Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

Paper ID: TUE-AM-370

Siddarth Asokan[1] and Chandra Sekhar Seelamantula[2]
[1]Robert Bosch Centre for Cyber Physical Systems
[2]Department of Electrical Engineering
Indian Institute of Science (IISc), Bengaluru-560012, India

Corresponding Email: [1]siddartha@iisc.ac.in

CVPR 2023

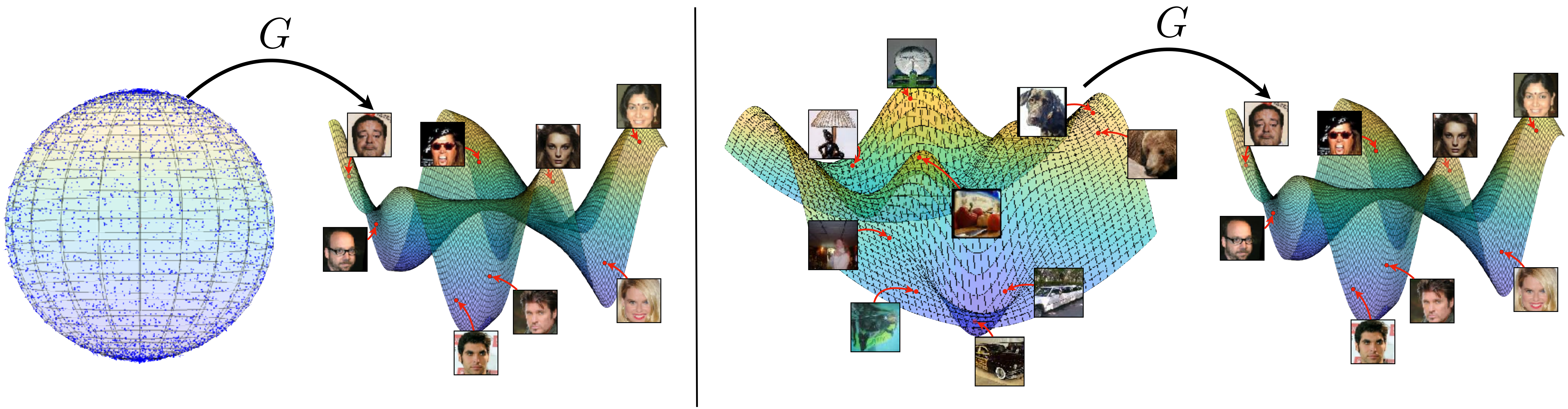
Robert Bosch Centre for CYBER PHYSICAL SYSTEMS

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

SPECTRUM LAB

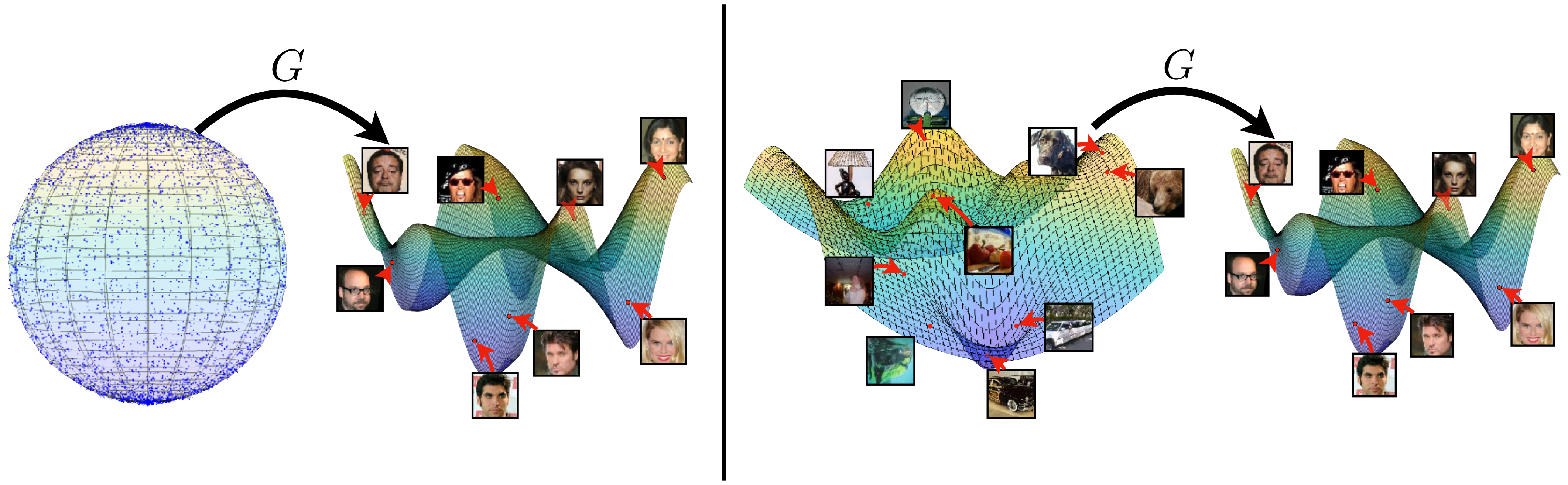
Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

- A alternative to improving the generator – Choose a "better" input distribution.
- Replace Gaussian noise with samples from a *closely related* dataset.



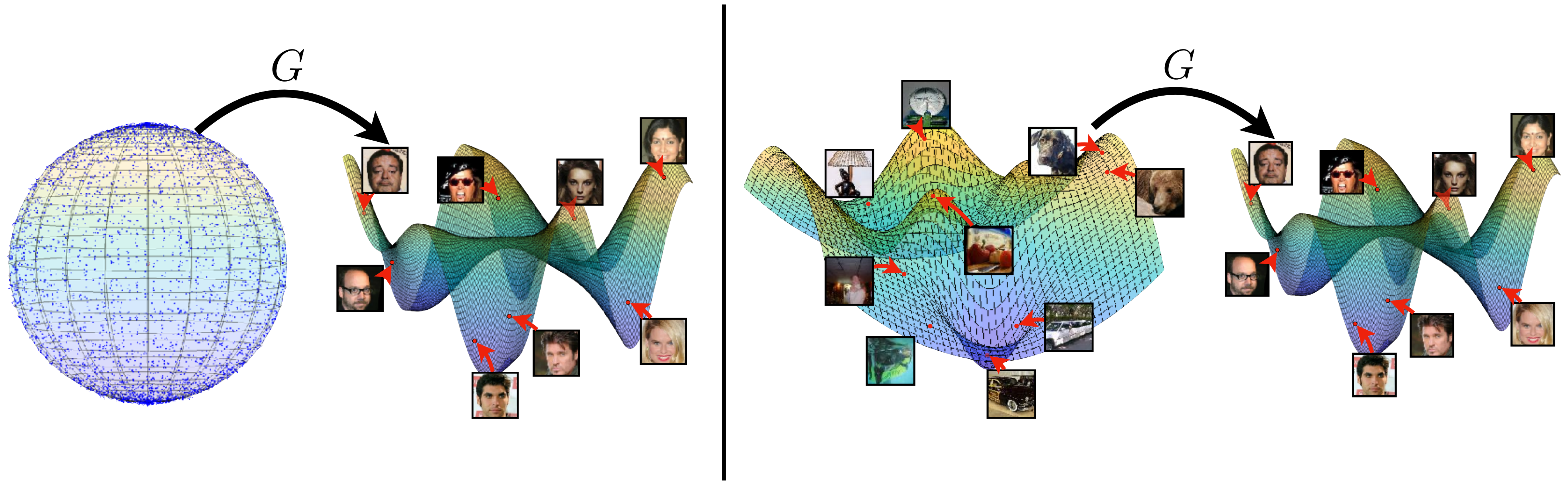
Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

- Choose a “better” input distribution.

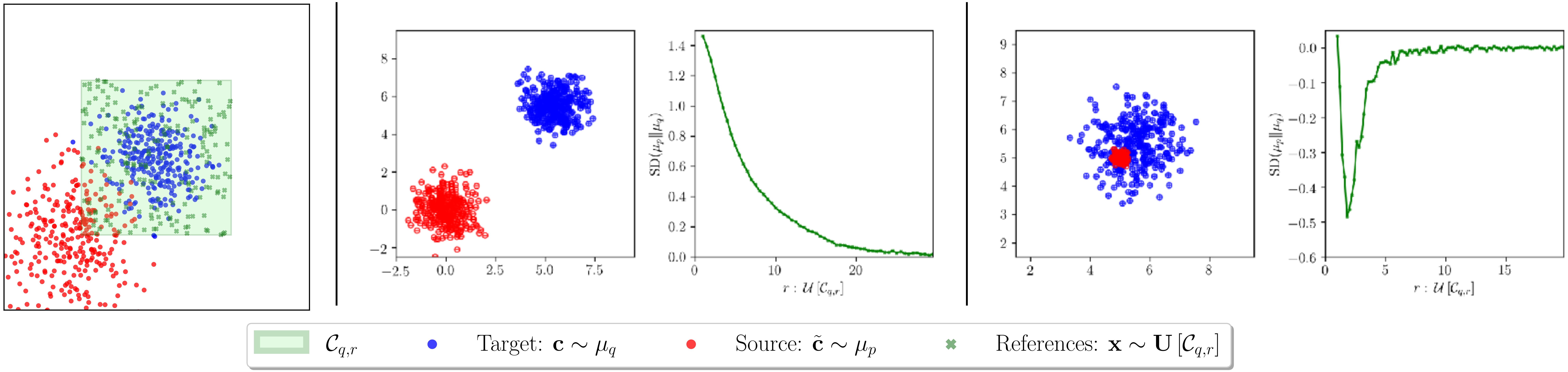


Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

- Choose a "better" input distribution.

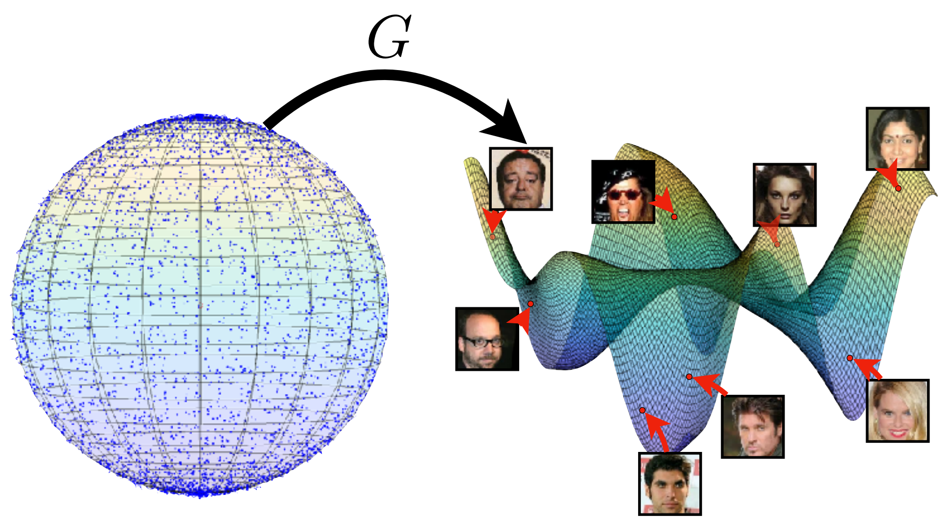


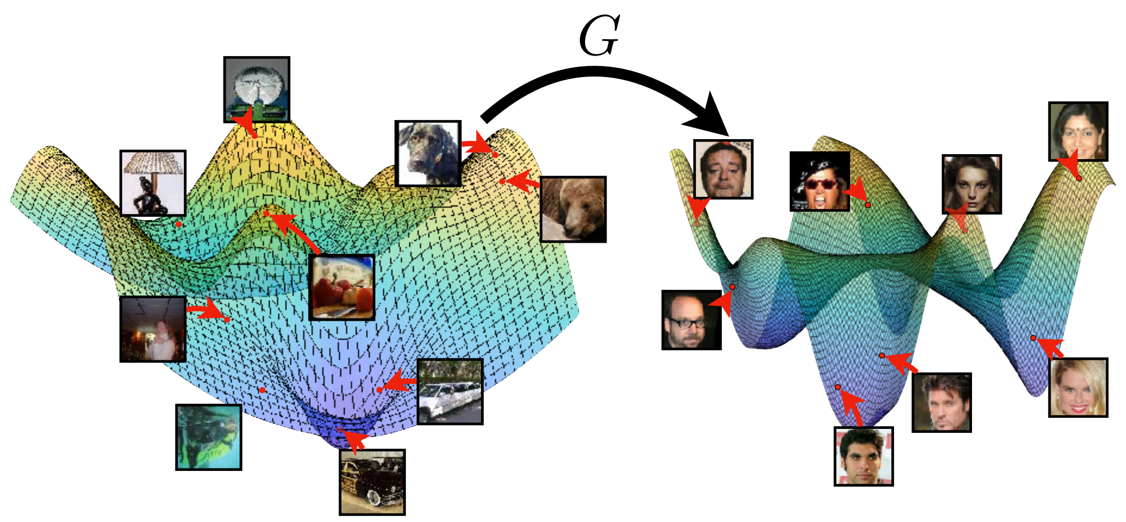
- **The signed Inception distance** $\mathrm{SID}(p_s \| p_t)$ of a source p_s w.r.t. a reference *target* p_t — A kernel function evaluated over samples drawn from a cube $\mathcal{C}_r$ of side r, around target p_t.



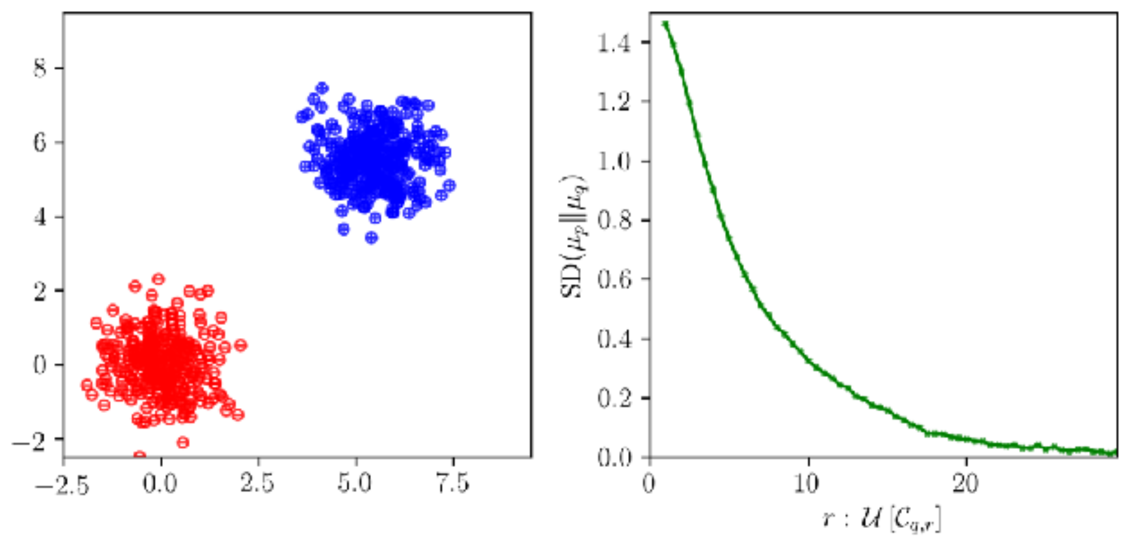
Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

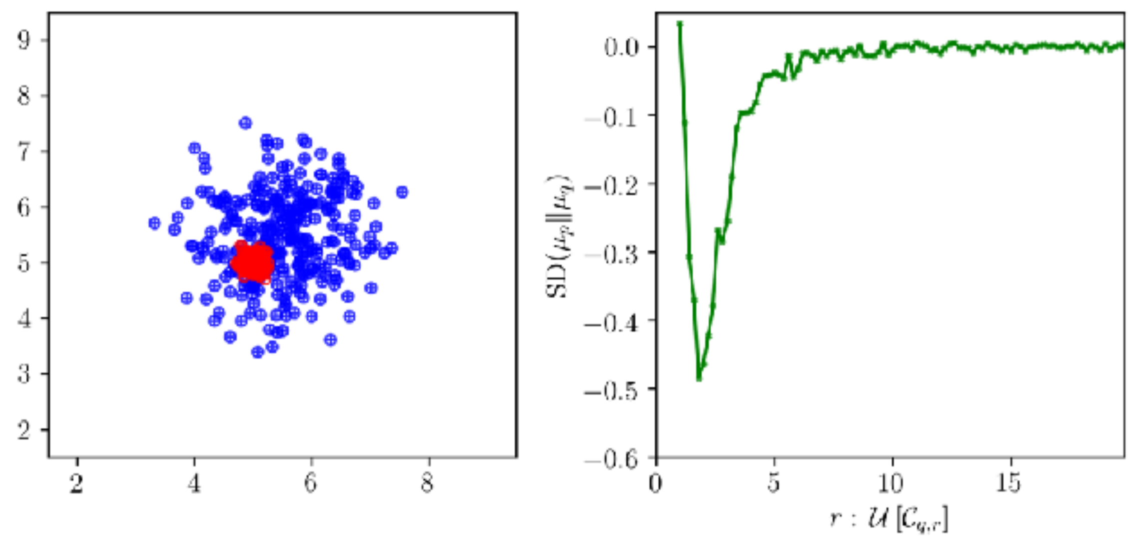
- Choose a “better” input distribution.





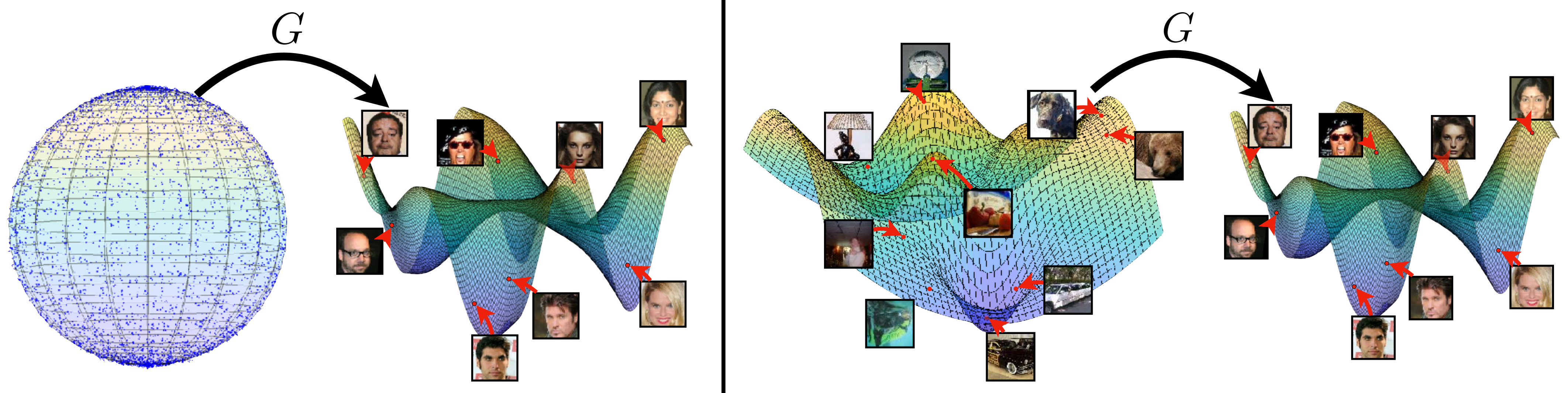
- Employ **SID** to identify *friendly neighbors*.



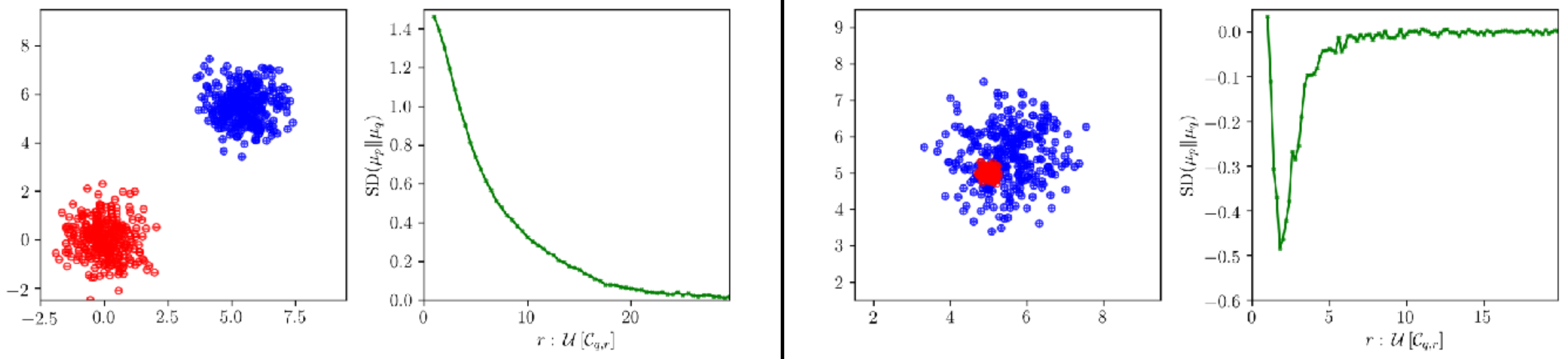


Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

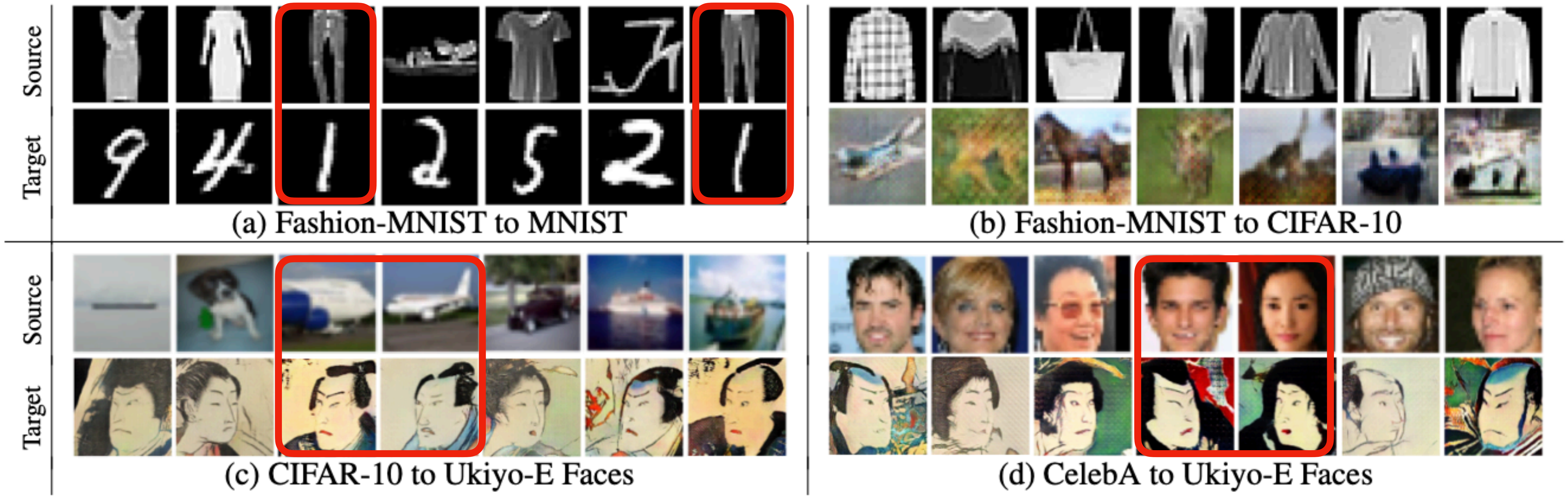
- Choose a "better" input distribution.



- Employ **SID** to identify *friendly neighbors*.

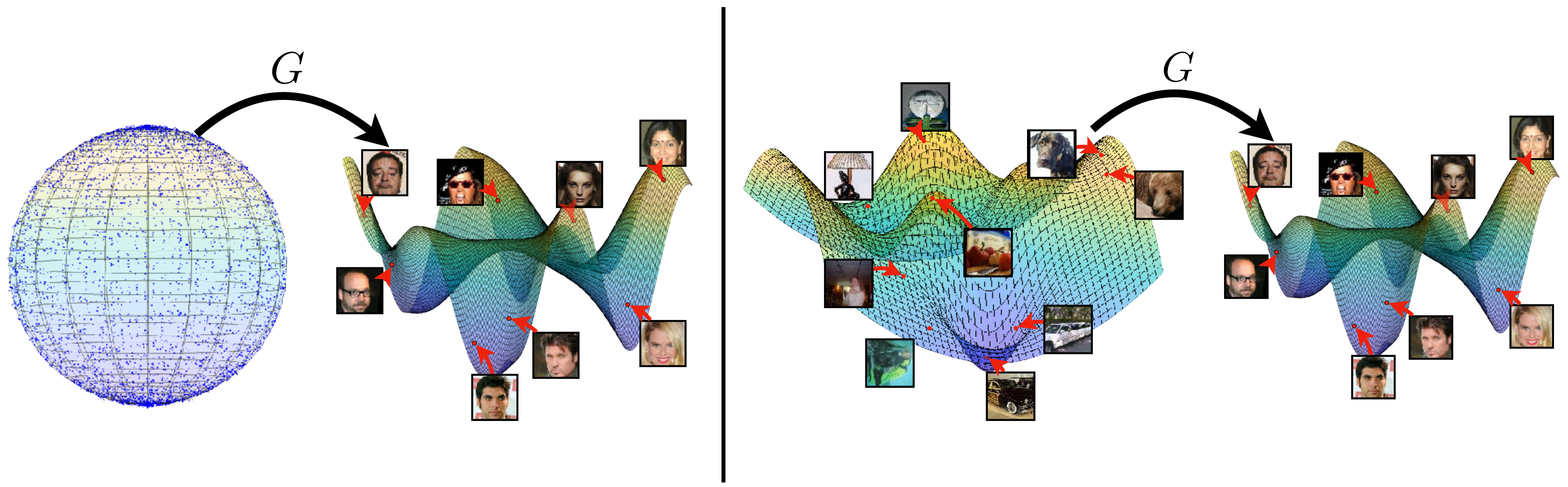


- Visual similarity in not necessary; Underling structural similarity is implicitly leveraged.

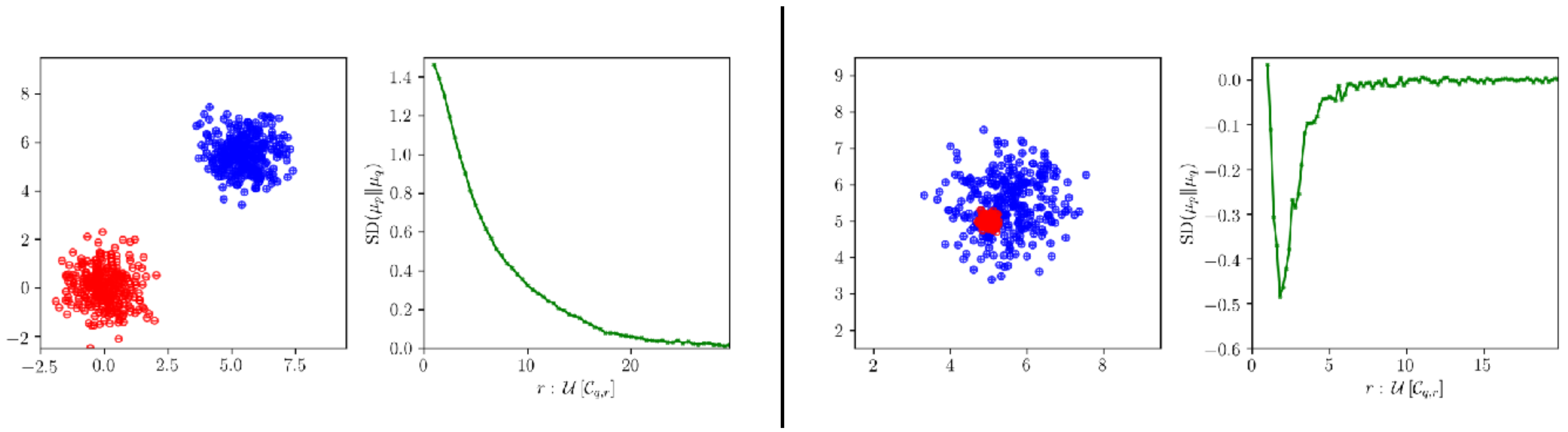


Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

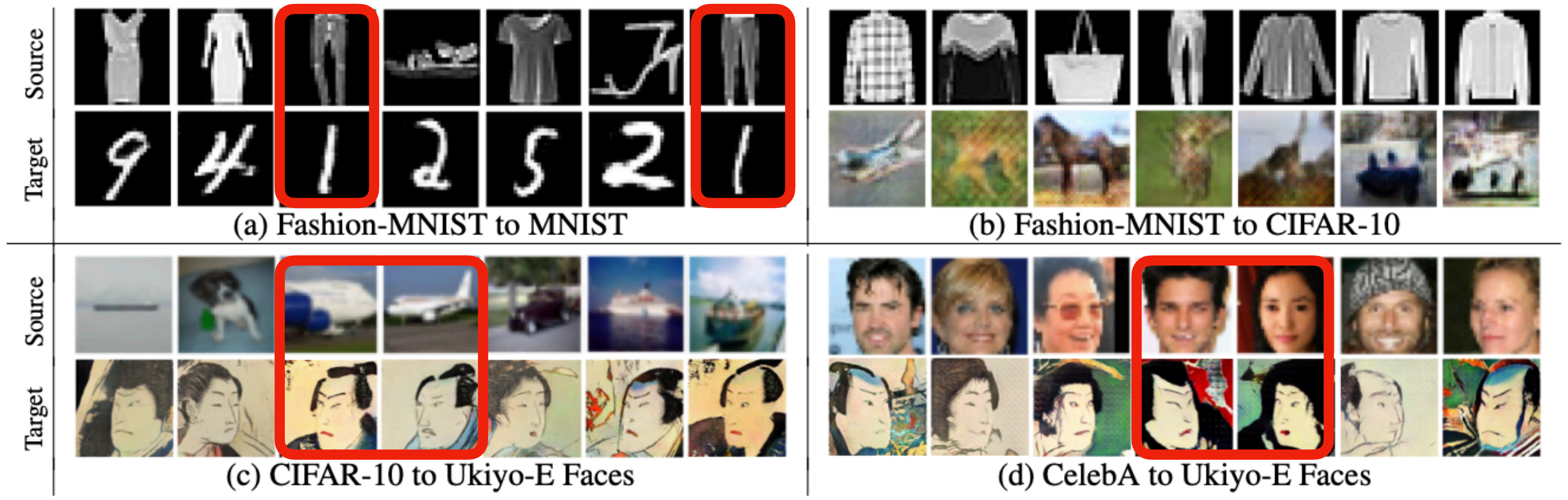
- Choose a “better” input distribution.



- Employ **SID** to identify *friendly neighbors*.



- Structural similarity is implicitly leveraged.



(a) Fashion-MNIST to MNIST

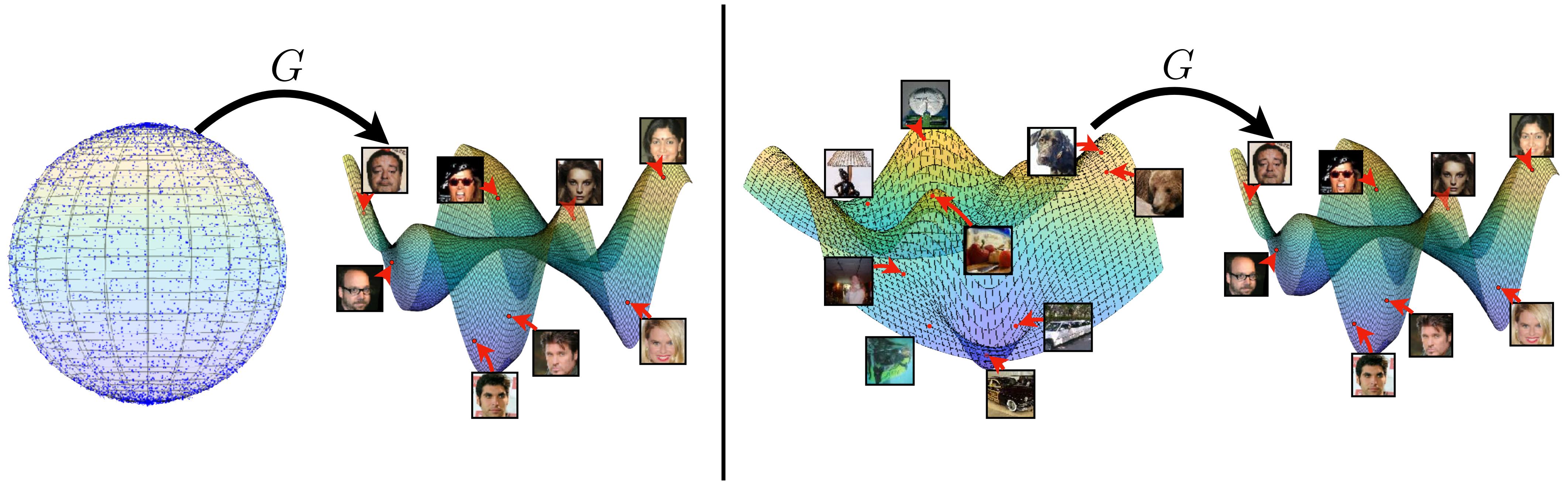
(b) Fashion-MNIST to CIFAR-10

(c) CIFAR-10 to Ukiyo-E Faces

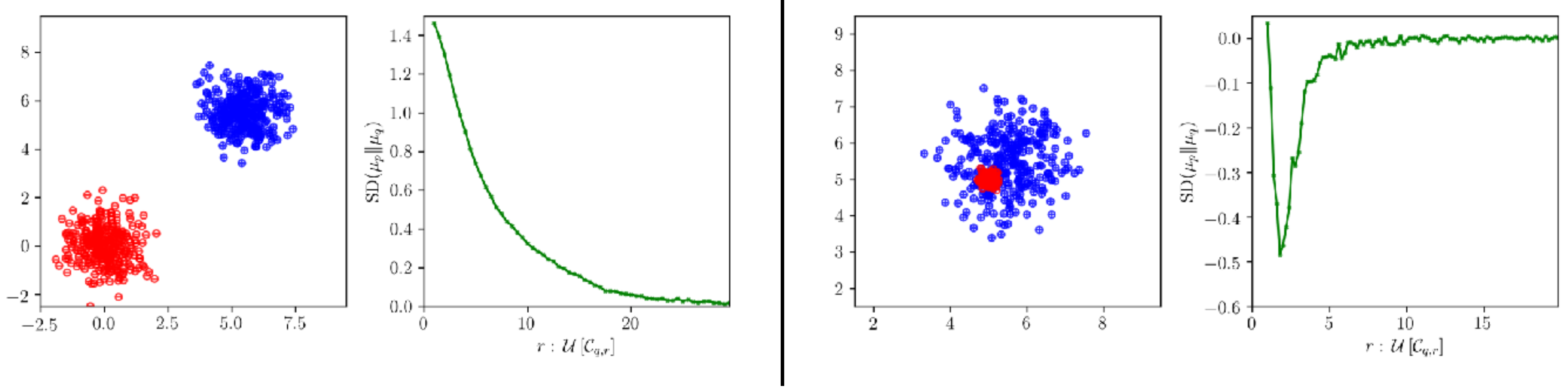
(d) CelebA to Ukiyo-E Faces

Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

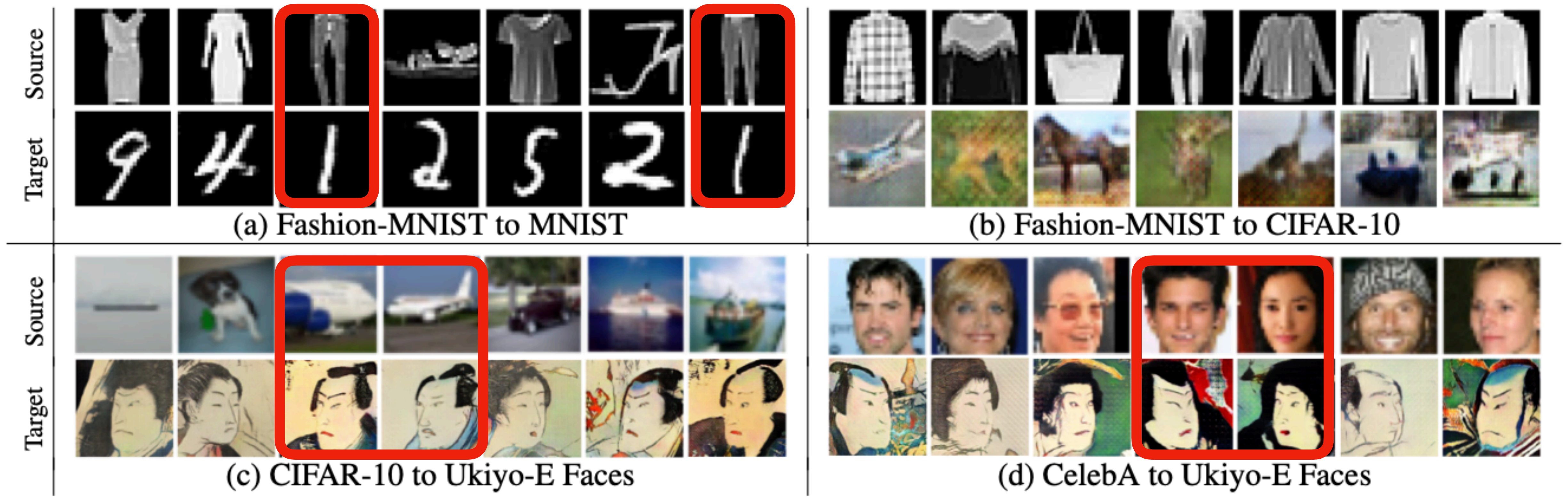
- Choose a “better” input distribution.



- Employ **SID** to identify *friendly neighbors*.



- Structural similarity is implicitly leveraged.



- State-of-the-art FID in a fifth of the training time!

Spider GANs: Leveraging Friendly Neighbors to Accelerate GAN Training

Paper ID: TUE-AM-370

Siddarth Asokan[1] and Chandra Sekhar Seelamantula[2]
[1]Robert Bosch Centre for Cyber Physical Systems
[2]Department of Electrical Engineering
Indian Institute of Science (IISc), Bengaluru-560012, India

Corresponding Email: [1]siddartha@iisc.ac.in

CVPR 2023

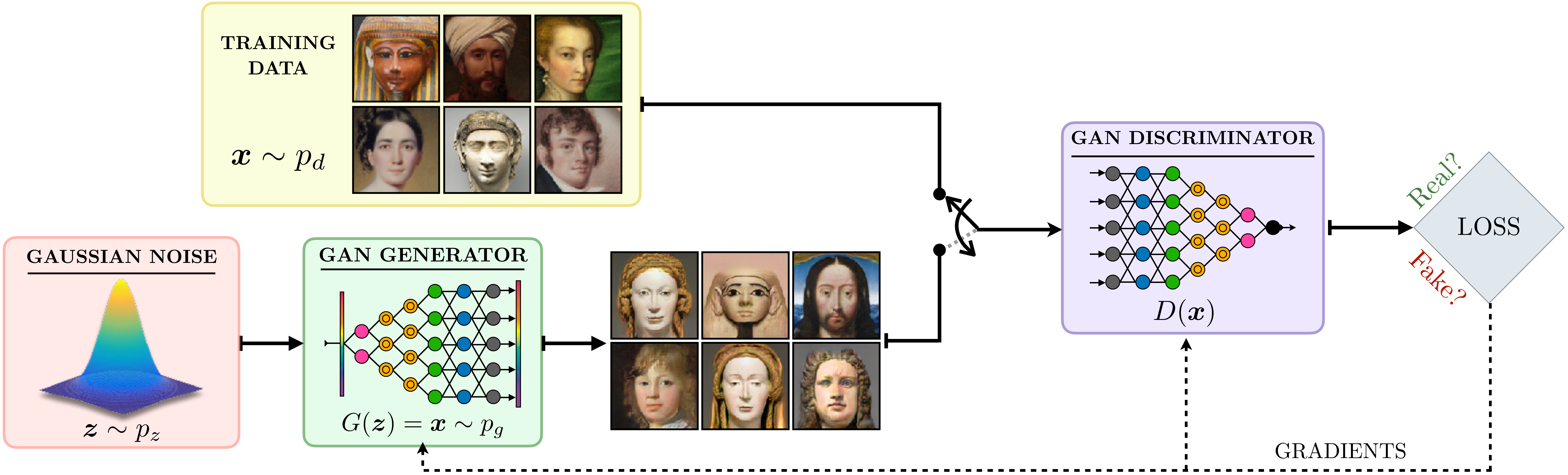
Robert Bosch Centre for CYBER PHYSICAL SYSTEMS

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

SPECTRUM LAB

Generative Adversarial Networks (GANs)

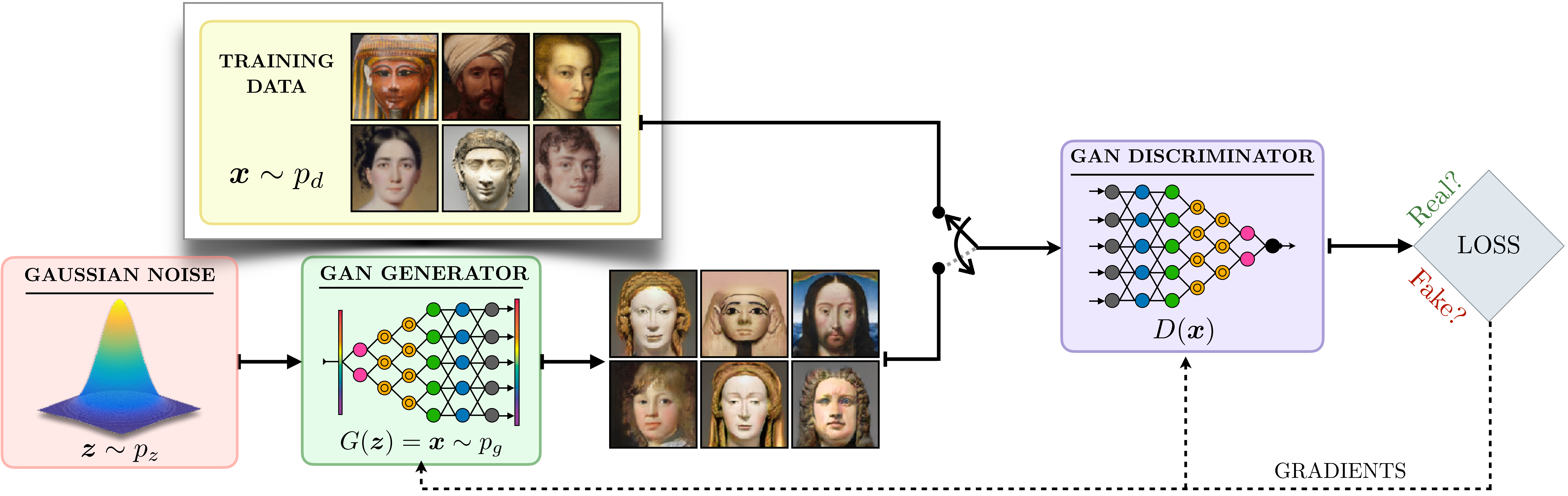
- Objective: Learn a desired data distribution.



[1] Goodfellow et al., "Generative Adversarial Networks," NeurIPS 2014

Generative Adversarial Networks (GANs)

- Objective: Learn a desired data distribution.

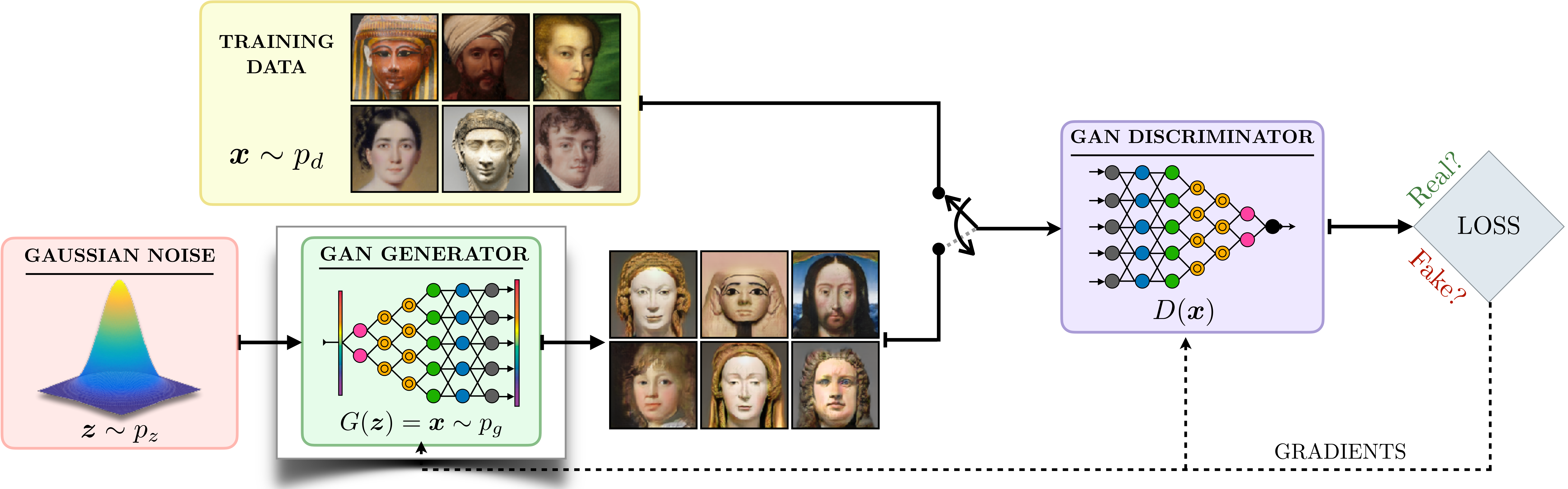


- A min-max game between two players:

[1] Goodfellow et al., "Generative Adversarial Networks," NeurIPS 2014

Generative Adversarial Networks (GANs)

- Objective: Learn a desired data distribution.

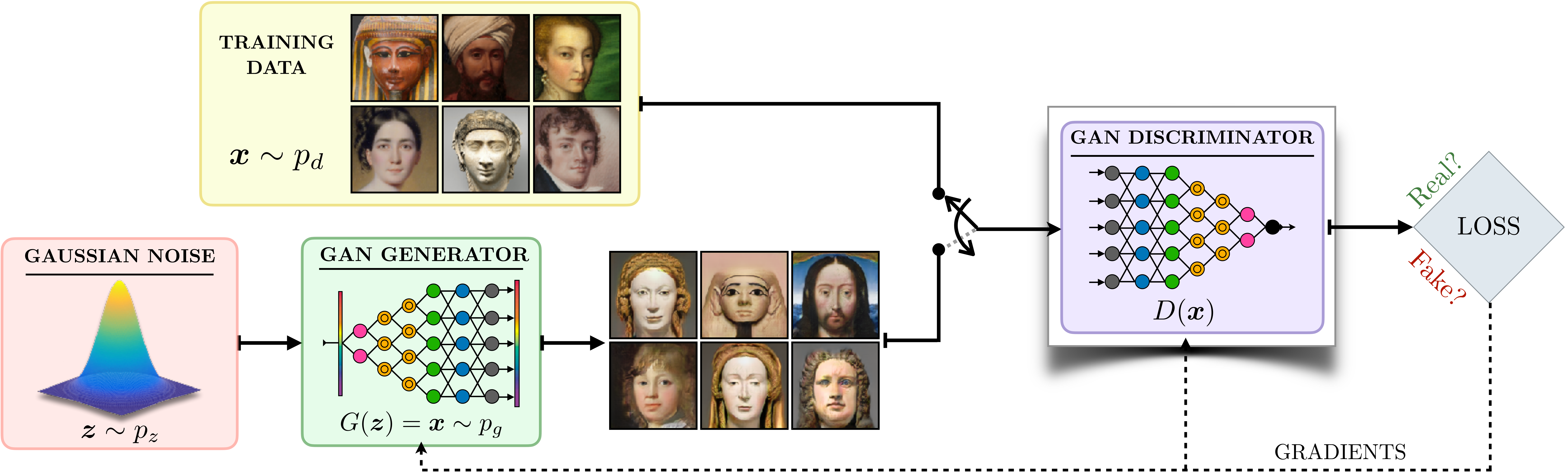


- A min-max game between two players:
  - *The generator* $G(\boldsymbol{z})$: Learns to create samples from the distribution of the real data.

[1] Goodfellow et al., "Generative Adversarial Networks," NeurIPS 2014

Generative Adversarial Networks (GANs)

- Objective: Learn a desired data distribution.

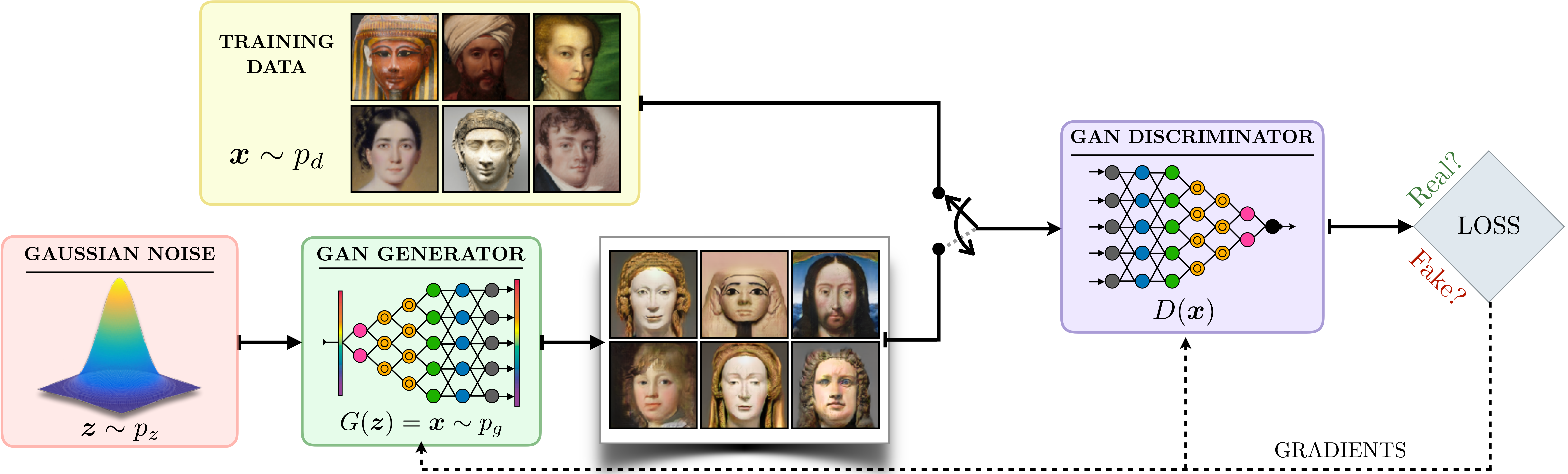


- A min-max game between two players:
  - *The generator* $G(z)$: Learns to create samples from the distribution of the real data.
  - *The discriminator* $D(x)$: Differentiates between the real and fake samples.

[1] Goodfellow et al., "Generative Adversarial Networks," NeurIPS 2014

Generative Adversarial Networks (GANs)

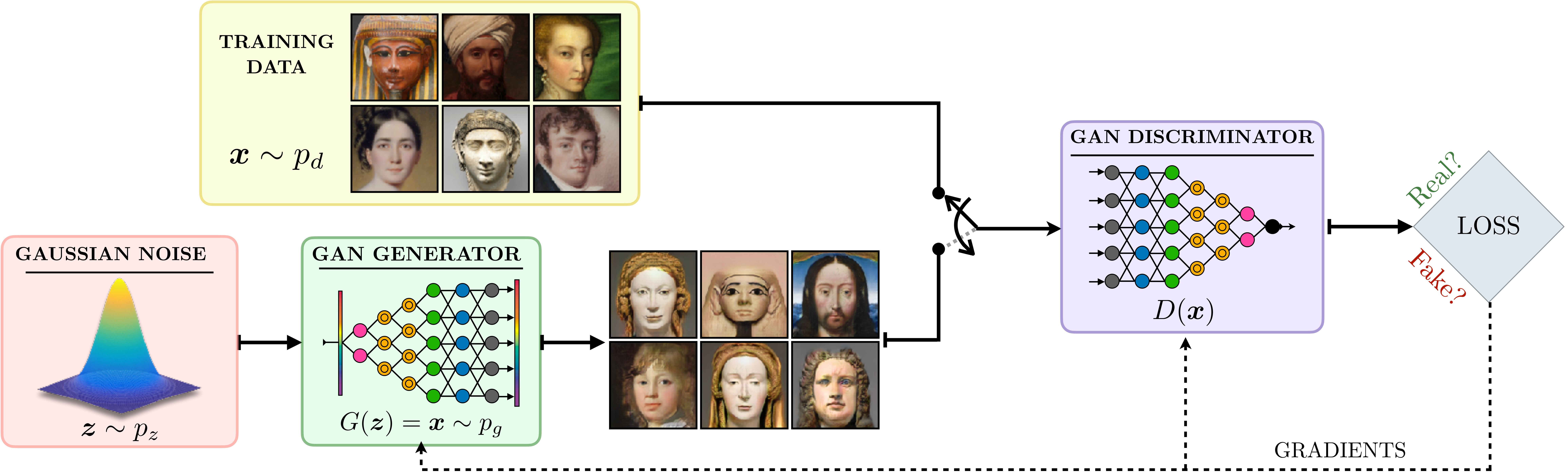
- Objective: Learn a desired data distribution.



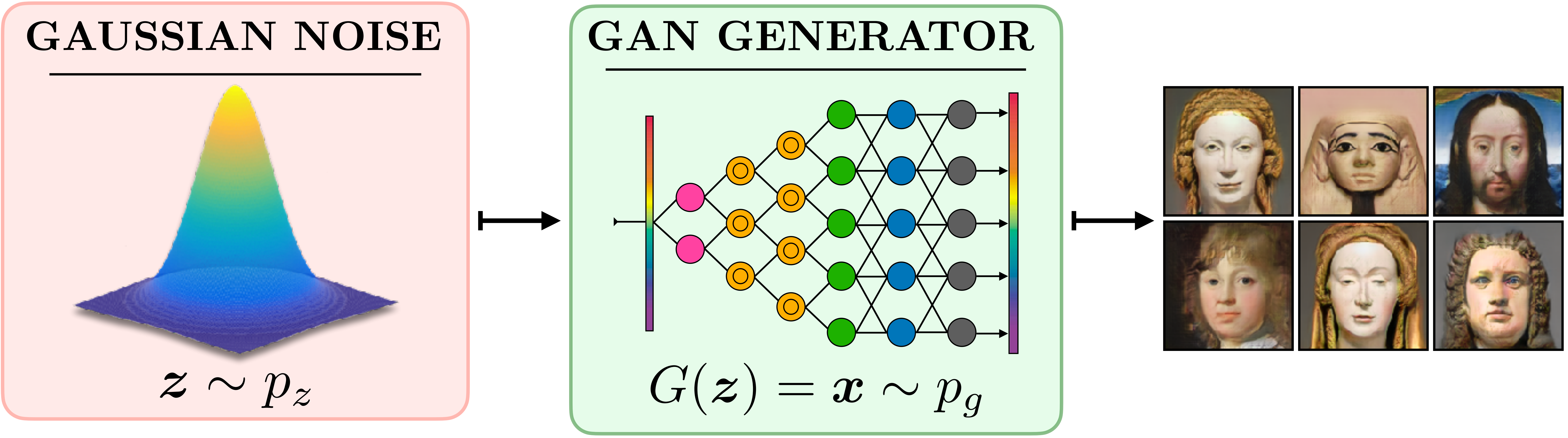
- A min-max game between two players:
  - *The generator* $G(\boldsymbol{z})$: Learns to create samples from the distribution of the real data.
  - *The discriminator* $D(\boldsymbol{x})$: Differentiates between the real and fake samples.
  - *The objective:* Learn the optimal $G(\boldsymbol{z})$

[1] Goodfellow et al., "Generative Adversarial Networks," NeurIPS 2014

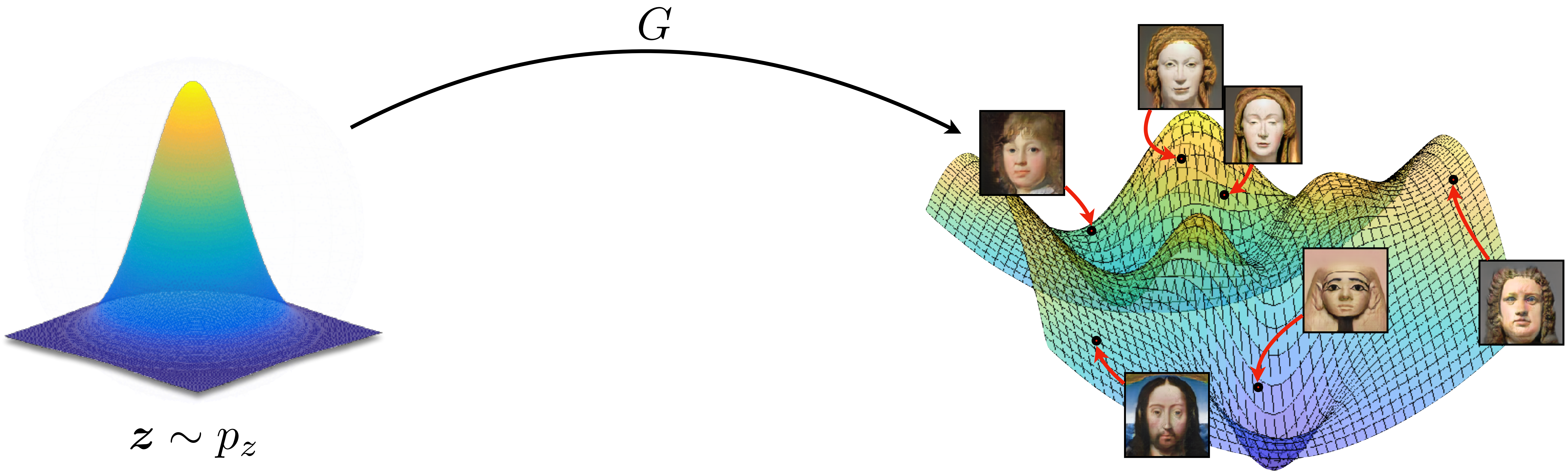
Our Motivation



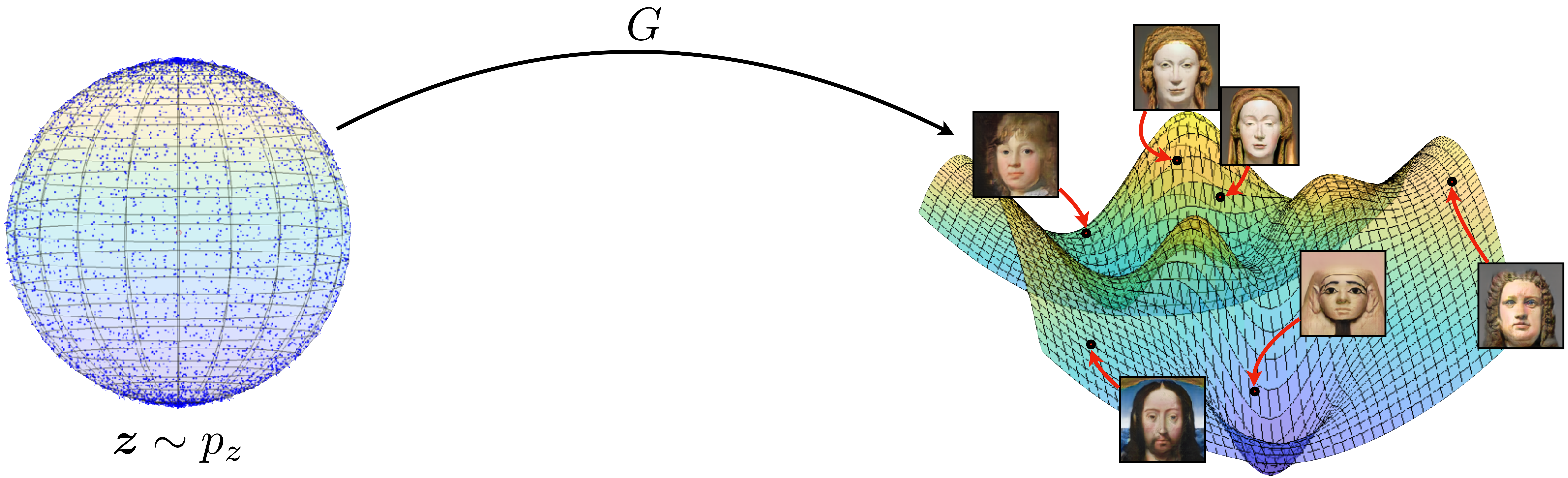
Our Motivation



Our Motivation



Our Motivation



State-of-the-art Approaches

- Various alternatives to the Gaussian/Uniform have been proposed – Gamma[2], Cauchy[3], Gaussian mixture[4], and non-parametric[5] priors.

State-of-the-art Approaches

- Various alternatives to the Gaussian/Uniform have been proposed – Gamma[2], Cauchy[3], Gaussian mixture[4], and non-parametric[5] priors.

- Image-to-image translation GANs[6−10] require *some form of* {semantic, pairwise, feature-level} consistency between inputs and outputs.

Fig. Domain adaptation

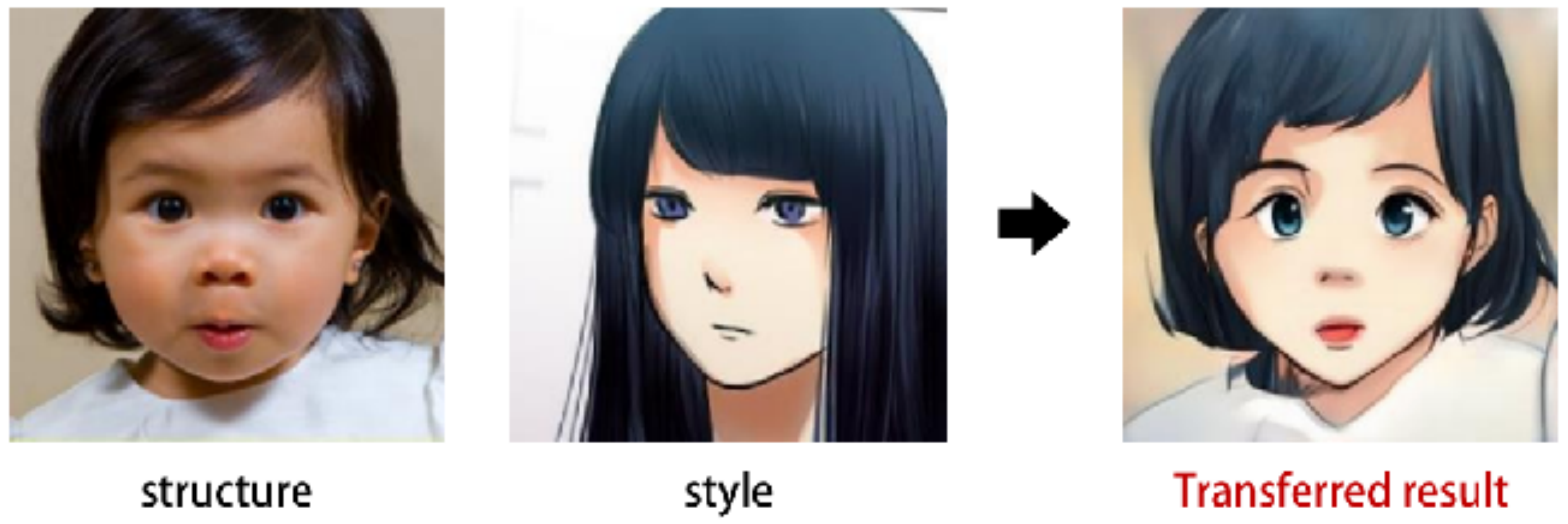


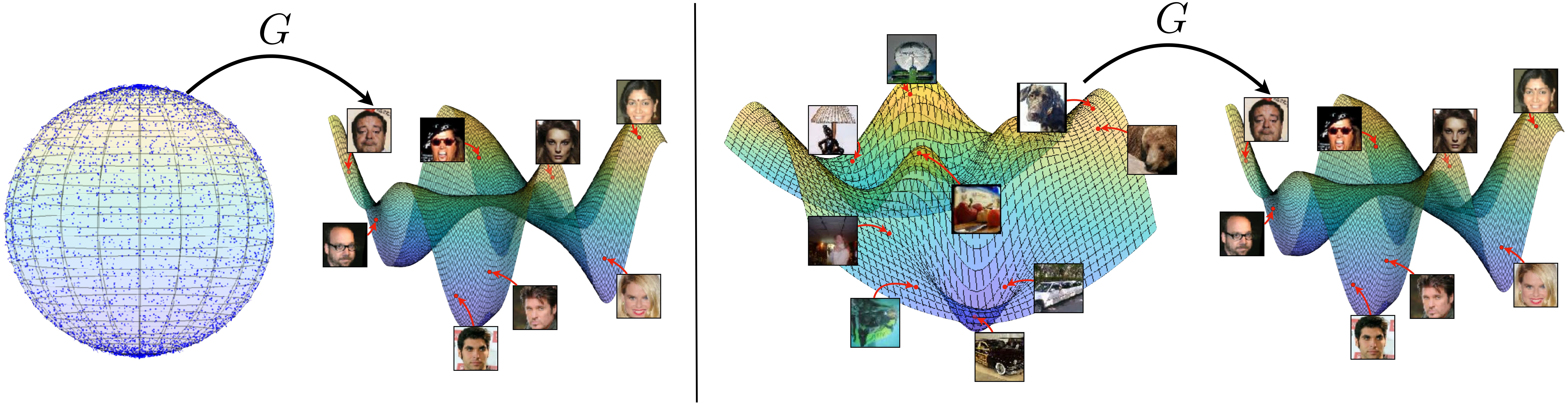
Fig. Inter-domain style transfer

Fig. Intra-domain style transfer

[02]Kilcher et al., ICLR 2018; [03]Leśniak et al., ICLR 2019; [04]Gurumurthy et al., CVPR 2017; [05]Singh et al., ICML 2019;
[06]Chen et al., NeurIPS 2016; [7]Karras et al., CVPR 2019; [08]Shen et al., CVPR 2020; [09]Isola et al., CVPR 2017; [10]Zhu et al., ICCV 2017

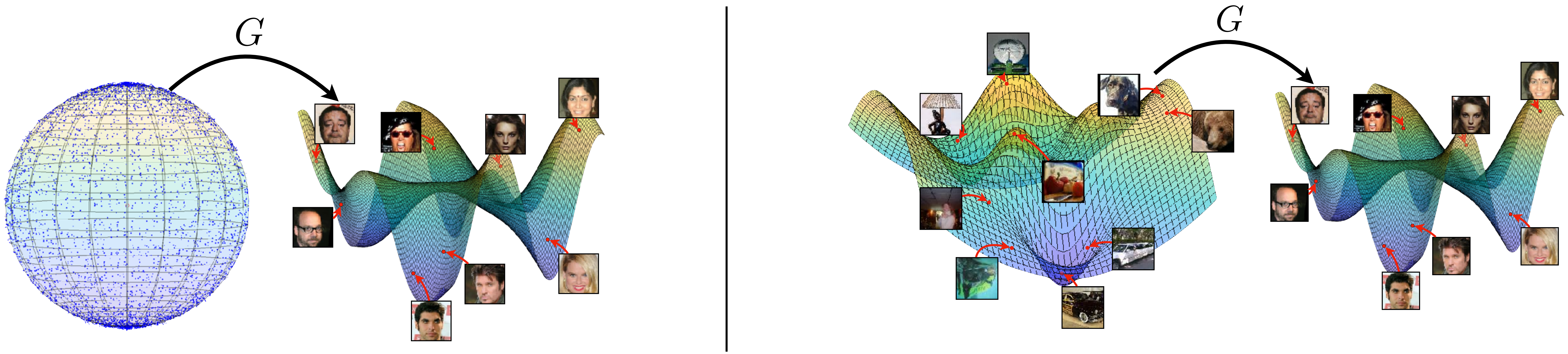
Spider GANs and "Friendly Neighborhoods"

- A alternative to improving the generator – Choose a "better" input distribution.
- Replace Gaussian noise with samples from a *closely related* dataset.
- A generalization of image-to-image translation GANs – dataset-to-dataset transformations!



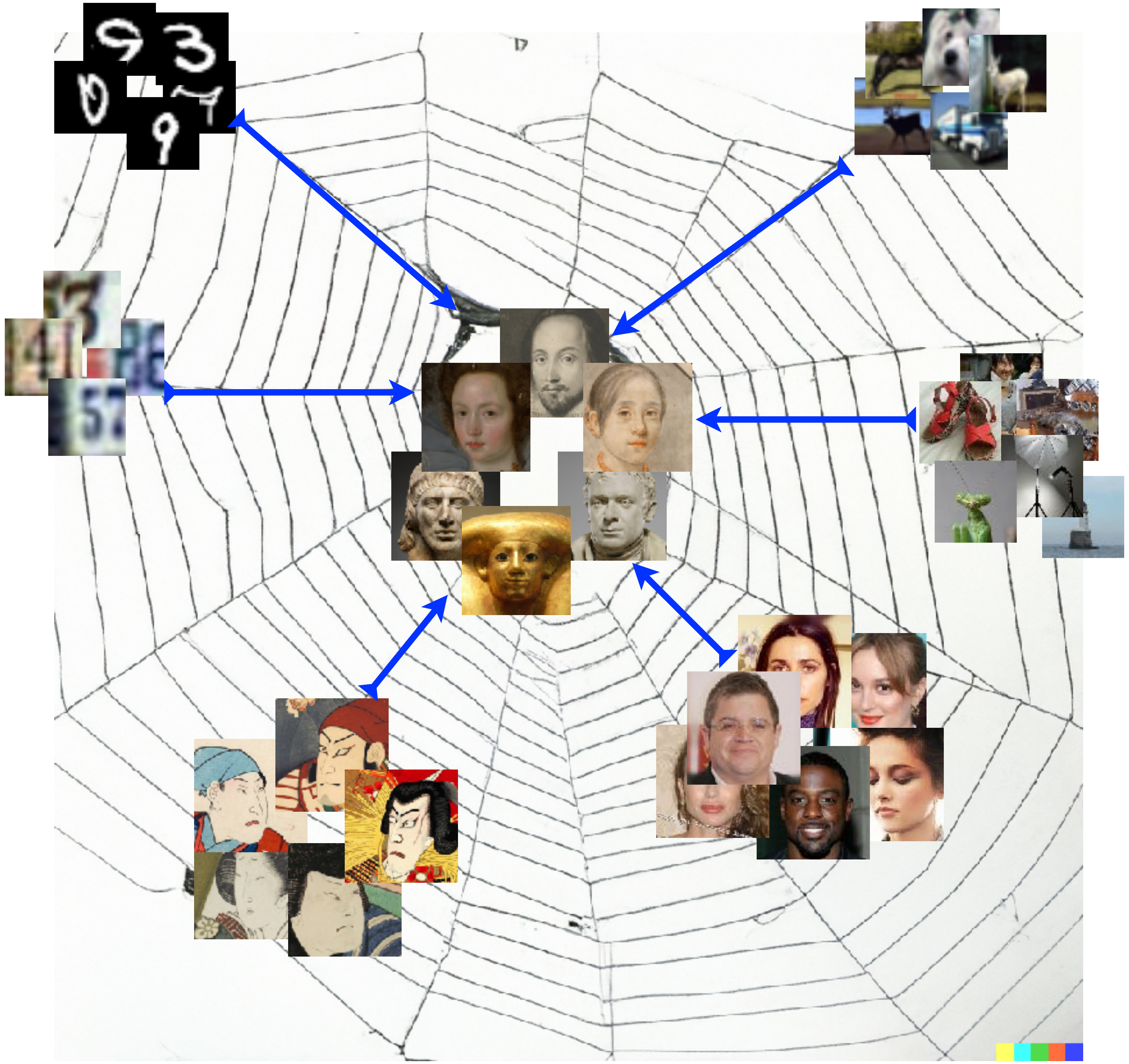
Spider GANs and "Friendly Neighborhoods"

- A alternative to improving the generator – Choose a "better" input distribution.
- Replace Gaussian noise with samples from a *closely related* dataset.
- A generalization of image-to-image translation GANs – dataset-to-dataset transformations!

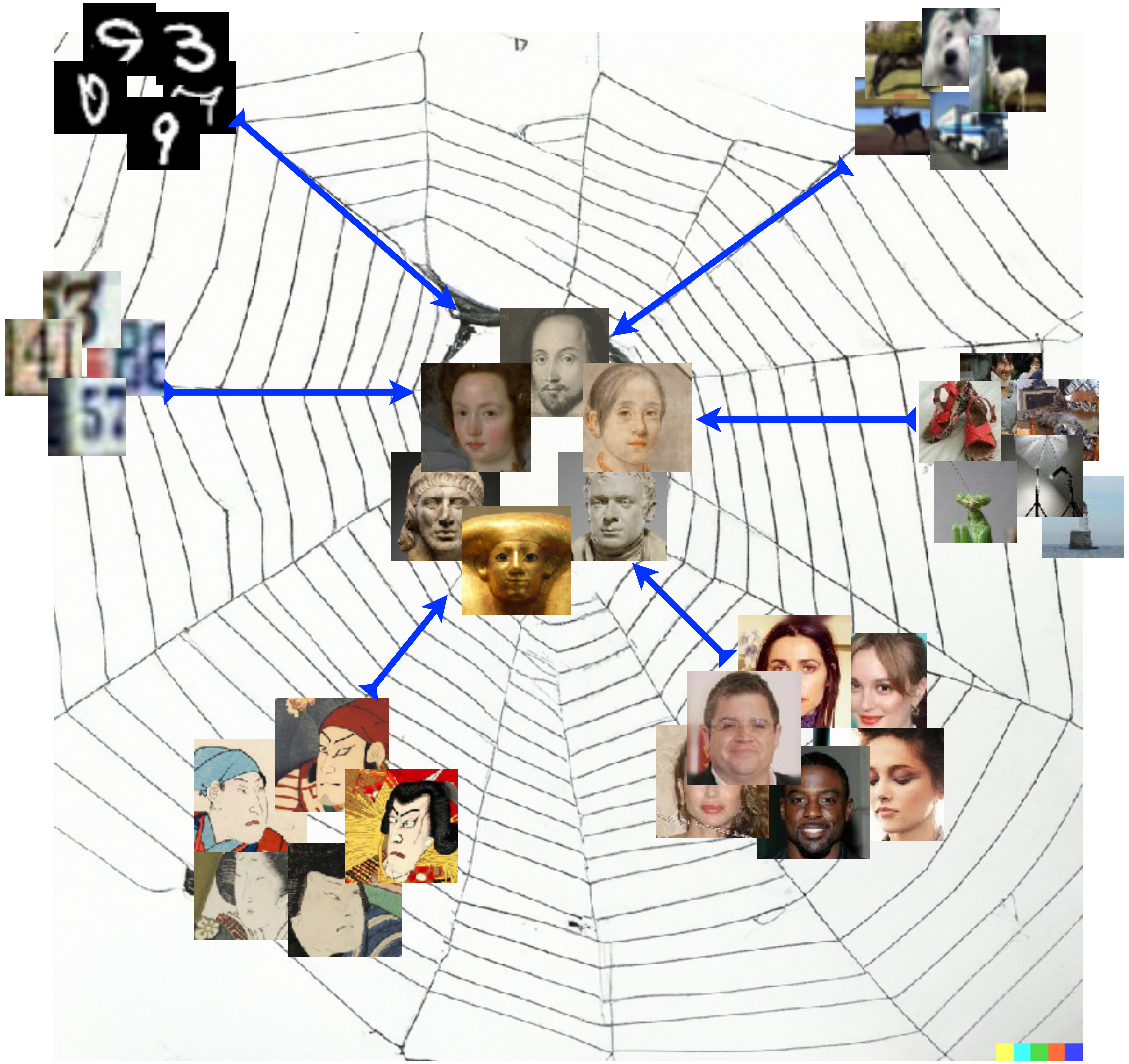


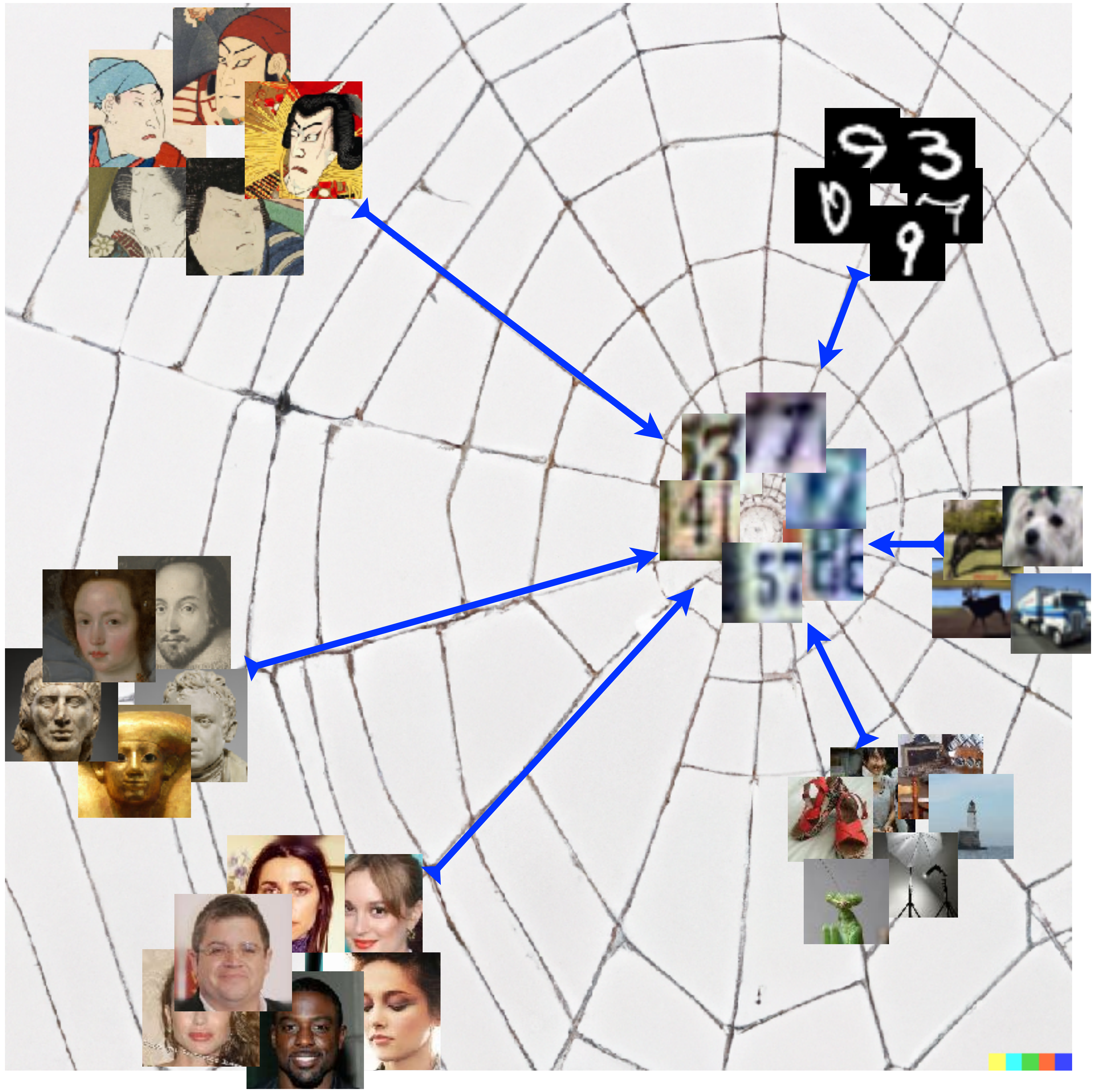
- How do we identify these *friendly neighborhood* datasets?

What is a “Friendly Neighborhood”?

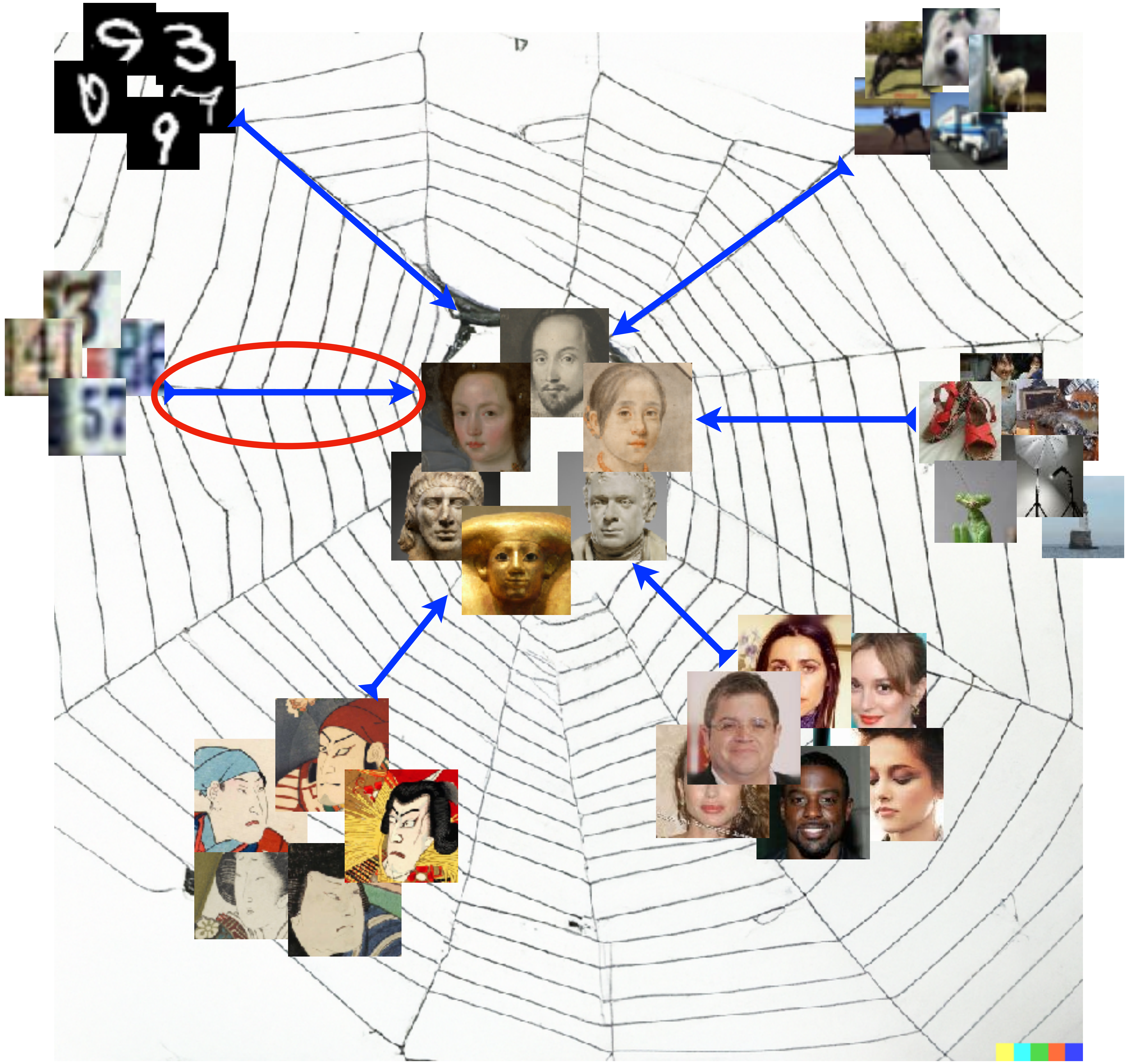


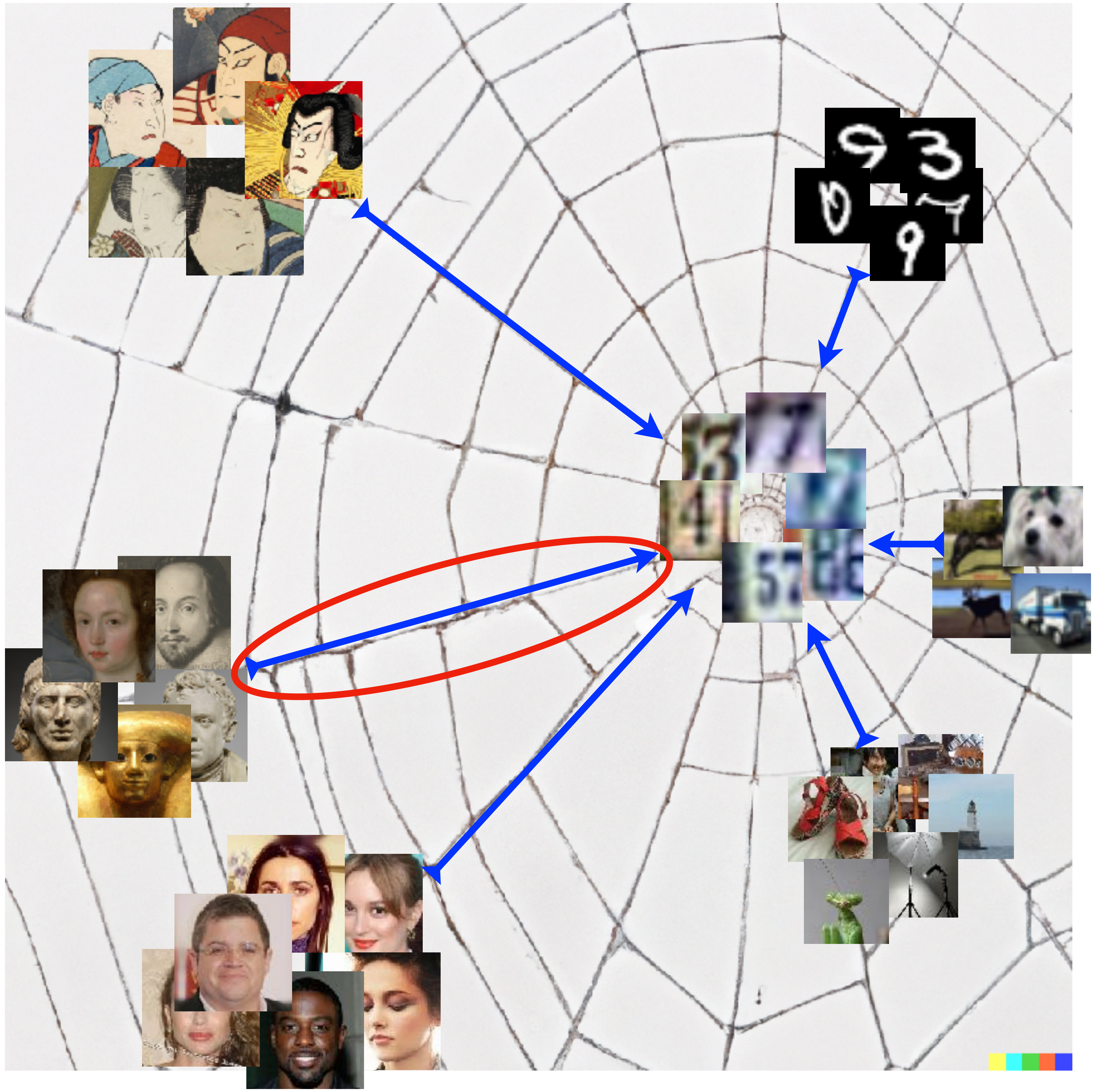
What is a "Friendly Neighborhood"?





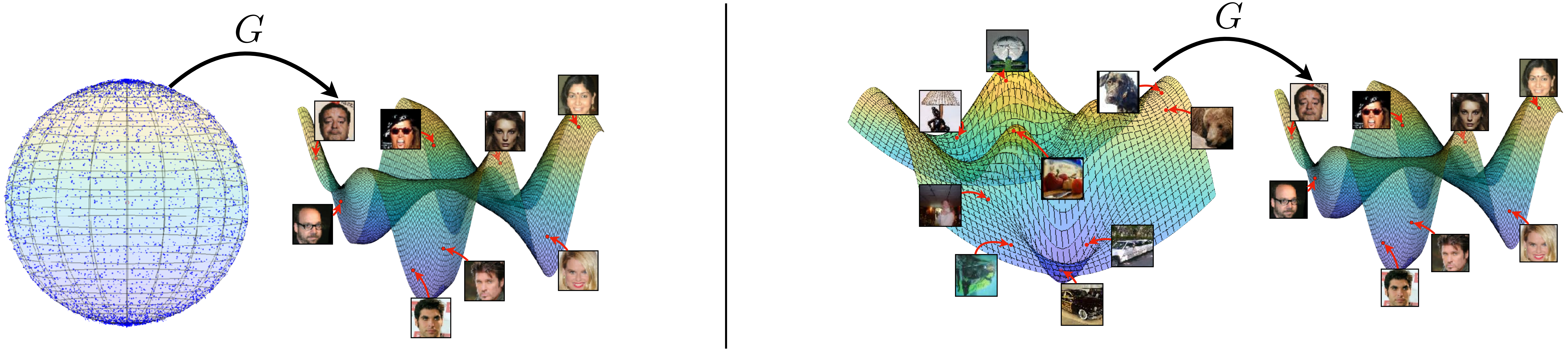
What is a "Friendly Neighborhood"?





Spider GANs and "Friendly Neighborhoods"

- A alternative to improving the generator – Choose a "better" input distribution.
- Replace Gaussian noise with samples from a *closely related* dataset.
- A generalization of image-to-image translation GANs – dataset-to-dataset transformations!



- The recently proposed Poly-LSGAN[11] discriminator can be used as a distance measure!

$$\tilde{D}^*_p(\boldsymbol{x}) = \frac{\kappa_{m,n}}{N}\left(\sum_{\boldsymbol{g}_j\sim p_g}\psi_{m,n}(\boldsymbol{x}-\boldsymbol{g}_j) - \sum_{\boldsymbol{d}_i\sim p_d}\psi_{m,n}(\boldsymbol{x}-\boldsymbol{d}_i)\right), \text{ where } \psi_{m,n}(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\|^{2m-n} & \text{if } 2m-n<0 \text{ or } n \text{ is odd} \\ \|\boldsymbol{x}\|^{2m-n}\ln(\|\boldsymbol{x}\|) & \text{if } 2m-n\geq 0 \text{ and } n \text{ is even} \end{cases}.$$

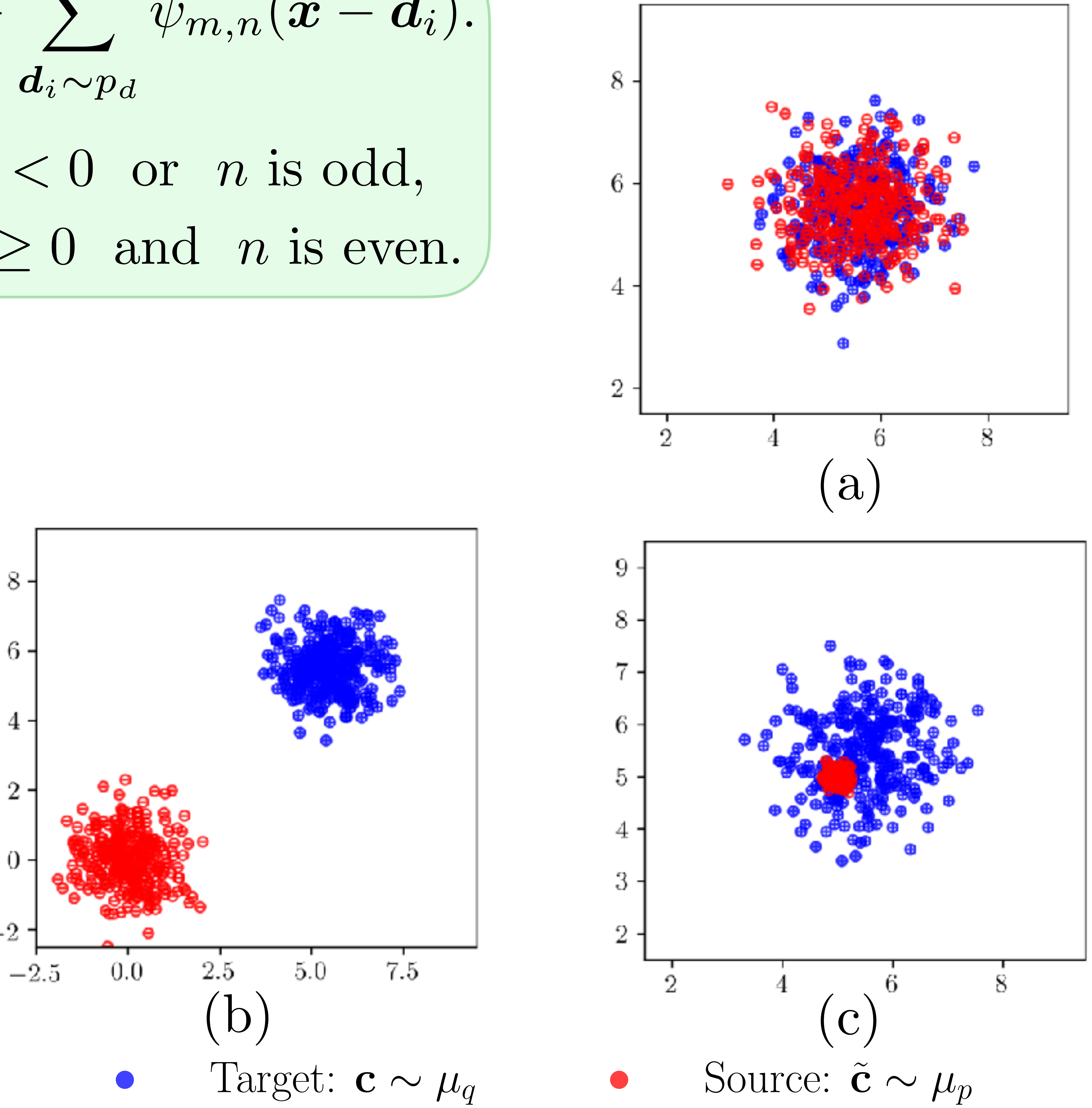
[11] Asokan and Seelamantula, NeurIPS DLDE-II 2022

The Signed Distance

$$\tilde{D}_p^*(\boldsymbol{x}) = \frac{\kappa_{m,n}}{N_g} \sum_{\boldsymbol{g}_j \sim p_g} \psi_{m,n}(\boldsymbol{x} - \boldsymbol{g}_j) - \frac{\kappa_{m,n}}{N_d} \sum_{\boldsymbol{d}_i \sim p_d} \psi_{m,n}(\boldsymbol{x} - \boldsymbol{d}_i).$$

$$\text{where } \psi_{m,n}(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\|^{2m-n} & \text{if } \ 2m-n < 0 \ \text{ or } \ n \text{ is odd,} \\ \|\boldsymbol{x}\|^{2m-n} \ln(\|\boldsymbol{x}\|) & \text{if } \ 2m-n \geq 0 \ \text{ and } \ n \text{ is even.} \end{cases}$$

- $D_p^*(\boldsymbol{x})$ has promising properties!
- Consider a test point $\boldsymbol{x}$ drawn from p_d:
  - (a) If $p_g = p_d$, we have $D_p^*(\boldsymbol{x}) = 0, \ \forall \, \boldsymbol{x}$.
  - (b) If p_g is far from p_d, $D_p^*(\boldsymbol{x}) > 0$.
  - (c) If p_g has *mode-collapsed*, $D_p^*(\boldsymbol{x}) < 0$.

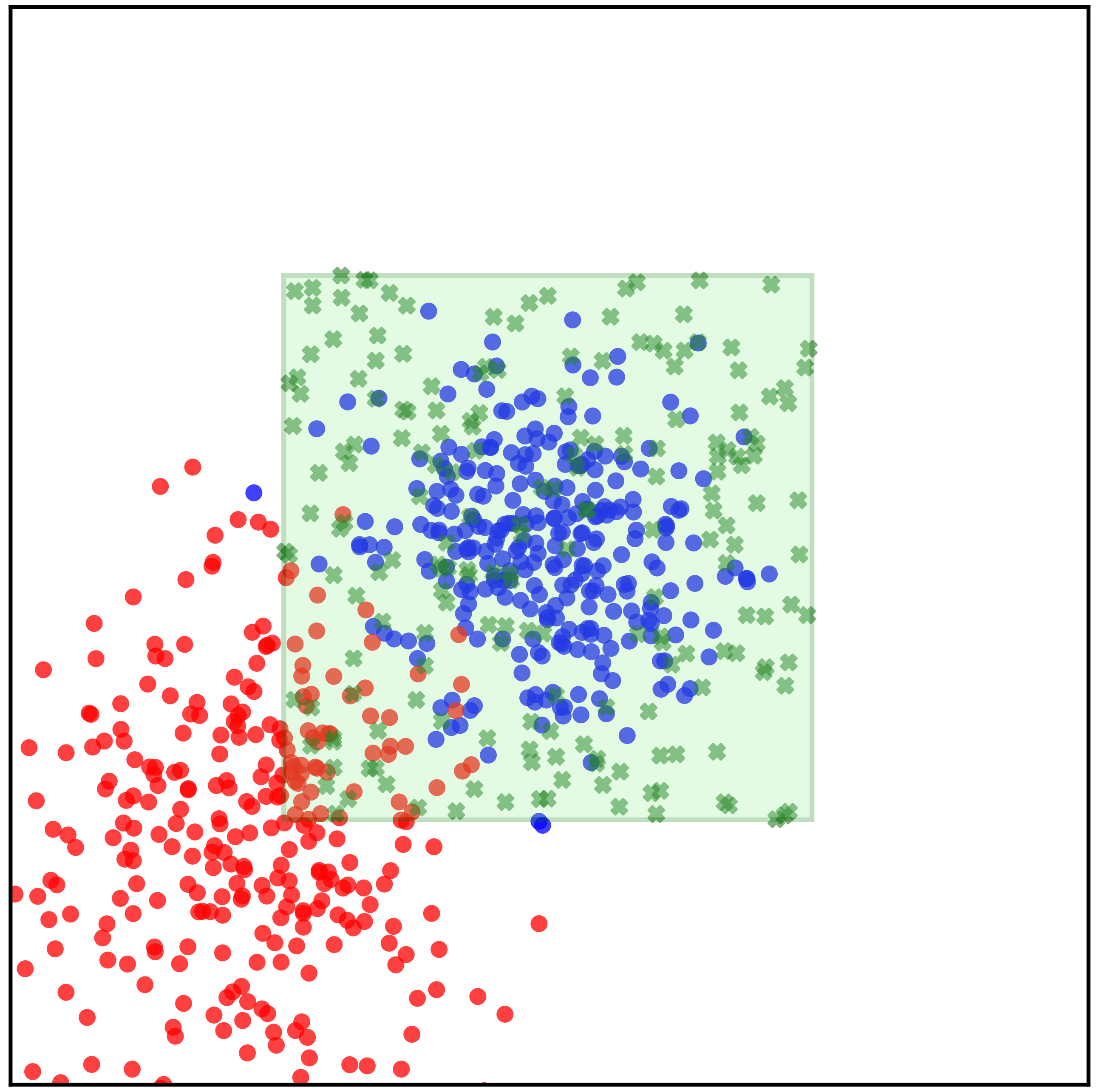


The Signed Distance

$$\sum_{\boldsymbol{x}} \tilde{D}_p^*(\boldsymbol{x}) \approx \mathrm{SD}(p_s \| p_t) = \sum_{\boldsymbol{x} \in \mathcal{U}[\mathcal{C}_r]} \left(\sum_{d_i \sim p_s} \psi_{m,n}(\boldsymbol{x} - \boldsymbol{d}_i) - \sum_{g_j \sim p_t} \psi_{m,n}(\boldsymbol{x} - \boldsymbol{g}_j) \right).$$

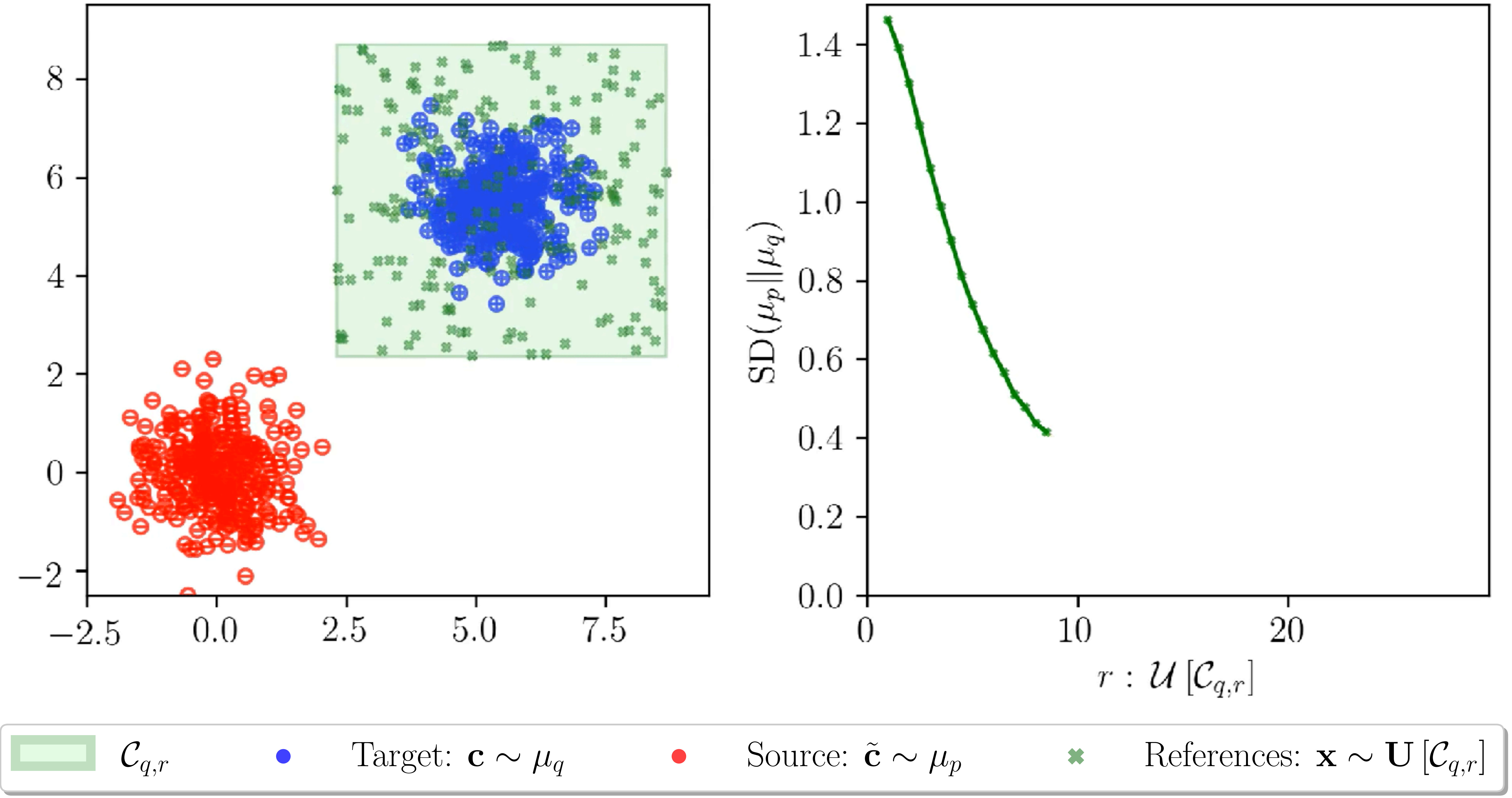
$$\text{where } \psi_{m,n}(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\|^{2m-n} & \text{if } \ 2m - n < 0 \ \text{ or } \ n \text{ is odd,} \\ \|\boldsymbol{x}\|^{2m-n} \ln(\|\boldsymbol{x}\|) & \text{if } \ 2m - n \geq 0 \ \text{ and } \ n \text{ is even.} \end{cases}$$

- Compute **the signed distance** $\mathrm{SD}(p_s \| p_t)$ of a source p_s w.r.t. a reference *target* p_t as the sum of the discriminator D_p^* evaluated over an entire batch of samples $\{\boldsymbol{x}_\ell\}_{\ell=1}^M$ drawn from a cube $\mathcal{C}_r$ of side r, centered around target p_t.

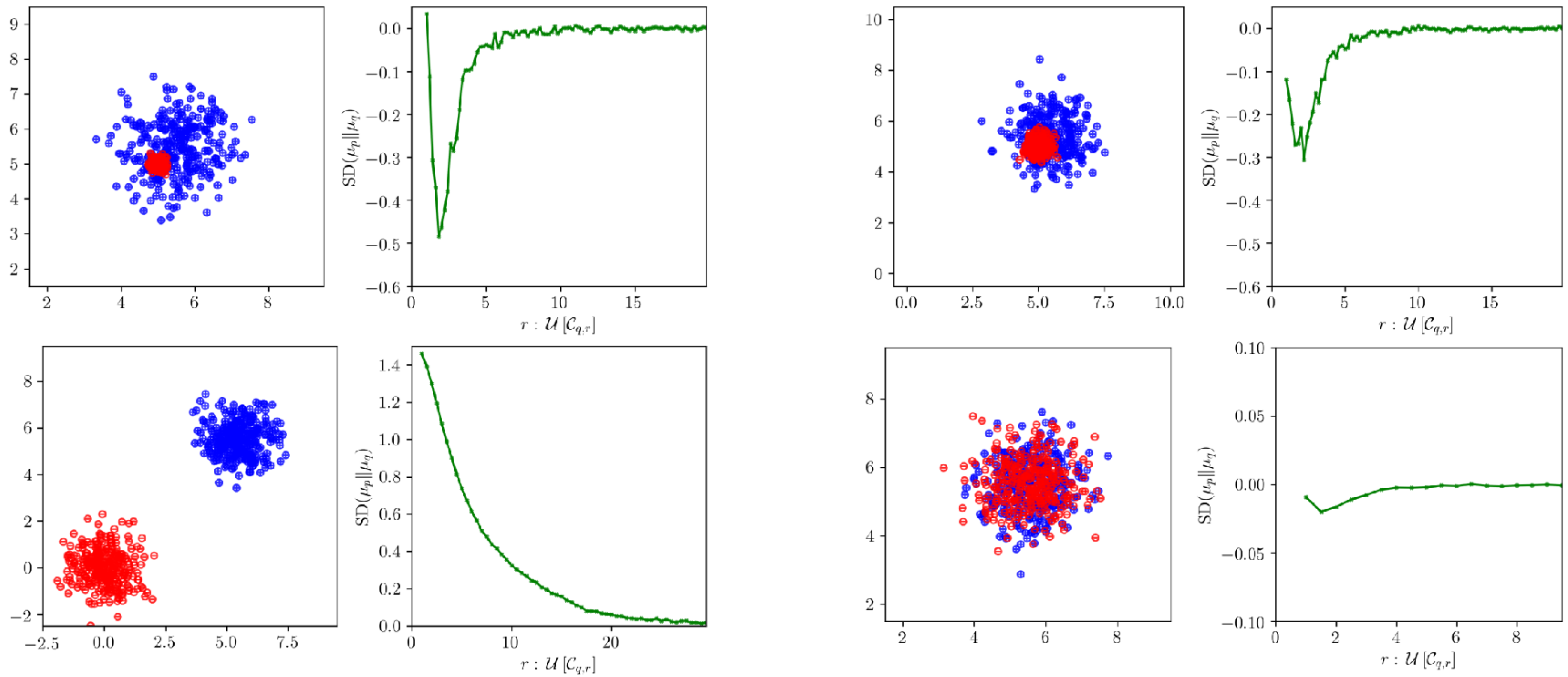


$\mathcal{C}_{q,r}$ • Target: $\mathbf{c} \sim \mu_q$ • Source: $\tilde{\mathbf{c}} \sim \mu_p$ ✖ References: $\mathbf{x} \sim \mathbf{U}[\mathcal{C}_{q,r}]$

The Signed Distance



The Signed Distance



$\mathcal{C}_{q,r}$ Target: $\mathbf{c} \sim \mu_q$ Source: $\tilde{\mathbf{c}} \sim \mu_p$ References: $\mathbf{x} \sim \mathbf{U}[\mathcal{C}_{q,r}]$

The Signed Inception Distance

$$\sum_{\boldsymbol{x}} \tilde{D}_p^*(\boldsymbol{x}) \approx \mathrm{SD}(p_s \| p_t) = \sum_{\boldsymbol{x} \in \mathcal{U}[\mathcal{C}_r]} \left(\sum_{d_i \sim p_s} \psi_{m,n}(\boldsymbol{x} - \boldsymbol{d}_i) - \sum_{g_j \sim p_t} \psi_{m,n}(\boldsymbol{x} - \boldsymbol{g}_j) \right).$$

where $\psi_{m,n}(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\|^{2m-n} & \text{if } \ 2m-n<0 \ \text{ or } \ n \text{ is odd,} \\ \|\boldsymbol{x}\|^{2m-n} \ln(\|\boldsymbol{x}\|) & \text{if } \ 2m-n \geq 0 \ \text{ and } \ n \text{ is even.} \end{cases}$

- Compute **the signed distance** $\mathrm{SD}(p_s \| p_t)$ of a source p_s w.r.t. a reference *target* p_t as the sum of the discriminator D_p^* evaluated over an entire batch of samples $\{\boldsymbol{x}_\ell\}_{\ell=1}^{M}$ drawn from a cube $\mathcal{C}_r$ of side r, centered around target p_t.

- How do we extend this to image data?

The Signed Inception Distance

$$\sum_{\boldsymbol{x}} \tilde{D}_p^*(\boldsymbol{x}) \approx \mathrm{SD}(p_s \| p_t) = \sum_{\boldsymbol{x} \in \mathcal{U}[\mathcal{C}_r]} \left(\sum_{d_i \sim p_s} \psi_{m,n}(\boldsymbol{x} - \boldsymbol{d}_i) - \sum_{g_j \sim p_t} \psi_{m,n}(\boldsymbol{x} - \boldsymbol{g}_j) \right).$$

where $\psi_{m,n}(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\|^{2m-n} & \text{if } 2m-n<0 \text{ or } n \text{ is odd,} \\ \|\boldsymbol{x}\|^{2m-n} \ln(\|\boldsymbol{x}\|) & \text{if } 2m-n \geq 0 \text{ and } n \text{ is even.} \end{cases}$

- Compute **the signed distance** $\mathrm{SD}(p_s \| p_t)$ of a source p_s w.r.t. a reference *target* p_t as the sum of the discriminator D_p^* evaluated over an entire batch of samples $\{\boldsymbol{x}_\ell\}_{\ell=1}^M$ drawn from a cube $\mathcal{C}_r$ of side r, centered around target p_t.

- How do we extend this to image data?

- The **signed Inception distance**, $\mathrm{SID}(p_s \| p_t)$: Compute $\mathrm{SD}(p_s \| p_t)$ over Inception embeddings.

The Signed Inception Distance

$$\sum_{\boldsymbol{x}} \tilde{D}_p^*(\boldsymbol{x}) \approx \mathrm{SID}(p_s\|p_t) = \sum_{\boldsymbol{x}\in\mathcal{U}[\mathcal{C}_r]} \left(\sum_{d_i\sim p_s} \psi_{m,n}(\mathrm{I}_{v3}(\boldsymbol{x}) - \mathrm{I}_{v3}(\boldsymbol{d}_i)) - \sum_{g_j\sim p_t} \psi_{m,n}(\mathrm{I}_{v3}(\boldsymbol{x}) - \mathrm{I}_{v3}(\boldsymbol{g}_j)) \right).$$

$$\text{where } \psi_{m,n}(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\|^{2m-n} & \text{if } 2m-n<0 \text{ or } n \text{ is odd,} \\ \|\boldsymbol{x}\|^{2m-n}\ln(\|\boldsymbol{x}\|) & \text{if } 2m-n\geq 0 \text{ and } n \text{ is even.} \end{cases}$$

- Compute **the signed distance** $\mathrm{SD}(p_s\|p_t)$ of a source p_s w.r.t. a reference *target* p_t as the sum of the discriminator D_p^* evaluated over an entire batch of samples $\{\boldsymbol{x}_\ell\}_{\ell=1}^{M}$ drawn from a cube $\mathcal{C}_r$ of side r, centered around target p_t.

- How do we extend this to image data?

- **Signed Inception distance** ($\mathrm{SID}(p_s\|p_t)$): Compute $\mathrm{SD}(p_s\|p_t)$ over Inception (I_{v3}) embeddings.

SID and Spider GANs

- Identify *friendly neighbors* p_s of a target p_t, by computing area under the $(\mathrm{SID}(p_s \| p_t))$ curve.
- Train GANs with the input samples drawn from the *friendly neighbors*.

SID and Spider GANs

- Identify *friendly neighbors* p_s of a target p_t, by computing area under the $(\mathrm{SID}(p_s \| p_t))$ curve.
- Train GANs with the input samples drawn from the *friendly neighbors*.

Table: Area under the SID curve between various Target and Source datasets can be used to identify the first, second and third "friendliest neighbors" of a dataset.

Src \ Tar	MNIST	F-MNIST	SVHN	CIFAR-10	T-ImgNet	CelebA	Ukiyo-E	Church
MNIST	0.1865	**21.886**	37.227	29.298	9.436	198.714	201.550	205.322
F-MNIST	**162.962**	0.1097	46.938	19.051	-0.5571	167.840	191.010	181.458
SVHN	**212.473**	77.357	-0.0566	**34.534**	**21.668**	195.631	214.507	219.790
CIFAR-10	221.337	**65.426**	**52.051**	-0.1478	-7.109	**180.491**	198.991	**173.655**
T-ImgNet	230.916	75.737	**67.902**	**12.892**	0.6743	**157.520**	**197.447**	**184.977**
CelebA	**204.794**	68.828	65.299	**23.685**	**8.829**	0.6241	**184.170**	191.927
Ukiyo-E	250.226	92.741	82.157	39.792	**18.727**	191.930	0.5494	**180.697**
Church	212.452	**48.676**	**56.136**	-4.655	-23.115	**185.740**	**198.750**	-0.5258

SID and Spider GANs

- Identify *friendly neighbors* p_s of a target p_t, by computing area under the $(\text{SID}(p_s \| p_t))$ curve.
- Train GANs with the input samples drawn from the *friendly neighbors*.

Table: Area under the SID curve between various Target and Source datasets is used to identify the first, second and third "friendliest neighbors" of a dataset.

Src \ Tar	MNIST	F-MNIST	SVHN	CIFAR-10	T-ImgNet	CelebA	Ukiyo-E	Church
MNIST	0.1865	**21.886**	37.227	29.298	9.436	198.714	201.550	205.322
F-MNIST	**162.962**	0.1097	46.938	19.051	-0.5571	167.840	191.010	181.458
SVHN	**212.473**	77.357	-0.0566	**34.534**	**21.668**	195.631	214.507	219.790
CIFAR-10	221.337	**65.426**	**52.051**	-0.1478	-7.109	**180.491**	198.991	**173.655**
T-ImgNet	230.916	75.737	**67.902**	**12.892**	0.6743	**157.520**	**197.447**	**184.977**
CelebA	**204.794**	68.828	65.299	**23.685**	**8.829**	0.6241	**184.170**	191.927
Ukiyo-E	250.226	92.741	82.157	39.792	**18.727**	191.930	0.5494	**180.697**
Church	212.452	**48.676**	**56.136**	-4.655	-23.115	**185.740**	**198.750**	-0.5258

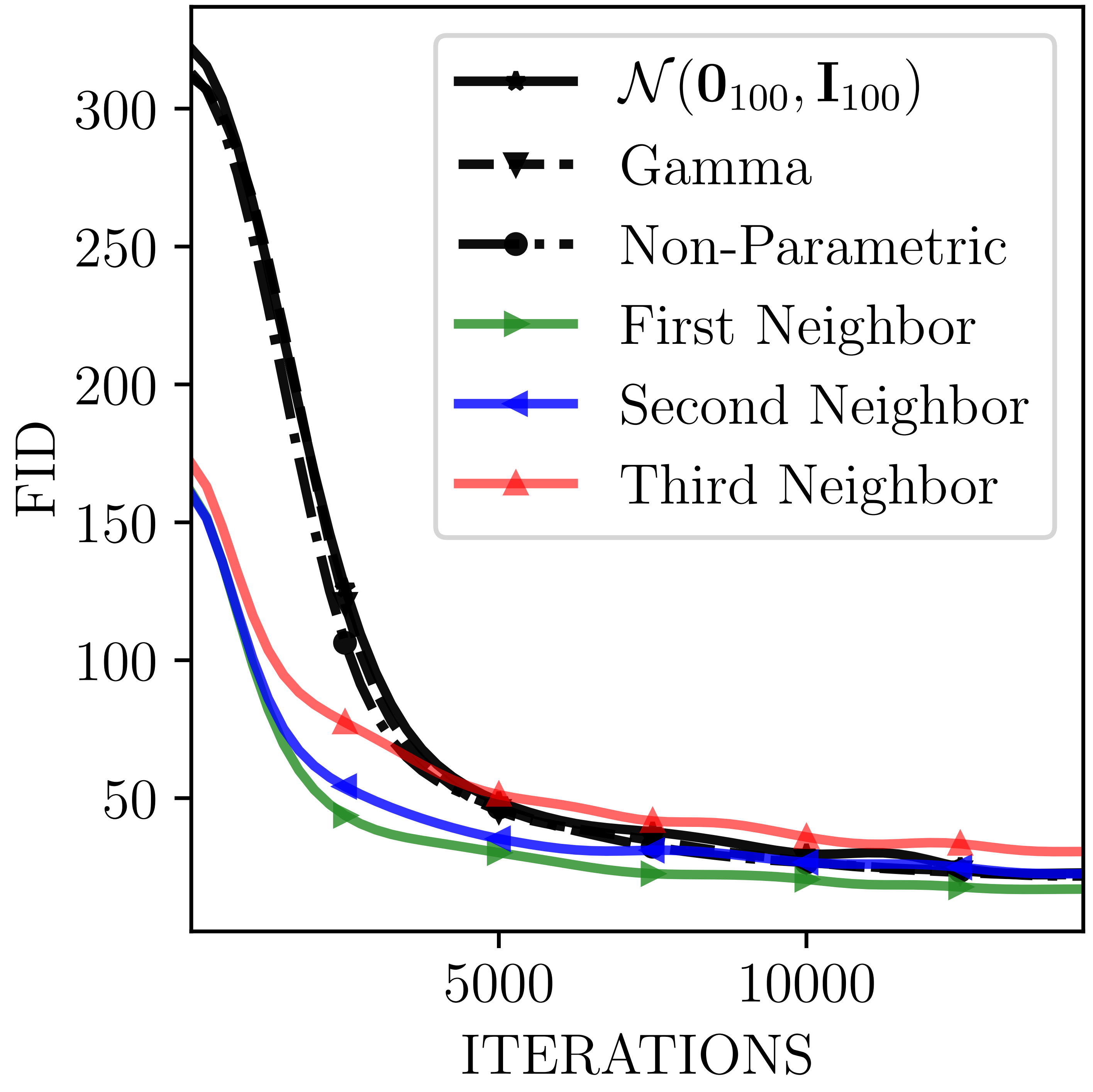
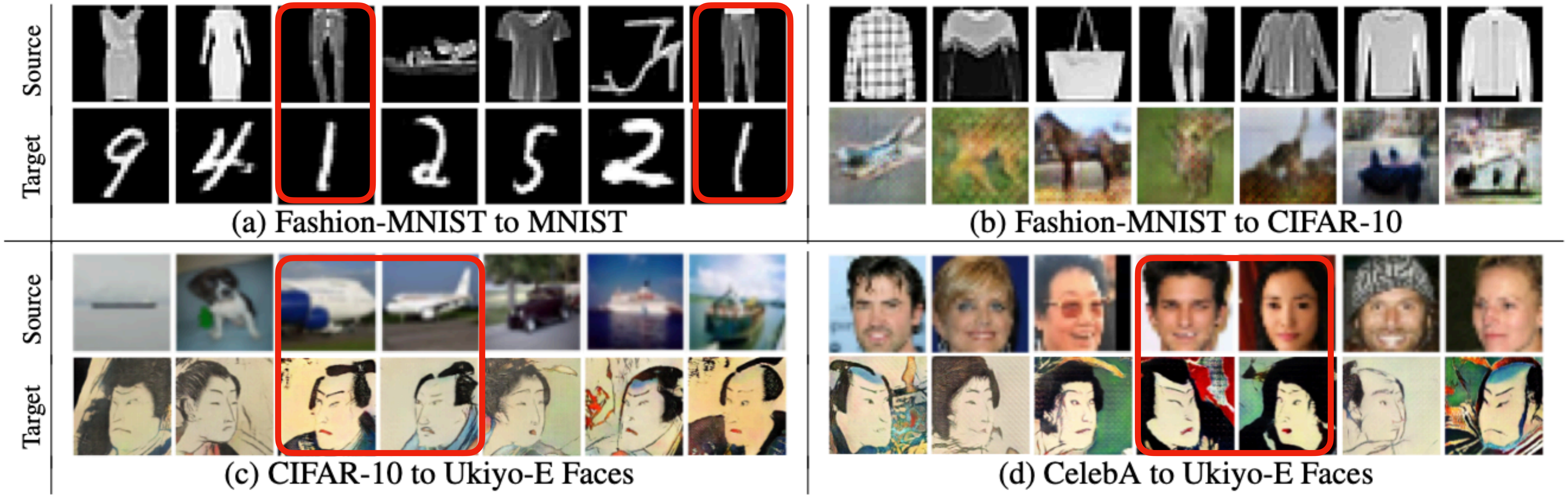


Fig: FID vs. Iterations on learning the MNIST dataset for various choices of input datasets.

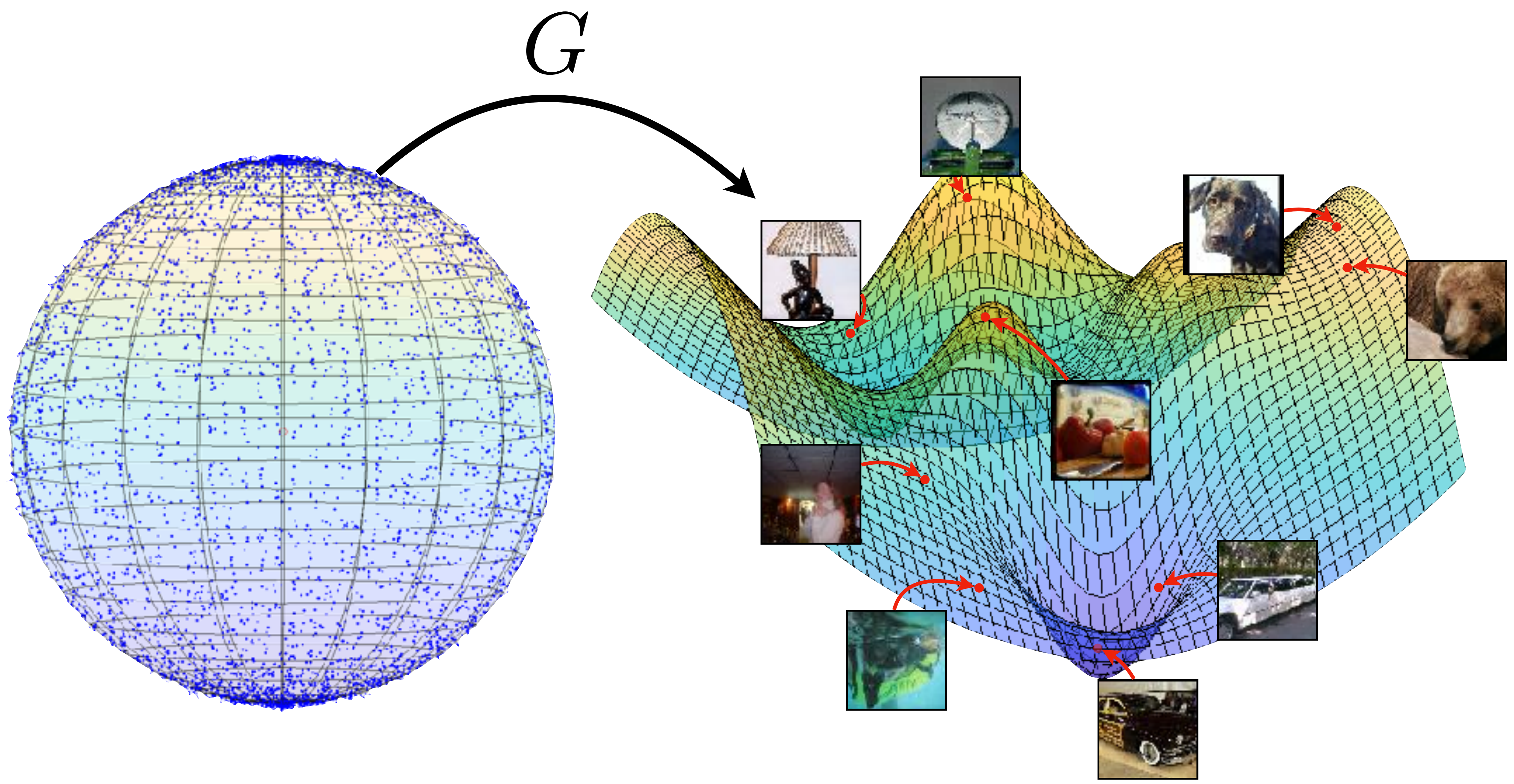
Learning from the Friendly Neighbors

- The generator learns transformations between datasets
- Visual similarity in not necessary; Underling structural similarity is implicitly leveraged.

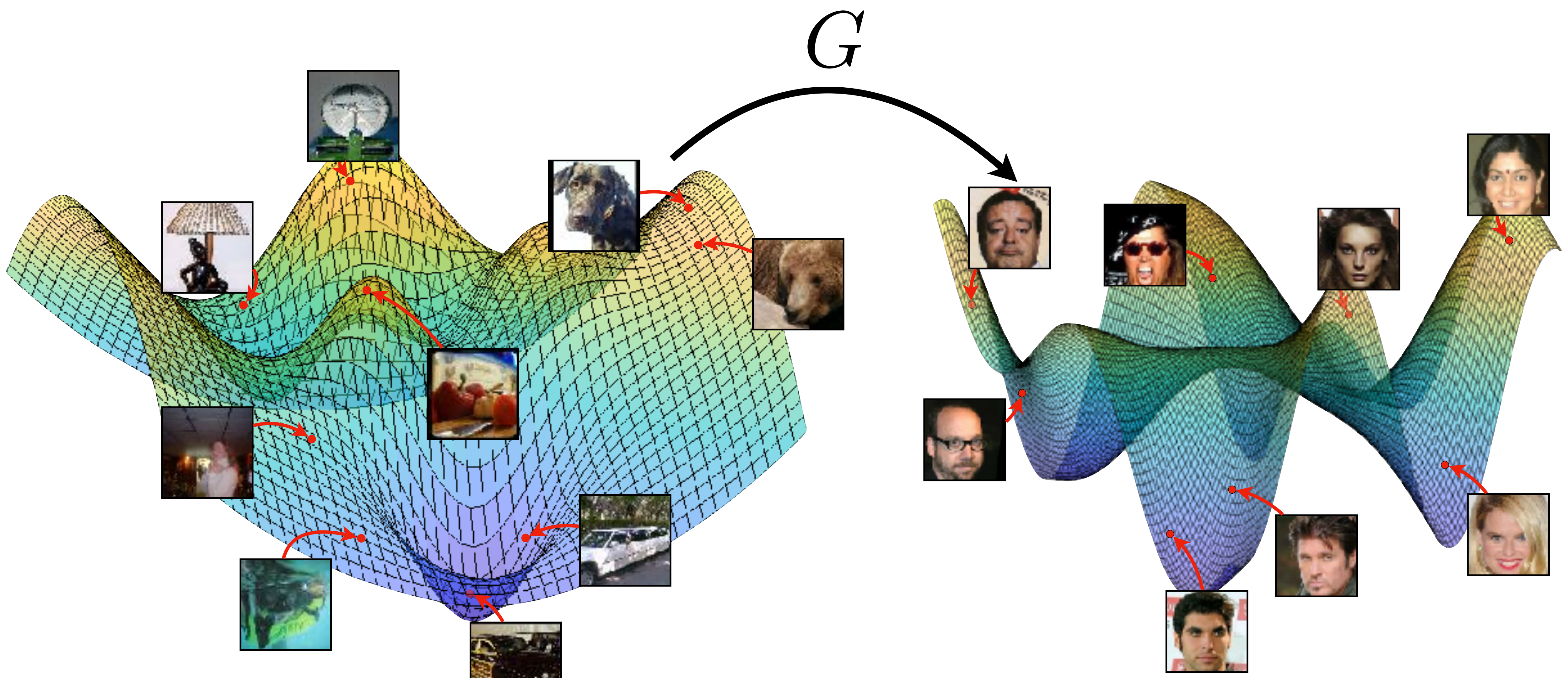


Cascading Spider GANs

Standard GAN



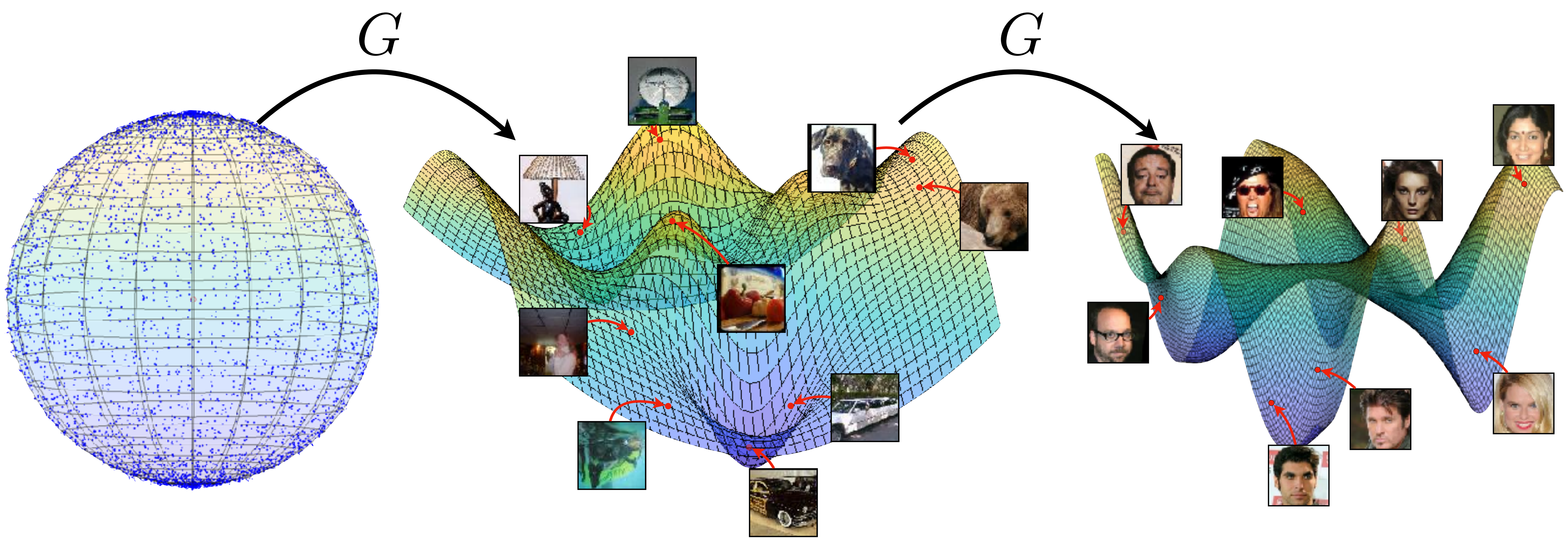
Spider GAN



Cascading Spider GANs

Pre-trained Stage-I GAN

Stage-II Spider GAN



Cascading Spider GANs

- Cascaded Spider Style GANs achieve **state-of-the-art FID** scores in **a fifth of the training iterations** compared to baseline PGGAN[12], StyleGAN2-ADA[13] and StyleGAN3[14]!

Table. Performance of AFHQ-Cats

Architecture	Weight Transfer	Input Distribution	Training steps	FID	KID ($\times 10^{-3}$)
StyleGAN2-ADA	–	Gaussian	25000	$5.13^{\star}$	$1.54^{\star}$
StyleGAN3-T	–	Gaussian	25000	$4.04^{\dagger}$	–
Spider StyleGAN3-T **(Ours)**	–	AFHQ-Dogs	5000	6.29	1.64
StyleGAN2-ADA	FFHQ	Gaussian	5000	3.55	0.35
Spider StyleGAN2-ADA **(Ours)**	FFHQ	Tiny-ImageNet	1000	3.91	1.23
StyleGAN2-ADA	AFHQ-Dogs	Gaussian	5000	$3.47^{\star}$	$0.37^{\star}$
Spider StyleGAN2-ADA **(Ours)**	AFHQ-Dogs	Tiny-ImageNet	1500	**3.07**	**0.29**
Spider StyleGAN3-T **(Ours)**	AFHQ-Dogs	Tiny-ImageNet	1000	3.86	1.01

Table. Performance of Ukiyo-E Faces and MetFaces

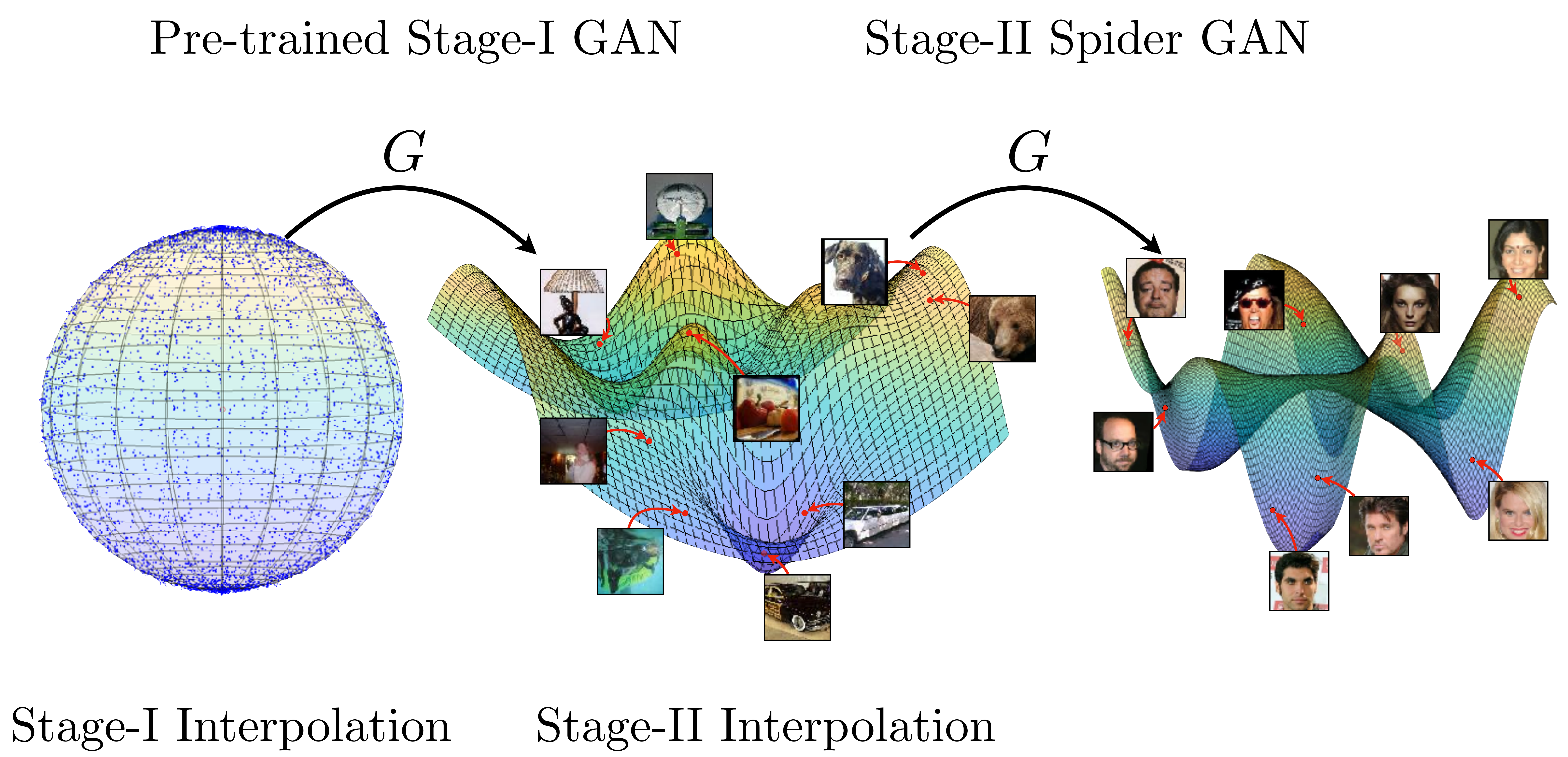
Architecture	Input	Ukiyo-E Faces		MetFaces	
		FID	KID	FID	KID
PGGAN	Gaussian	69.03	0.0762	85.74	0.0123
Spider PGGAN **Ours)**	TinyImageNet	57.63	0.0161	45.32	0.0063
StyleGAN2*	Gaussian	56.74	0.0159	65.74	0.0350
StyleGAN2-ADA*	Gaussian	26.74	0.0109	18.75	**0.0023**
Spider StyleGAN2 **(Ours)**	TinyImageNet	**20.44**	**0.0059**	**15.60**	0.0026
Spider StyleGAN2 **(Ours)**	AFHQ-Dogs	32.59	0.0269	29.82	0.0019

Table. Performance of FFHQ

Architecture	Input	FID	KID	CSID_m
StyleGAN2-ADA	Gaussian	$2.70^{\dagger}$	0.906×10^{-3}	2.65
StyleGAN3-T	Gaussian	$2.79^{\dagger}$	1.031×10^{-3}	2.95
StyleGAN-XL	Gaussian	$\mathbf{2.02}^{\dagger}$	$\mathbf{0.287 \times 10^{-3}}$	3.94
Spider StyleGAN2-ADA **(Ours)**	TinyImageNet	2.45	0.915×10^{-3}	**1.99**
Spider StyleGAN2-ADA **(Ours)**	AFHQ-Dogs	3.07	0.795×10^{-3}	2.55
Spider StyleGAN3-T **(Ours)**	TinyImageNet	2.86	1.162×10^{-3}	3.25

[12]Karras et al., ICLR, 2018; [13]Karras et al., CVPR, 2021; [14]Karras et al., NeurIPS 2022

Cascading Spider GANs



Cascading Spider GANs

Fig. Interpolations on Stage-I Spider StyleGAN2-ADA

Fig. Interpolations on Stage-II Spider StyleGAN2-ADA

Cascading Spider GANs

Conclusions

- We developed a novel framework for training GANs with *friendly neighbor* inputs.
- SID: A kernel-based discriminator employed as an evaluation/distance measure.
- GANs trained with closely-related datasets as input outperform baselines in fewer iterations.

Conclusions

- We developed a novel framework for training GANs with *friendly neighbor* inputs.
- SID: A kernel-based discriminator employed as an evaluation/distance measure.
- GANs trained with closely-related datasets as input outperform baselines in fewer iterations.

Spider StyleGANs

SID

References

[01] Goodfellow et al., “Generative adversarial networks,” NeurIPS 2014

[02] Kilcher et al., “Semantic interpo- lation in implicit models,” ICLR 2018

[03] Leśniak et al., “Distribution-interpolation trade off in generative models,” ICLR 2019

[04] Gurumurthy et al., “DeLi-GAN: Generative adversarial networks for diverse and limited data,” CVPR 2017

[05] Singh et al., “Non-parametric priors for generative adversarial networks,” ICML 2019

[06] Chen et al., “InfoGAN: Interpretable representation learning by information maximizing GANs,” NeurIPS 2016

[07] Karras et al., “A style-based generator architecture for generative adversarial networks,” CVPR 2019

[08] Shen et al., “Interpreting the latent space of GANs for semantic face editing,” CVPR 2020

[09] Isola et al., “Image-to-image translation with conditional adversarial networks,” CVPR 2017

[10] Zhu et al., “Unpaired image-to-image translation using cycle-consistent adversarial networks,” ICCV 2017

[11] Asokan and Seelamantula, “Bridging the Gap Between Coulomb GAN and Gradient-regularized WGAN,” NeurIPS DLDE 2022

[12] Karras et al., “Progressive growing of GANs for improved quality, stability, and variation,” ICLR, 2018

[13] Karras et al., “Training generative adversarial networks with limited data,” CVPR, 2021

[14] Karras et al., “Alias-free generative adversarial networks,” NeurIPS, 2022

Thank You!!