

GDPP

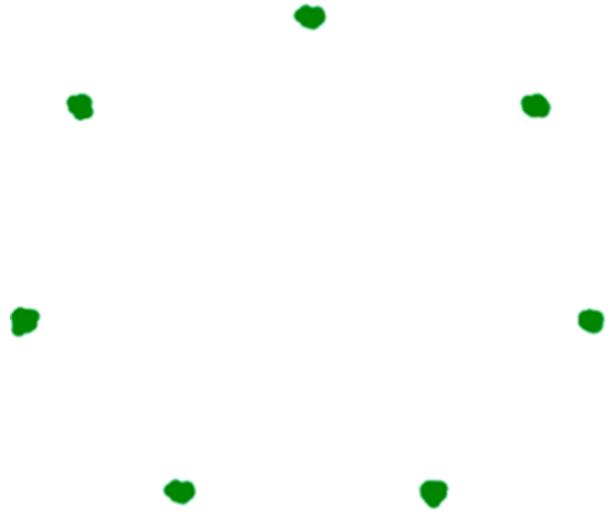
Learning Diverse Generations using Determinantal Point Process

Mohamed Elfeki, Camille Couprie,
Morgane Rivière and Mohamed Elhoseiny

* <https://github.com/M-Elfeki/GDPP>

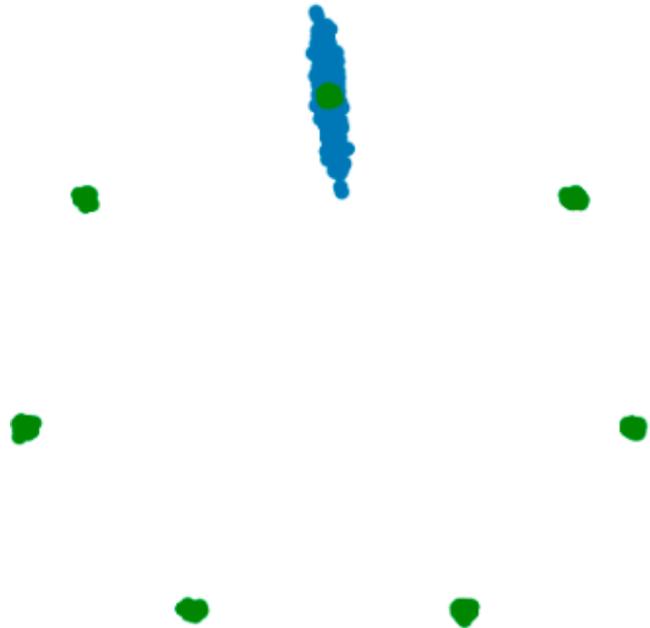


What's wrong with Generative models?



What's wrong with Generative models?

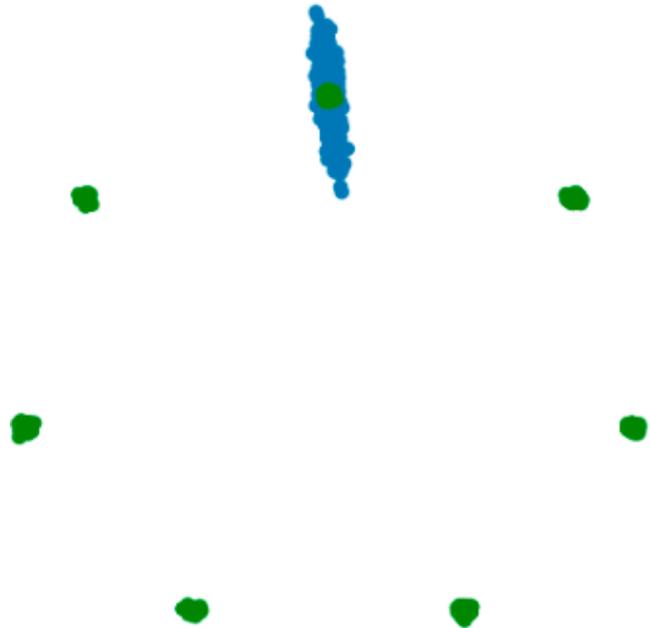
GAN



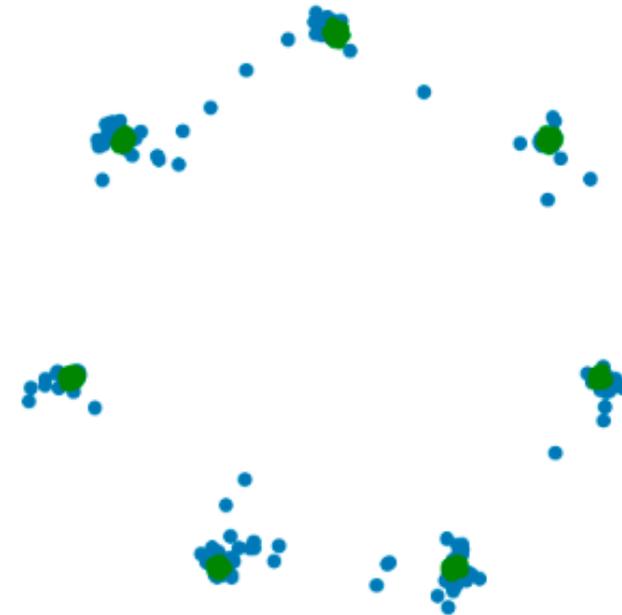
- Real Sample
- Fake Sample

What's wrong with Generative models?

GAN



GDPP-GAN



- Real Sample
- Fake Sample

Determinantal Point Process (DPP)

$$\mathcal{P}(S \subseteq Y) \propto \det(L_S)$$

$$\mathcal{P}(S \subseteq Y) \propto \det(\phi(S)^\top \phi(S))$$

ϕ is feature representation of subset S sampled from ground set Y

Determinantal Point Process (DPP)

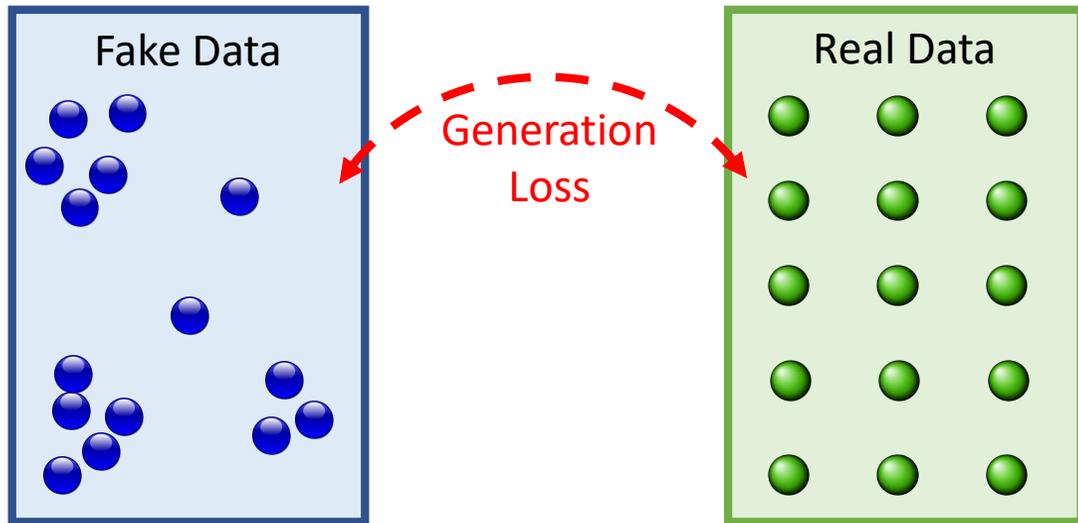
$$\mathcal{P}(S \subseteq Y) \propto \det(L_S)$$

$$\mathcal{P}(S \subseteq Y) \propto \det(\phi(S)^\top \phi(S))$$

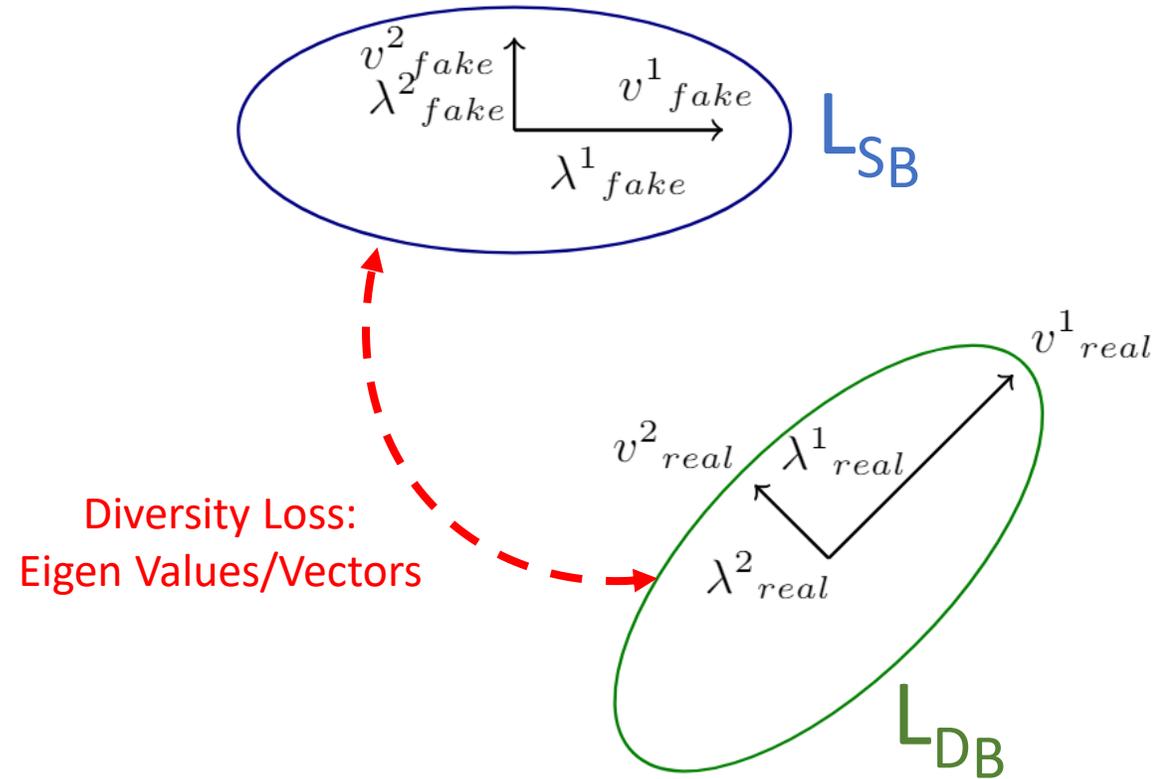
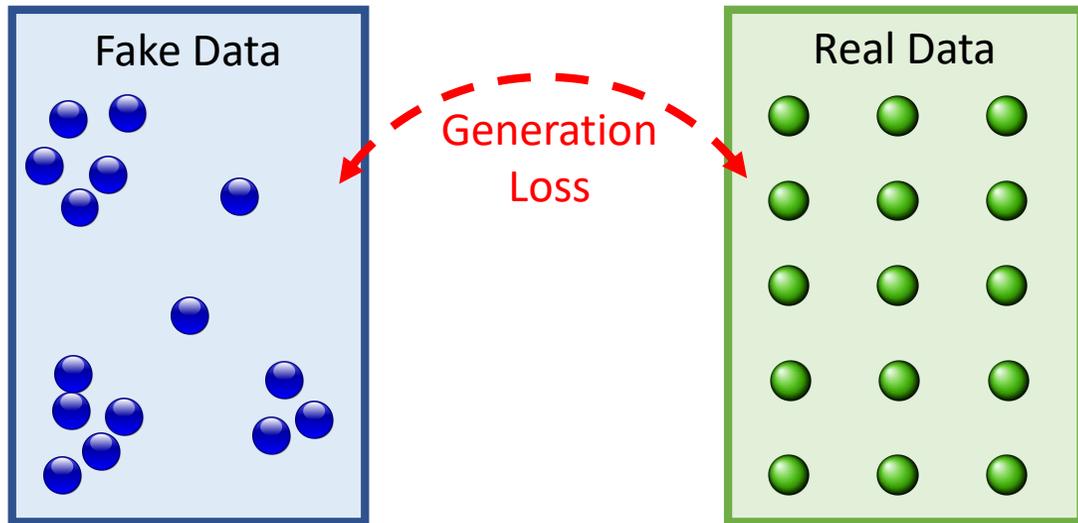
ϕ is feature representation of subset S sampled from ground set Y

L_S : DPP kernel, models the diversity of a mini-batch S

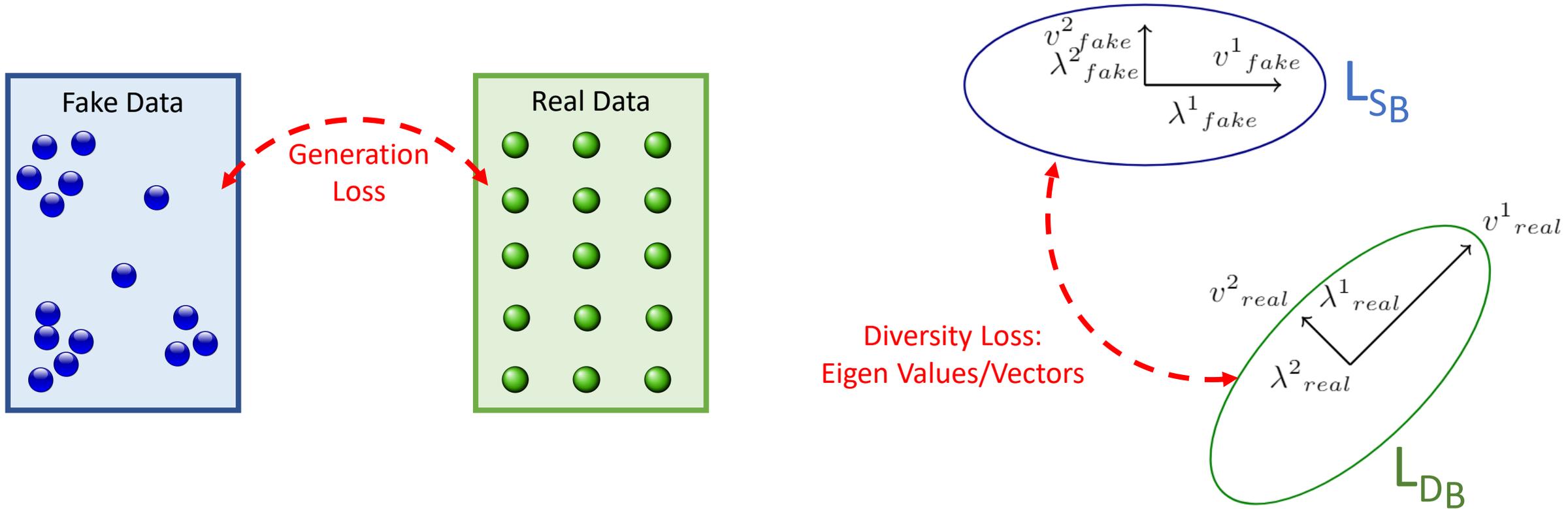
What is GDPP?



What GDPP?

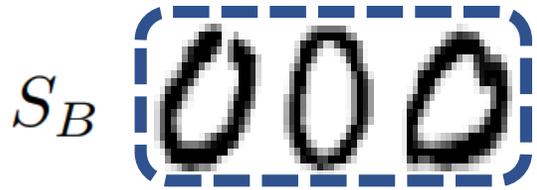


What GDPP?



$$\mathcal{L}_g^{DPP} = \mathcal{L}_m + \mathcal{L}_s = \sum_i \|\lambda_{real}^i - \lambda_{fake}^i\|_2 - \sum_i \hat{\lambda}_{real}^i \cos(v_{real}^i, v_{fake}^i)$$

How GDPP?

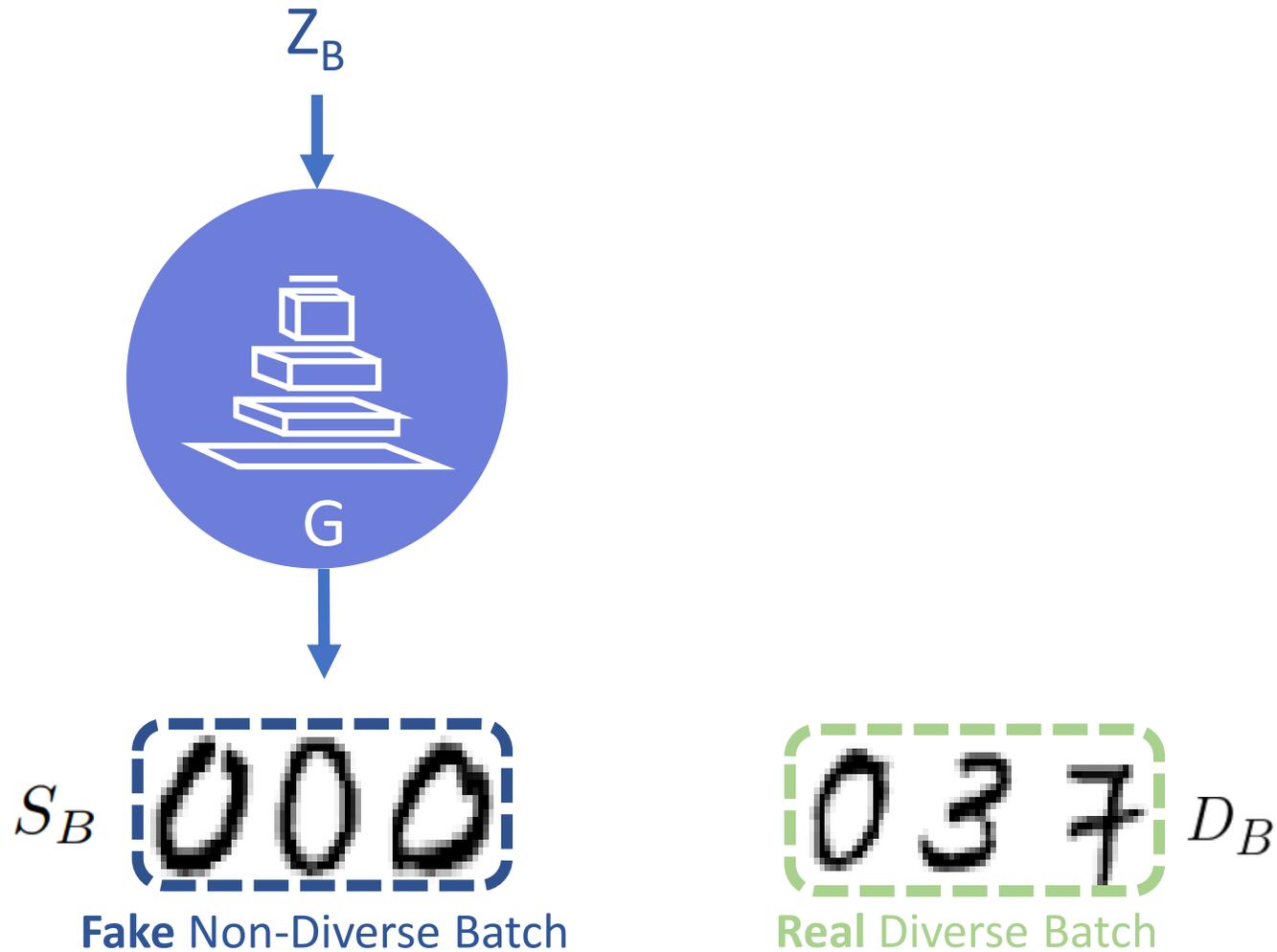


Fake Non-Diverse Batch

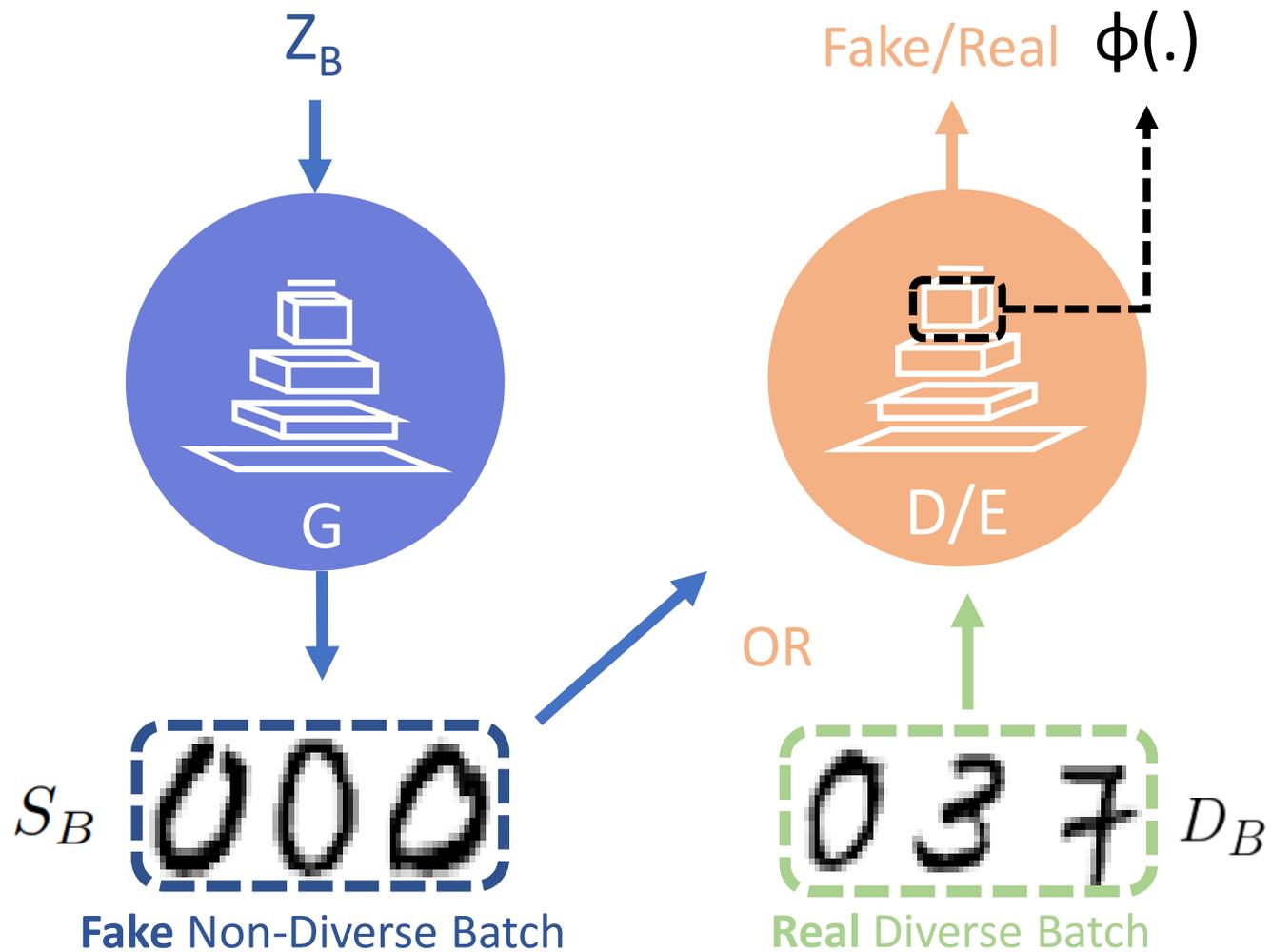


Real Diverse Batch

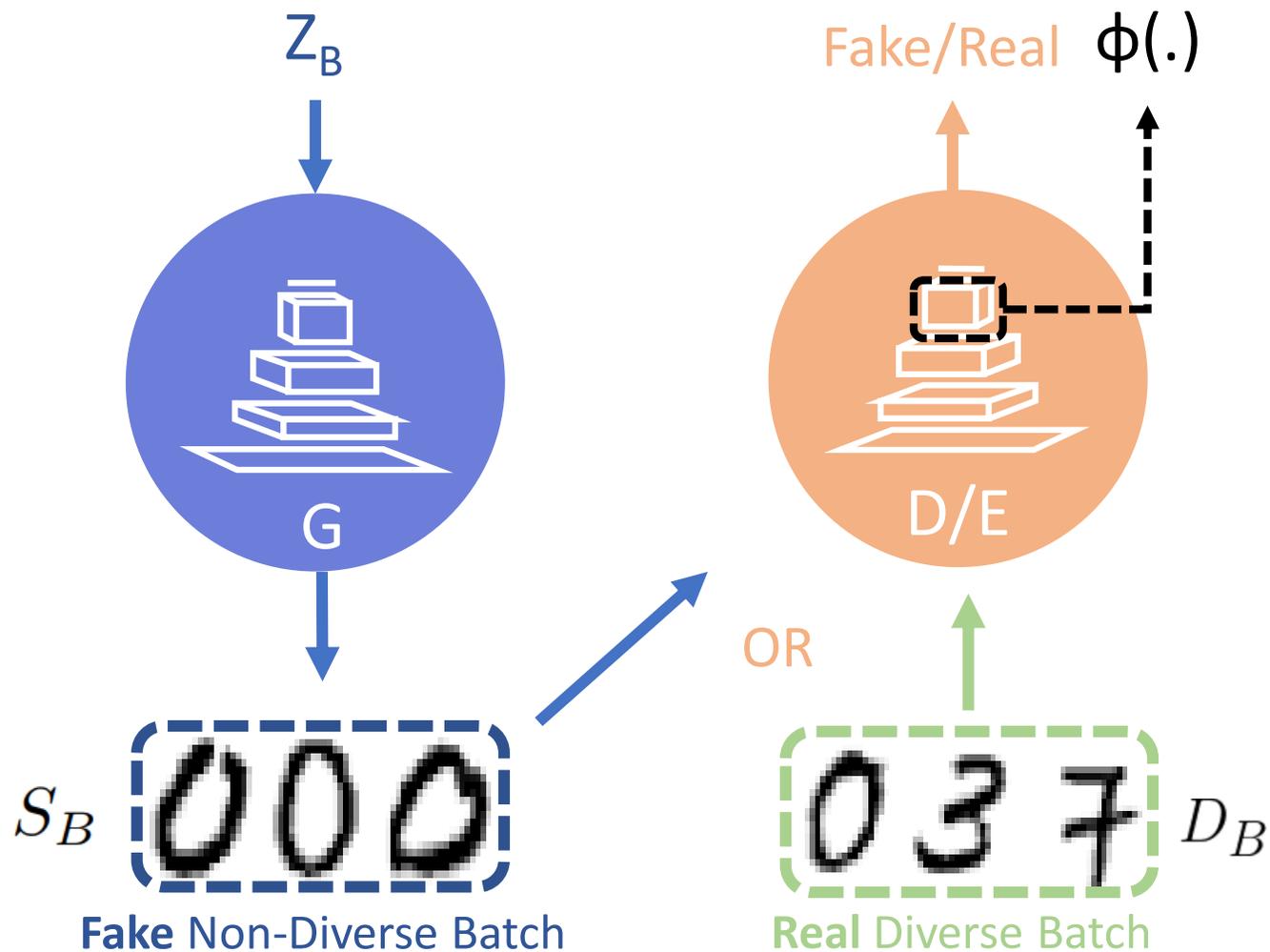
How GDPP?



How GDDPP?



How GDPP?



Diversity Loss

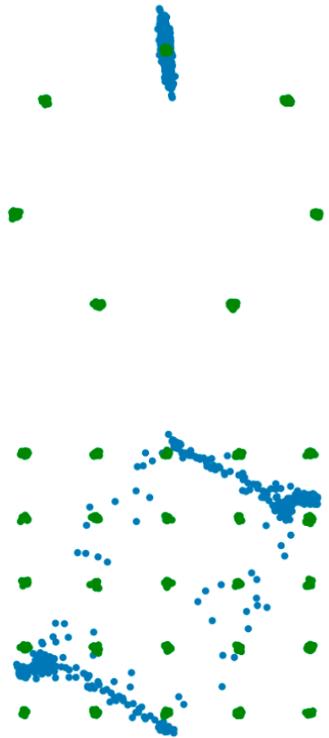
$$S_B = G(z_B) \quad D_B \sim \text{Real}$$

$$L_B = \phi(B)^T \phi(B)$$

Does it work? (Synthetic)

- Real Sample
- Fake Sample

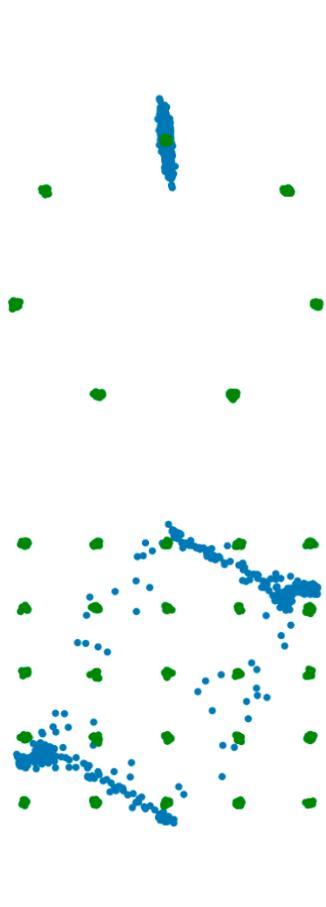
GAN



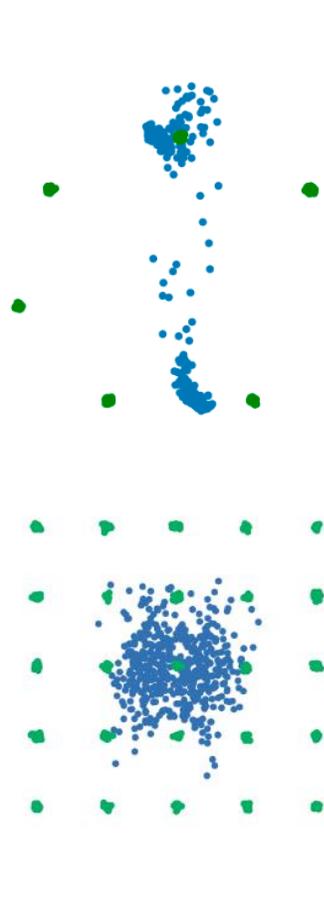
Does it work? (Synthetic)

- Real Sample
- Fake Sample

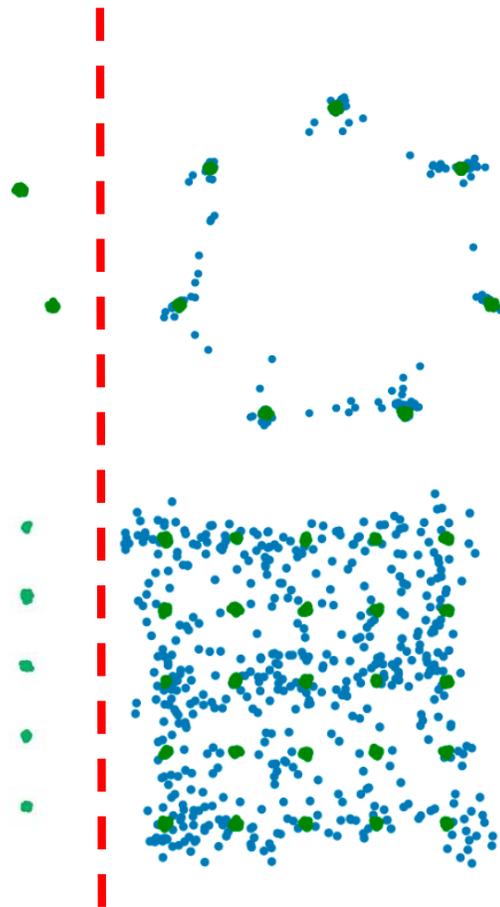
GAN



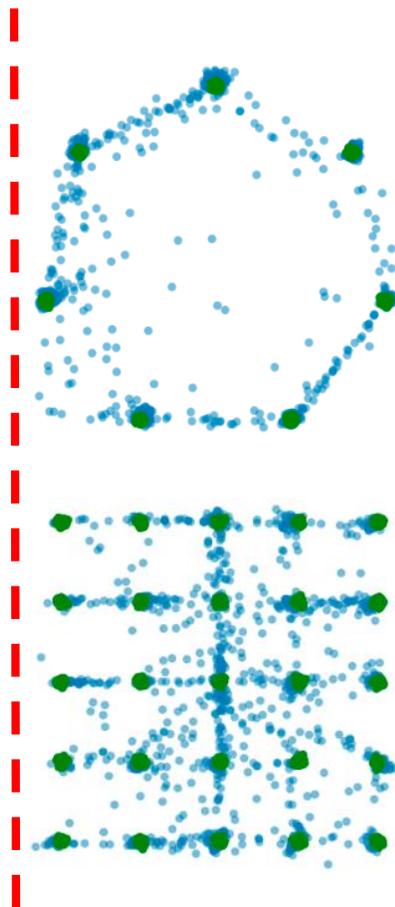
ALI



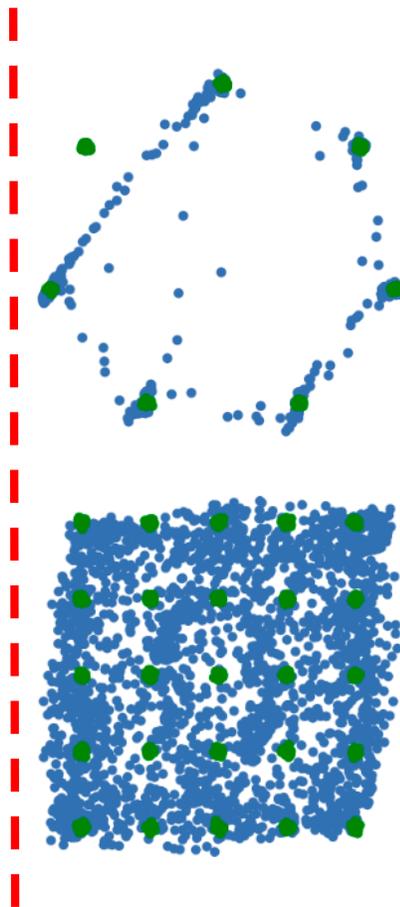
Unrolled-GAN



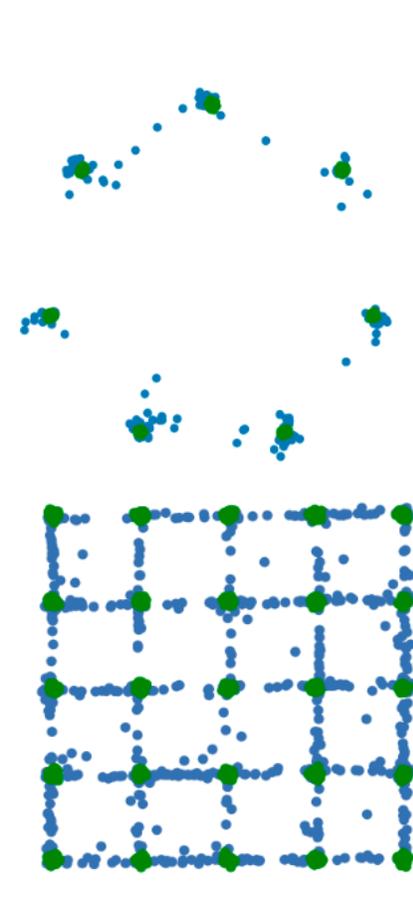
VEE-GAN



WP-GAN

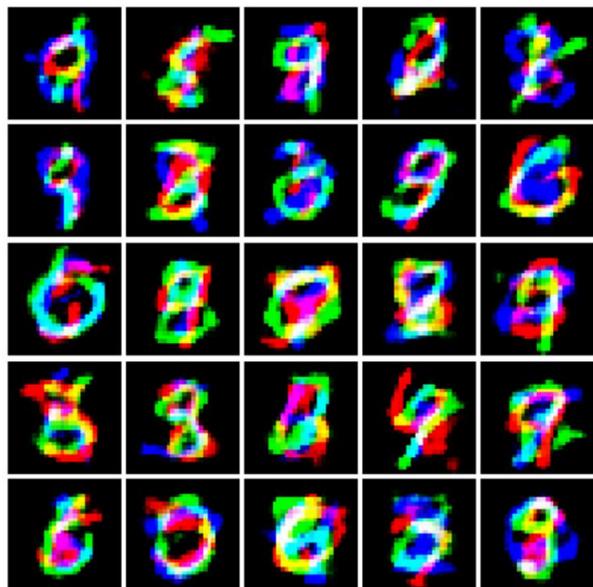


GDPP-GAN

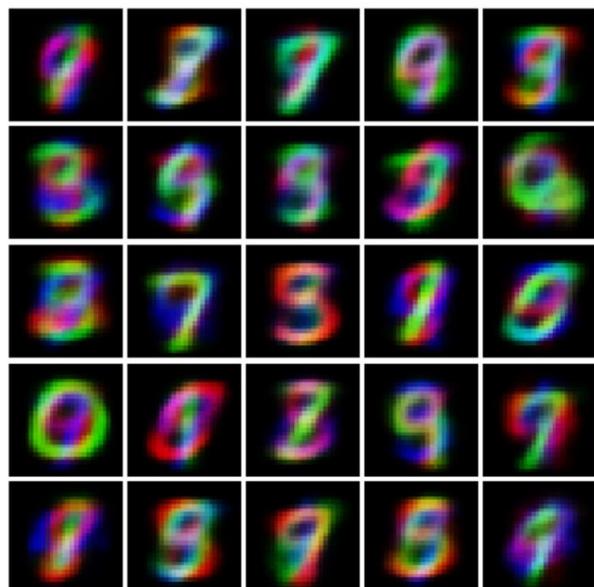


Does it work? (Real)

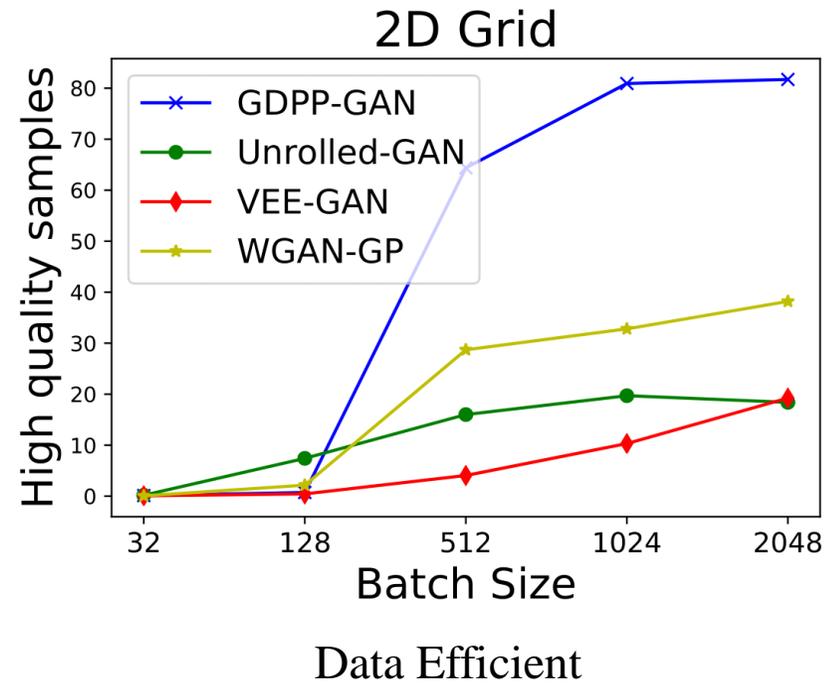
GDPP-GAN



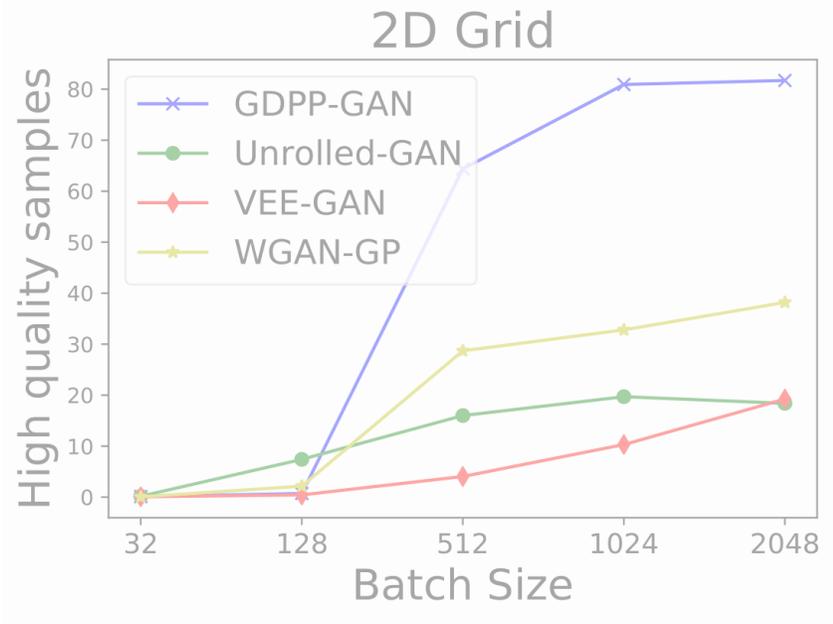
GDPP-VAE



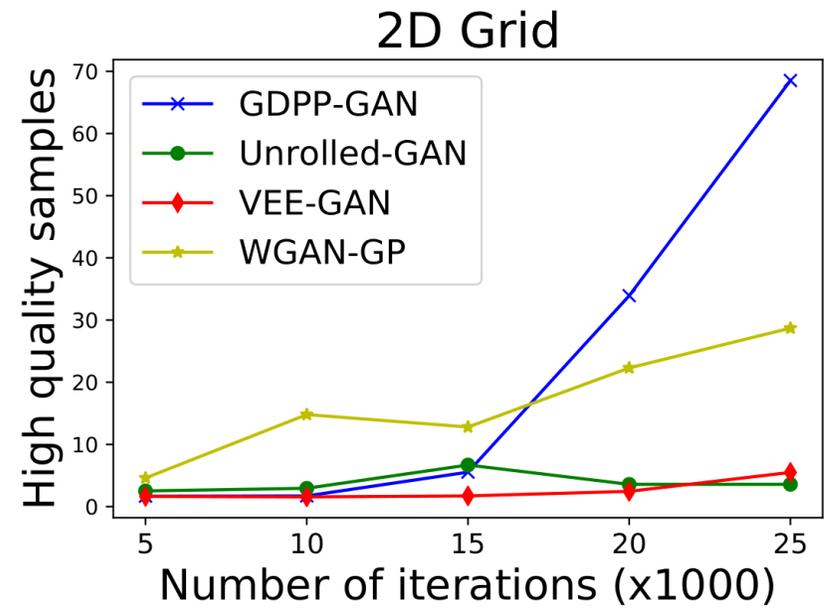
What else?



What else?

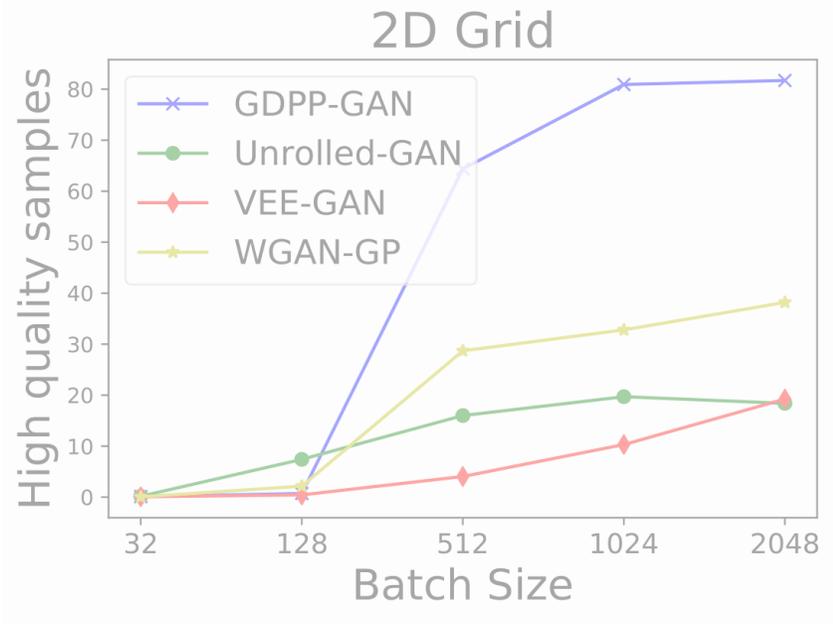


Data Efficient

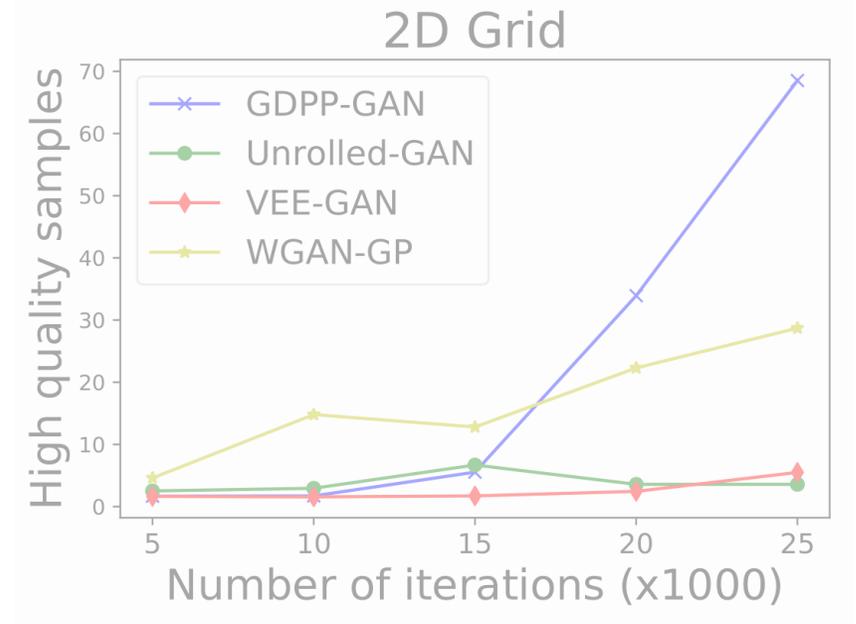


Time Efficient

What else?



Data Efficient

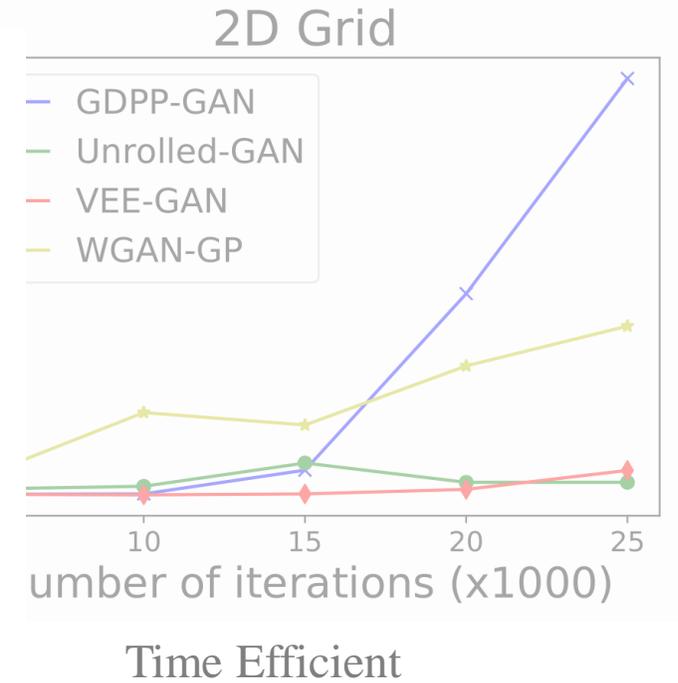
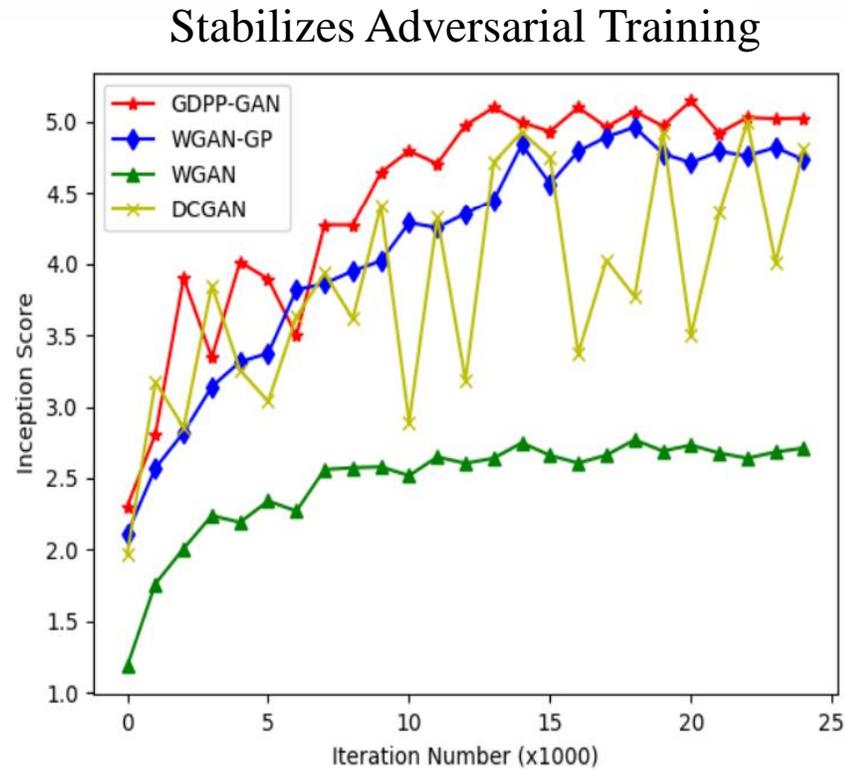
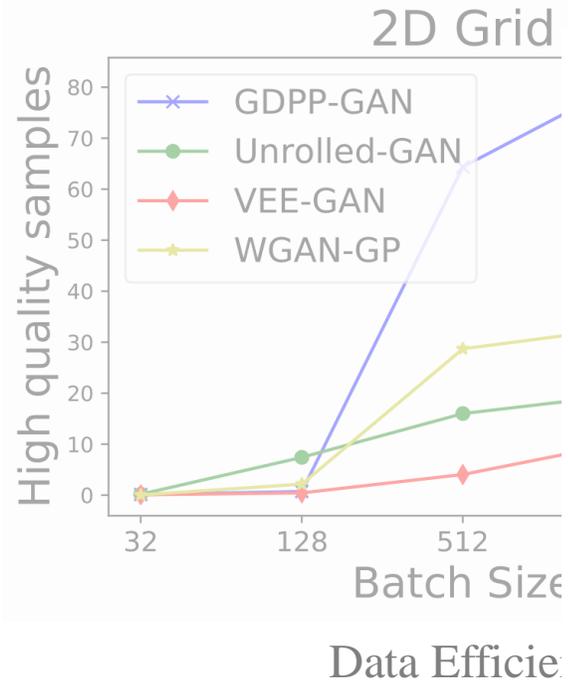


Time Efficient

	DCGAN	Unrolled-GAN	VEE-GAN	Reg-GAN	WGAN	WGAN-GP	GDPP-GAN
Avg. Iter. Time (s)	0.0674	0.2467	0.1978	0.1357	0.1747	0.4331	0.0746

Fast Training Time

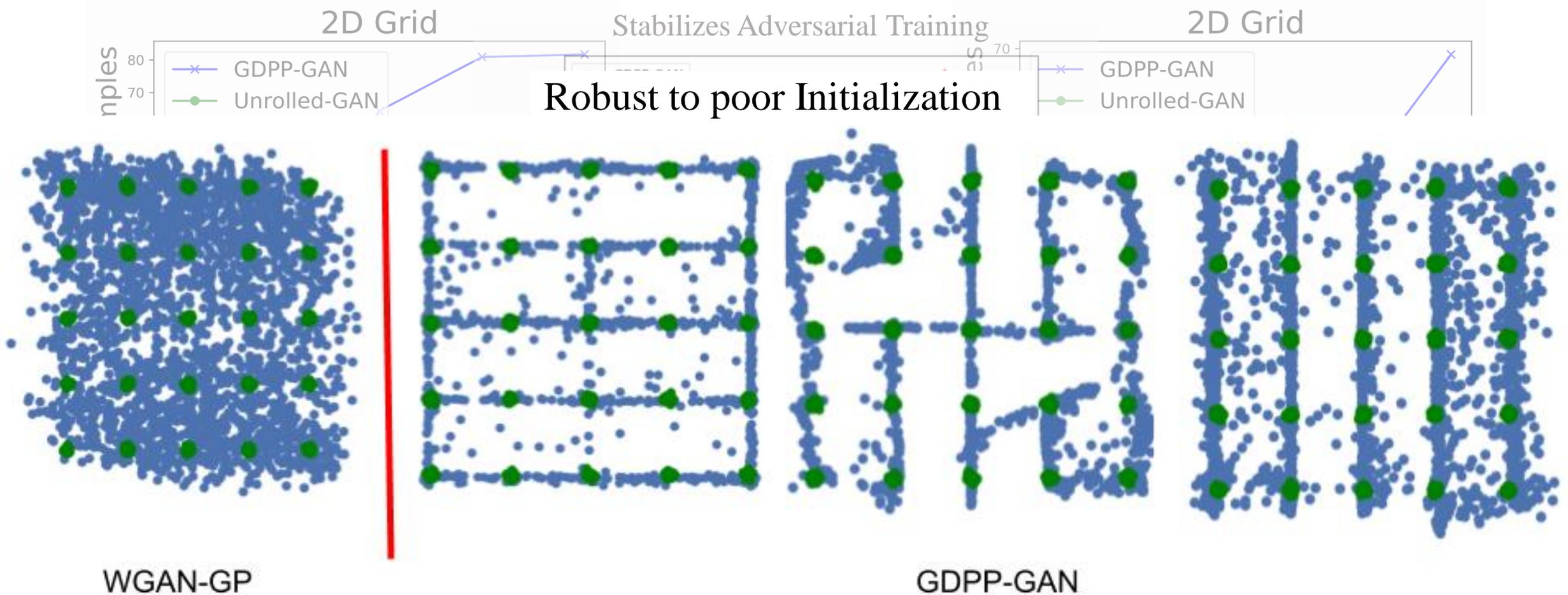
What else?



	DCGAN	Unrolled-GAN	VEE-GAN	Reg-GAN	WGAN	WGAN-GP	GDPP-GAN
Avg. Iter. Time (s)	0.0674	0.2467	0.1978	0.1357	0.1747	0.4331	0.0746

Fast Training Time

What else?



Fast Training Time

Why GDPP?

1. No extra trainable parameters (cost-free)

Why GDPP?

1. No extra trainable parameters (cost-free)
2. Unsupervised Setting (No labels)

Why GDPP?

1. No extra trainable parameters (cost-free)
2. Unsupervised Setting (No labels)
3. **Stabilizes Adversarial Training**

Why GDPP?

1. No extra trainable parameters (cost-free)
2. Unsupervised Setting (No labels)
3. Stabilizes Adversarial Training
4. Time and Data efficient

Why GDPP?

1. No extra trainable parameters (cost-free)
2. Unsupervised Setting (No labels)
3. Stabilizes Adversarial Training
4. Time and Data efficient
5. Architecture & Model Invariant (GAN & VAE)

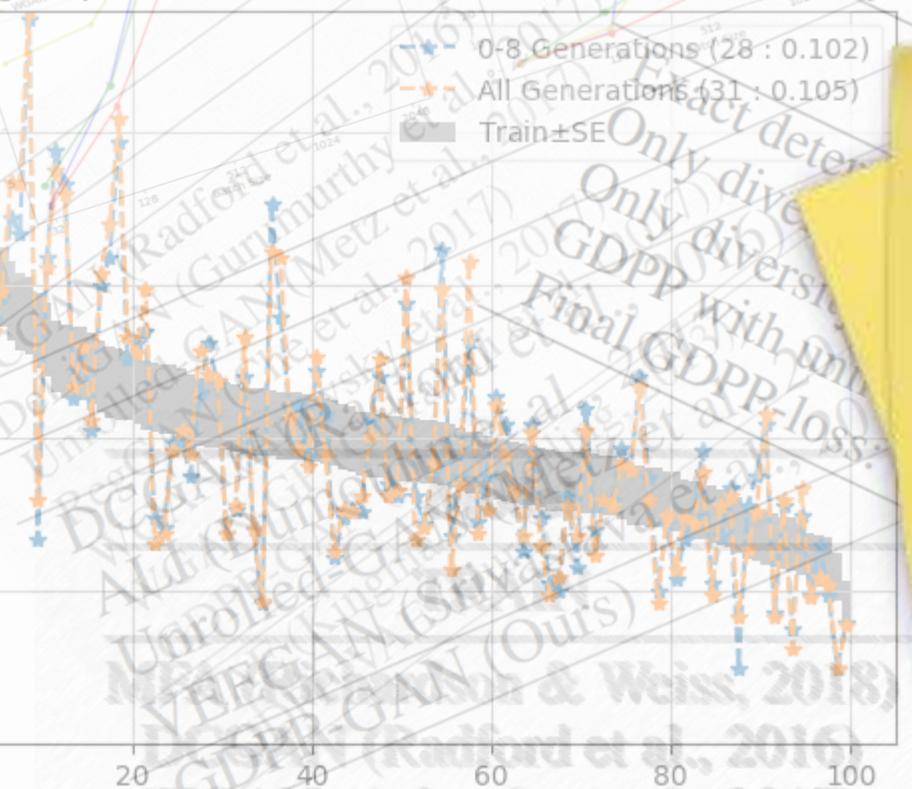
Why GDPP?

1. No extra trainable parameters (cost-free)
2. Unsupervised Setting (No labels)
3. Stabilizes Adversarial Training
4. Time and Data efficient
5. Architecture & Model Invariant (GAN & VAE)

Yet, Consistently outperforms state-of-the-art

For many more, join us in

Proportions Evaluation Results for 100 bins (NDB : JS)



143

Proportions Evaluation Results for 200 bins (NDB : JS)

