated data and test on real data) MLP Acc, CNN Acc, Avg Acc, Calibrated Acc

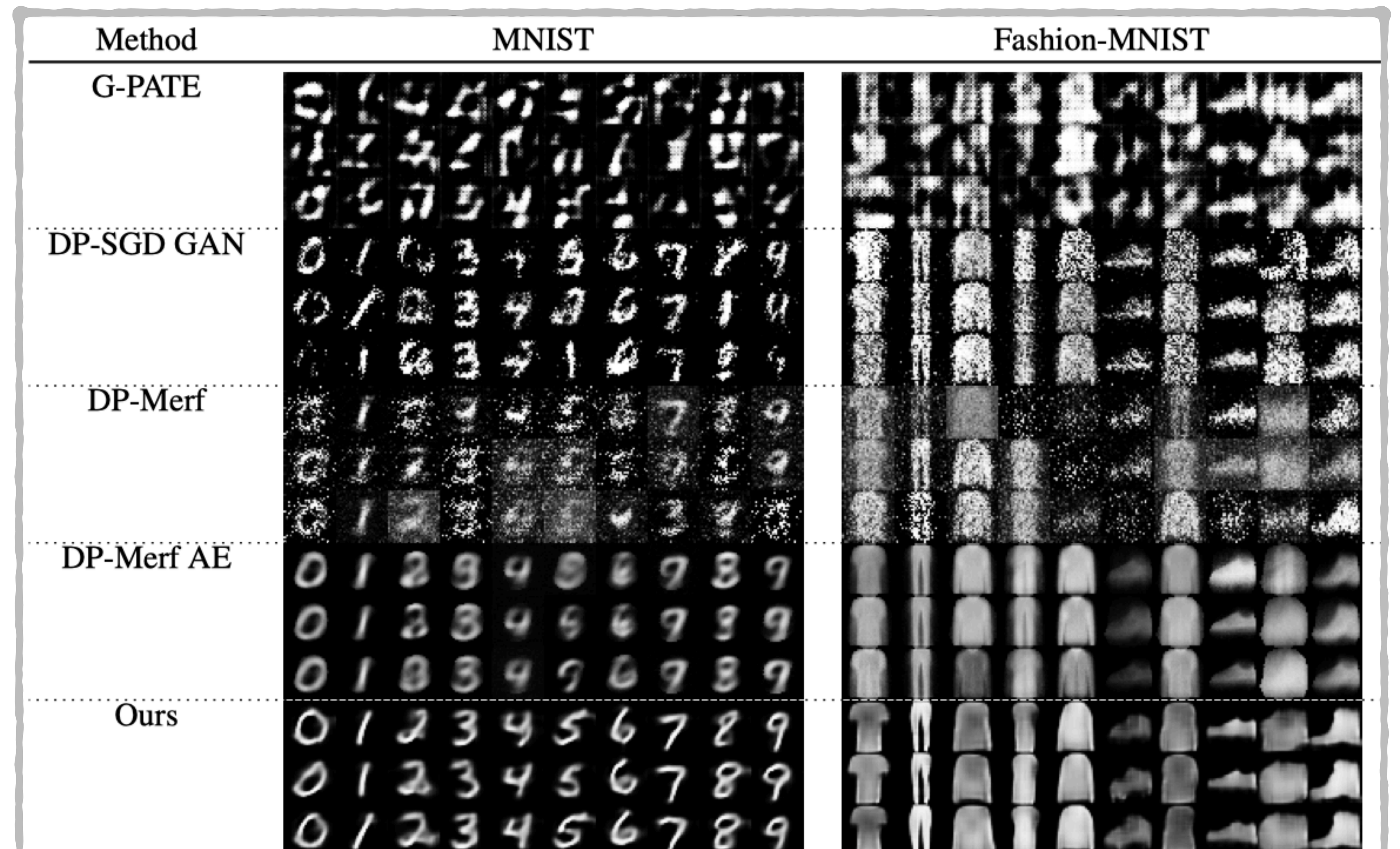


Figure 3: Generated samples with $(\varepsilon, \delta) = (10, 10^{-5})$

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

Improve communication efficiency:
- 10^2x gain in reducing CT

	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**	$\mathbf{5.99 \times 10^2}$	$\sim \mathbf{1.50 \times 10^5}$

Table 2: Quantitative Results on Federated EMNIST ($\delta = 1.15 \times 10^{-3}$)

More info: https://github.com/DingfanChen/GS-WGAN
(Source code and models are available)

References

[1] Goodfellow et al., "Generative Adversarial Nets". In: NIPS 2014.
[2] Dwork et al., "The Algorithmic Foundations of Differential Privacy". In: Foundations and Trends in Theoretical Computer Science.
[3] Abadi et al., "Deep Learning with Differential Privacy". In: CCS 2016.
[4] Arjovsky et al., "Wasserstein Generative Adversarial Network". In: ICML 2017.
[5] Gulrajani et al., "Improved Training of Wasserstein GANs". In: NIPS 2017.
[6] Augenstein et al., "Generative Models for Effective ML on Private, Decentralized Datasets". In: ICLR 2020.