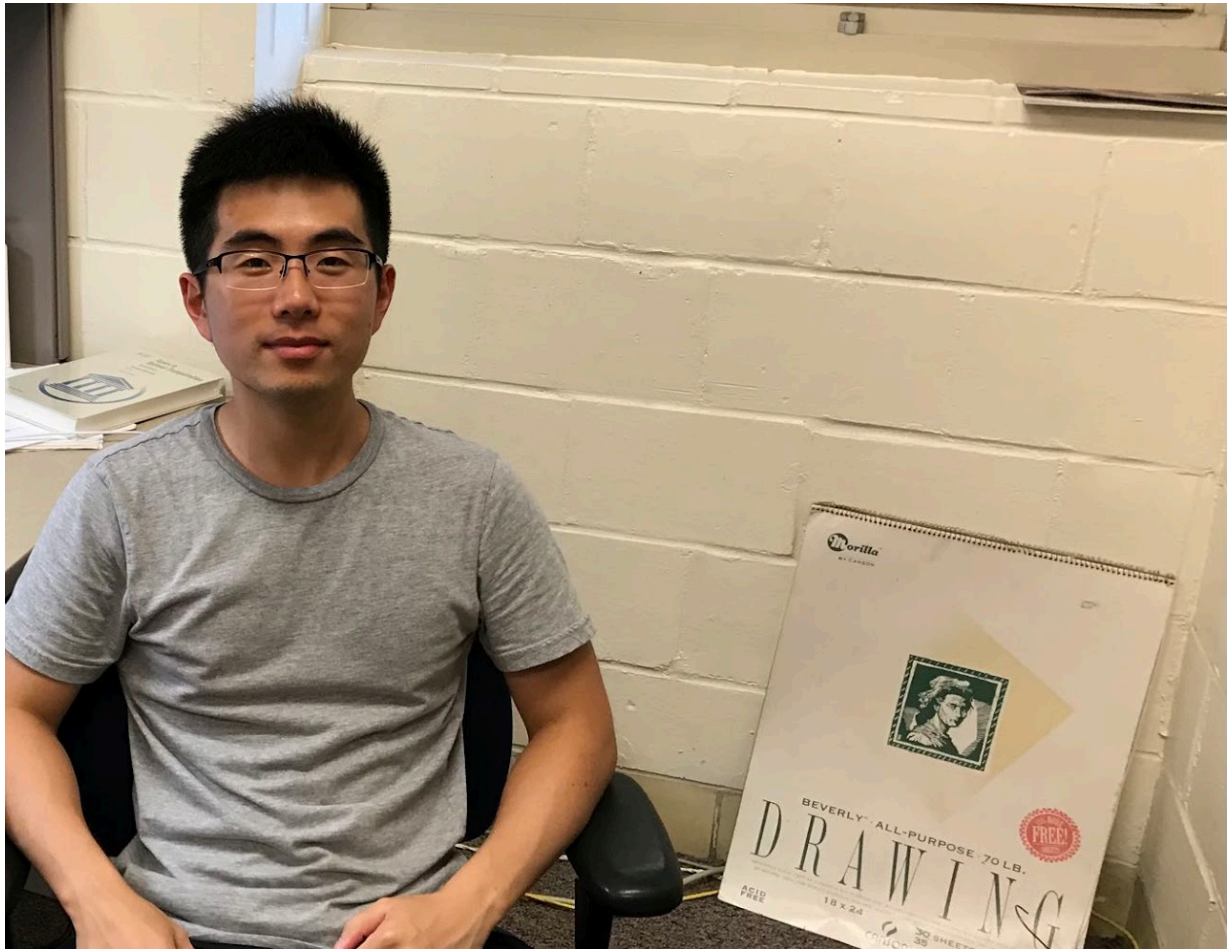
Generalization and Optimization in Symmetry-Preserving ML: Sample Complexity and Implicit Bias

Wei Zhu
University of Massachusetts Amherst

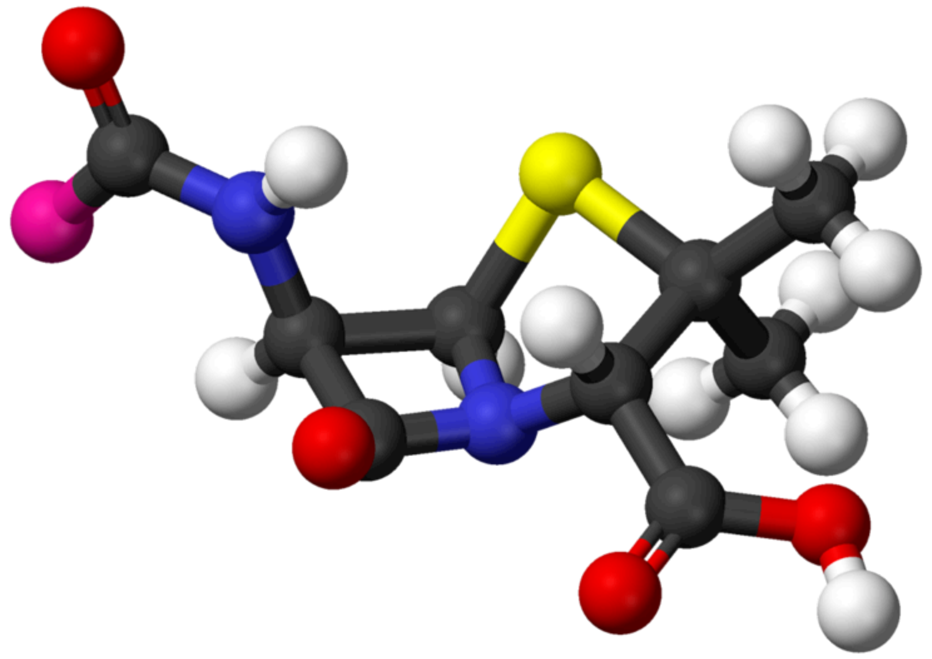
Boston Symmetry Day
MIT

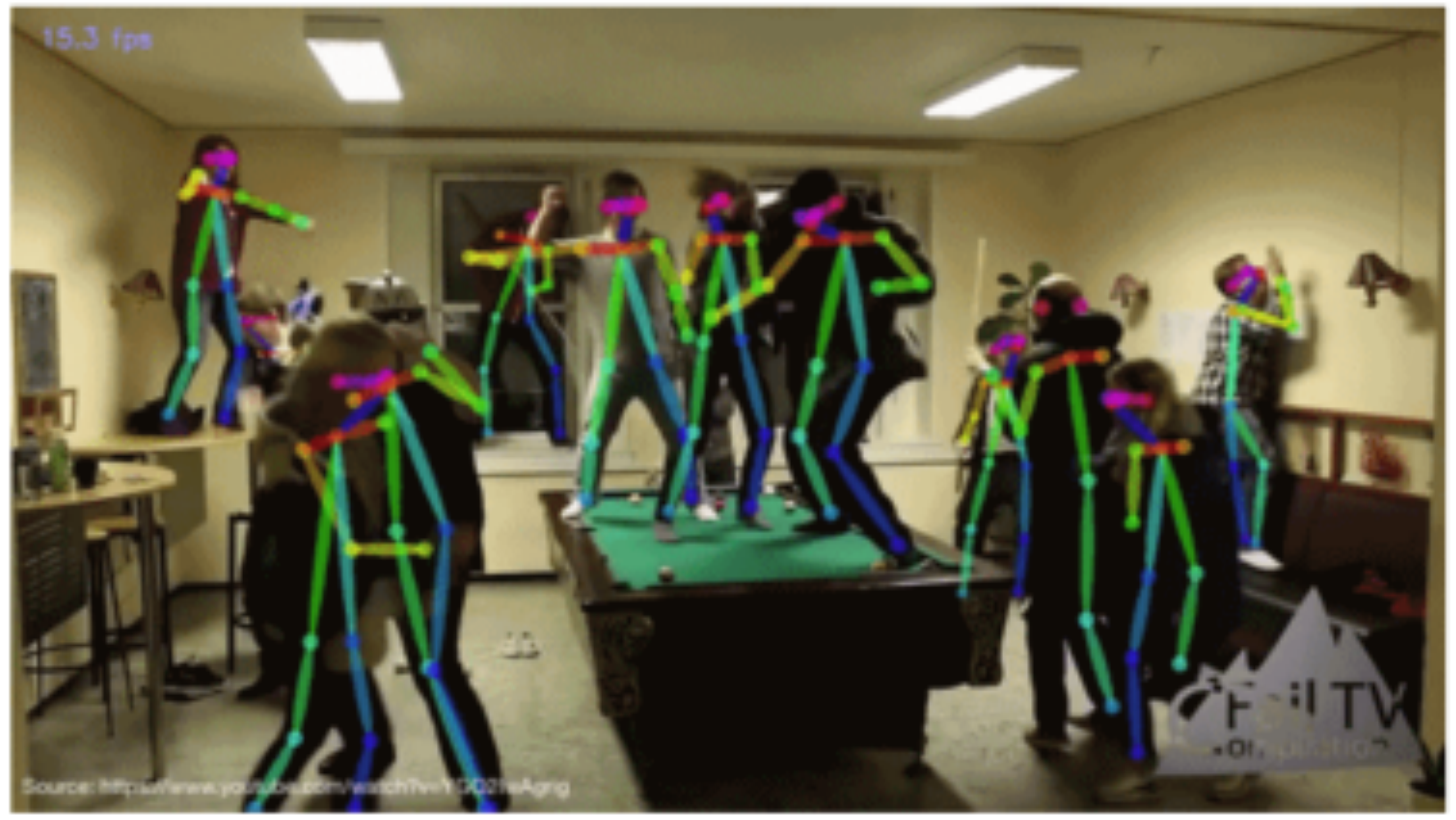
November 3, 2023

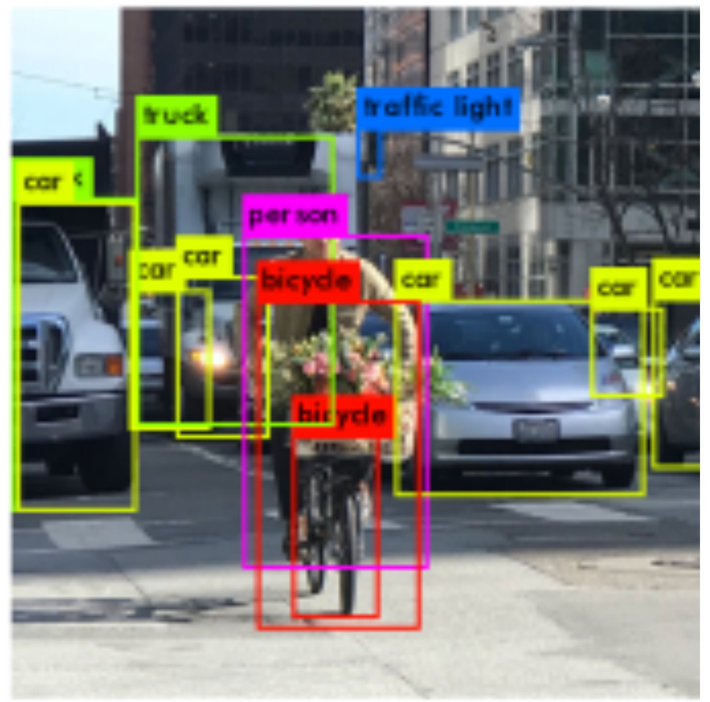
Joint work with many people, but mostly **Ziyu Chen** (UMass Amherst)

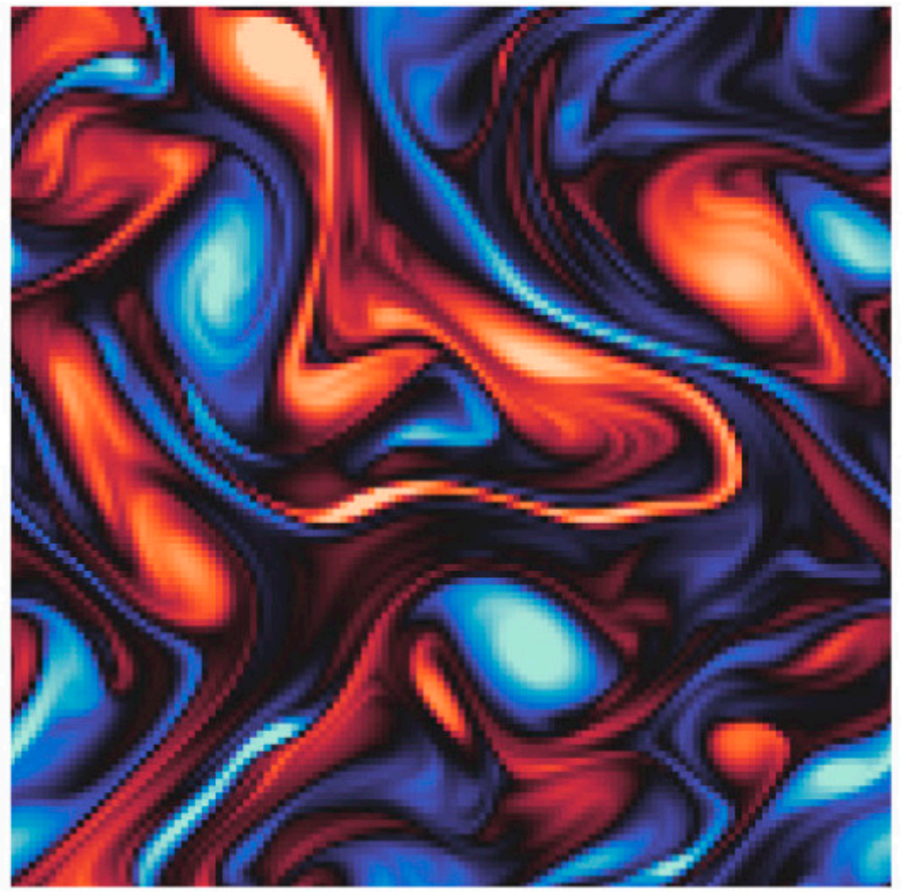


Symmetry is everywhere





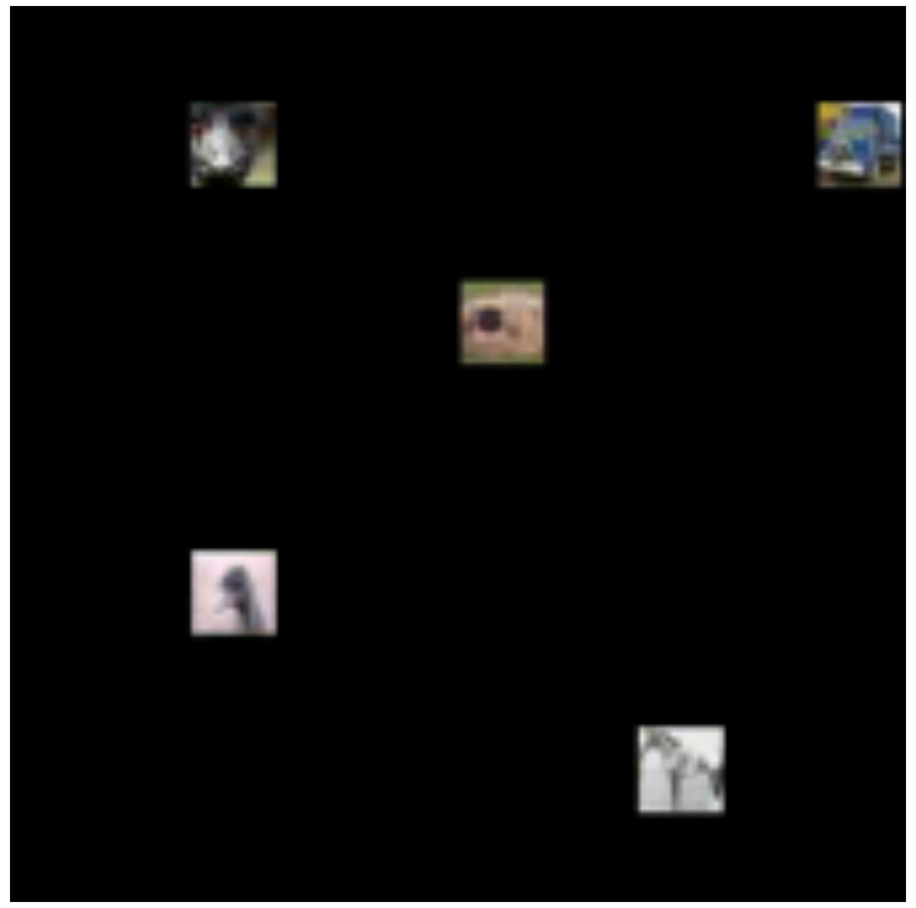




Missing pieces

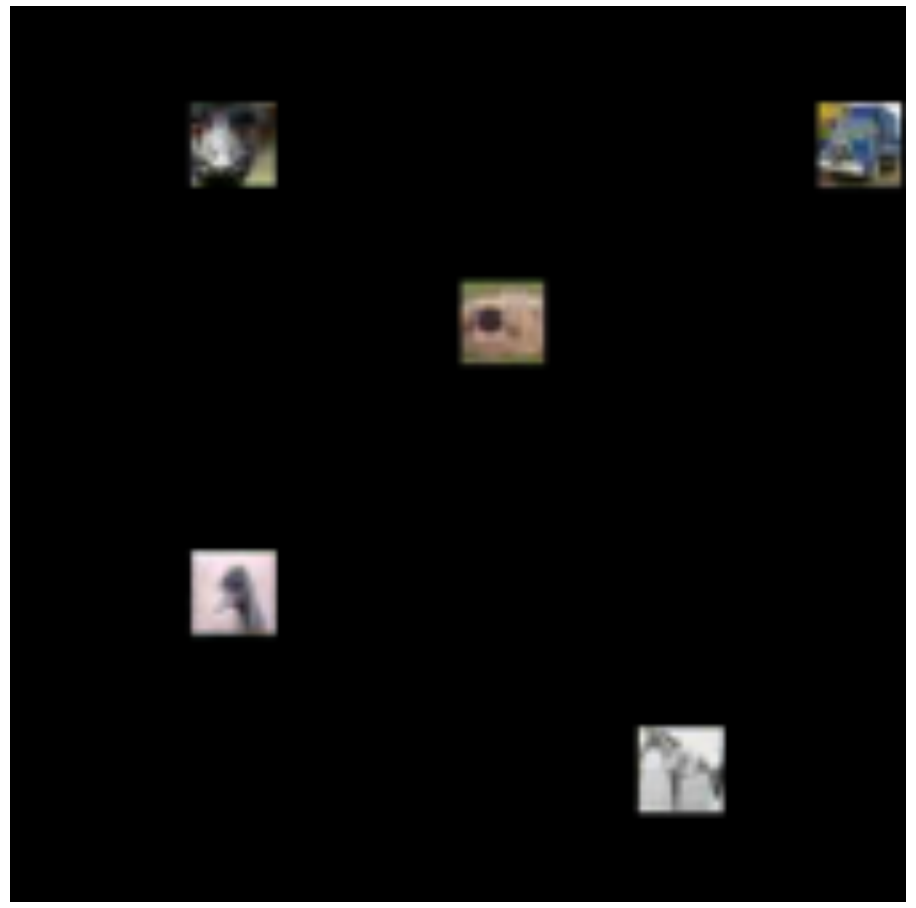
Missing pieces

- **Exact quantification of the improvement**
  - **Sample complexity** and **error bound**.



Missing pieces

- **Exact quantification of the improvement**
  - **Sample complexity** and **error bound**.

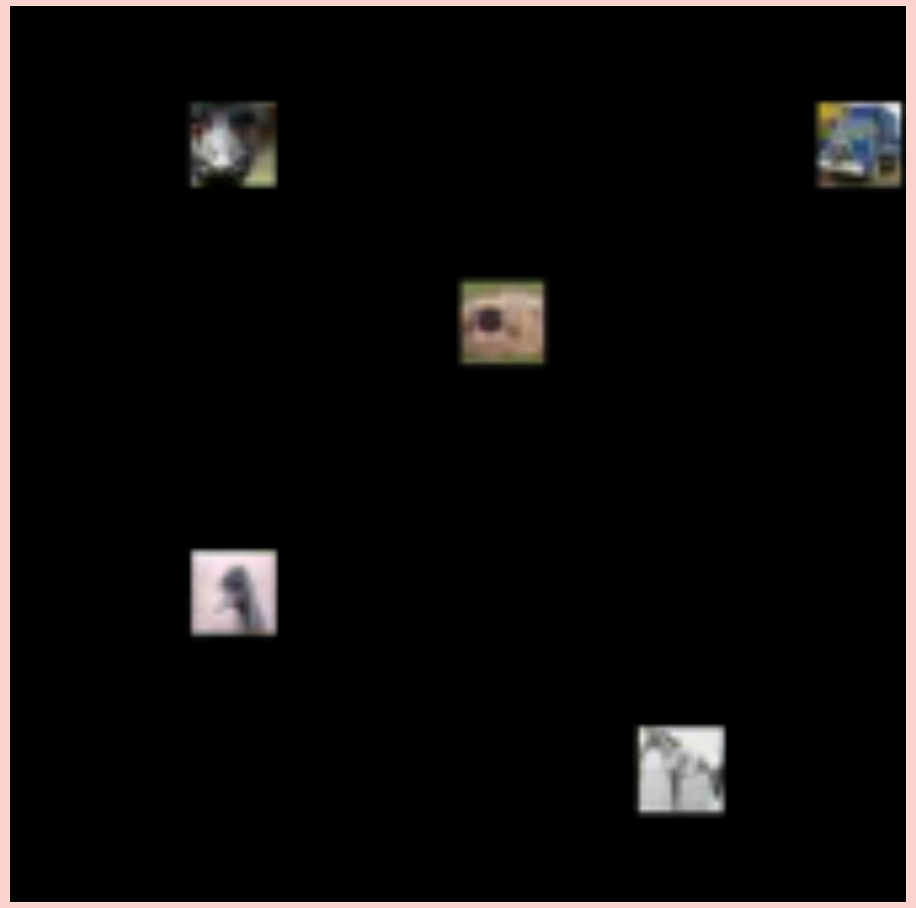


- **Does it converge? To what solution?**
  - **Training dynamics** of equivariant models

$$\boxed{\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathscr{L}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \mathbf{W}), y_i)}$$

Missing pieces

- **Exact quantification of the improvement**
  - **Sample complexity** and **error bound**.



- **Does it converge? To what solution?**
  - **Training dynamics** of equivariant models

$$\boxed{\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathscr{L}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \mathbf{W}), y_i)}$$

Symmetry-preserving GANs and their improved sample complexity

- J. Birrell, M.A. Katsoulakis, L. Rey-Bellet, **W. Zhu**. "Structure-preserving GANs". *ICML* (2022)

- Z. Chen, M.A. Katsoulakis, L. Rey-Bellet, **W. Zhu**. "Sample complexity of probability divergences under group symmetry". *ICML* (2023)

Generative adversarial networks (GANs)

StyleGAN2, Karras et al., CVPR 2020

StyleGAN3, Karras et al., NeurIPS 2021

DM-GAN, Zhu et al., CVPR 2019

Generative adversarial networks (GANs)

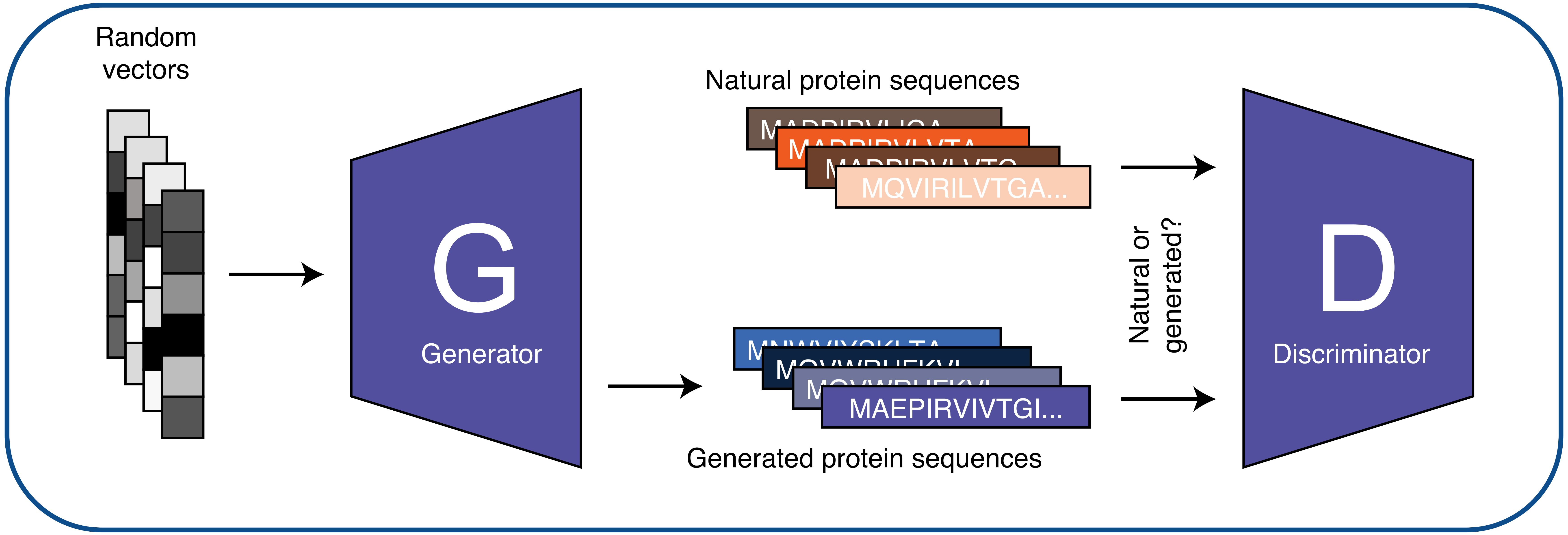


Figure: Repecka et al., *Nature Machine Intelligence* 2021

Generative adversarial networks (GANs)

- GANs use a pair of networks to learn (to sample from) an **unknown** probability distribution.

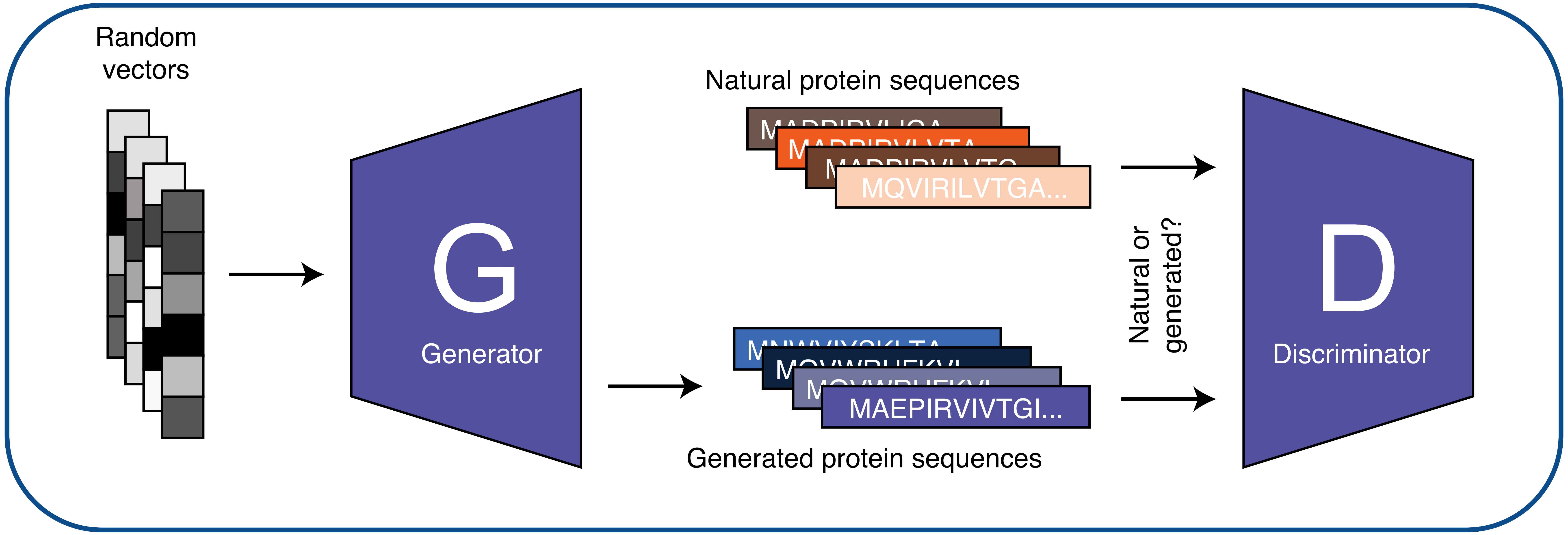


Figure: Repecka et al., *Nature Machine Intelligence* 2021

Generative adversarial networks (GANs)

- GANs use a pair of networks to learn (to sample from) an **unknown** probability distribution.
- **Zero-sum game** between discriminator and generator—**"the players".**

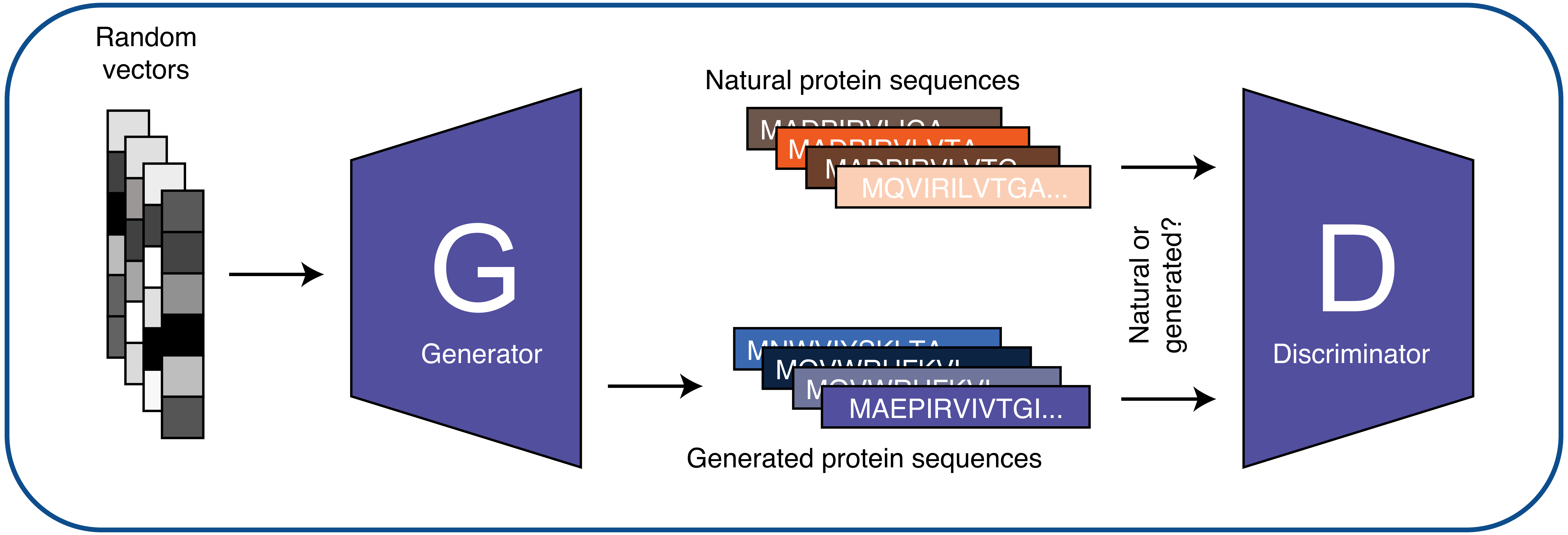


Figure: Repecka et al., *Nature Machine Intelligence* 2021

Generative adversarial networks (GANs)

- GANs use a pair of networks to learn (to sample from) an **unknown** probability distribution.
- **Zero-sum game** between discriminator and generator—**"the players".**
- Game ends when the players reach *consensus*: "fake data" looks like the "real" data.

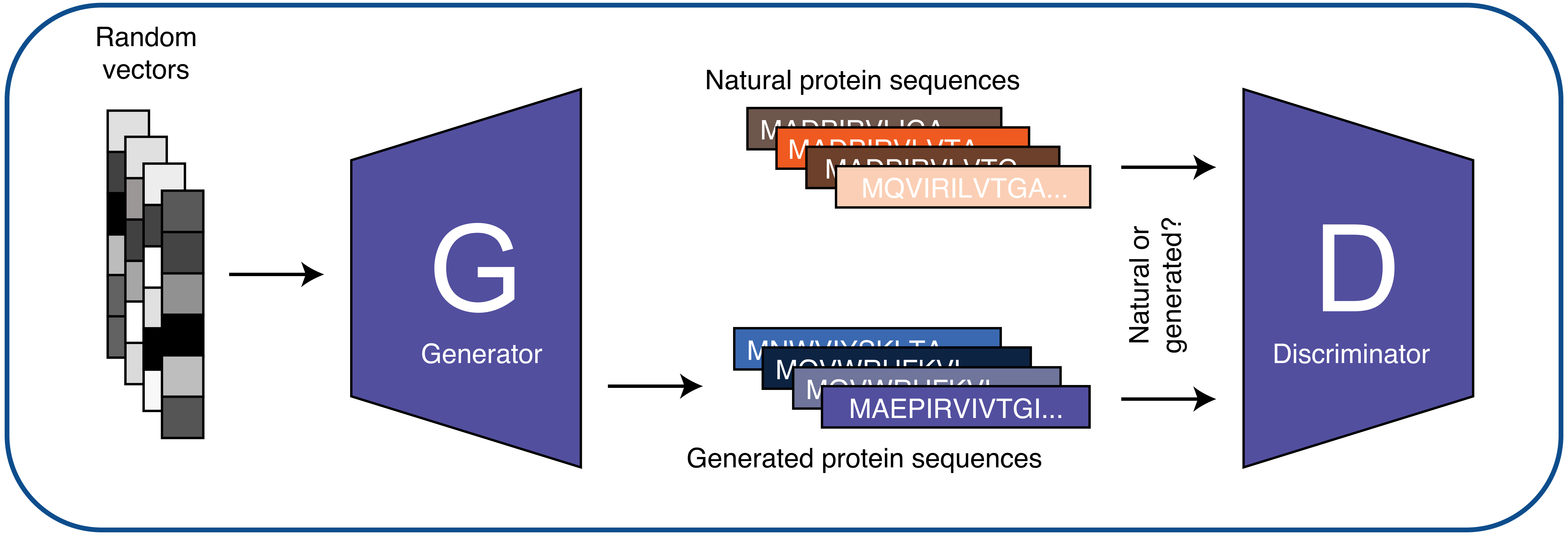
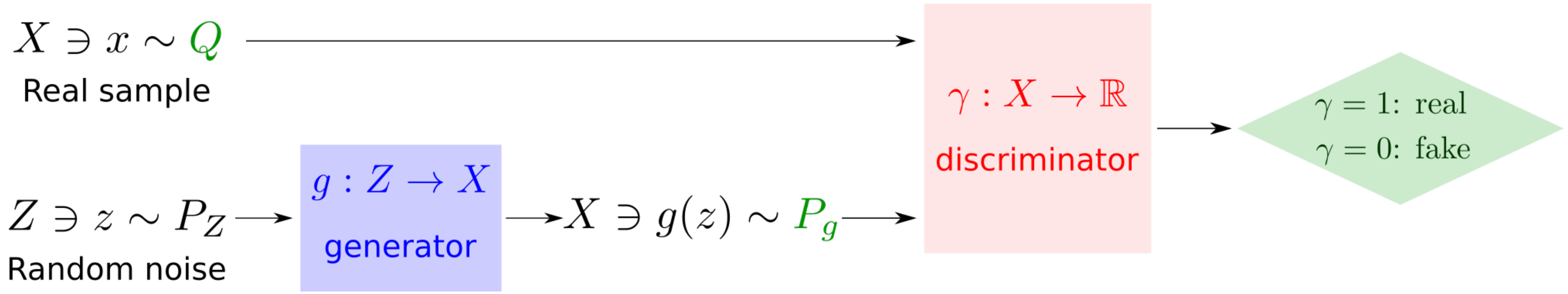
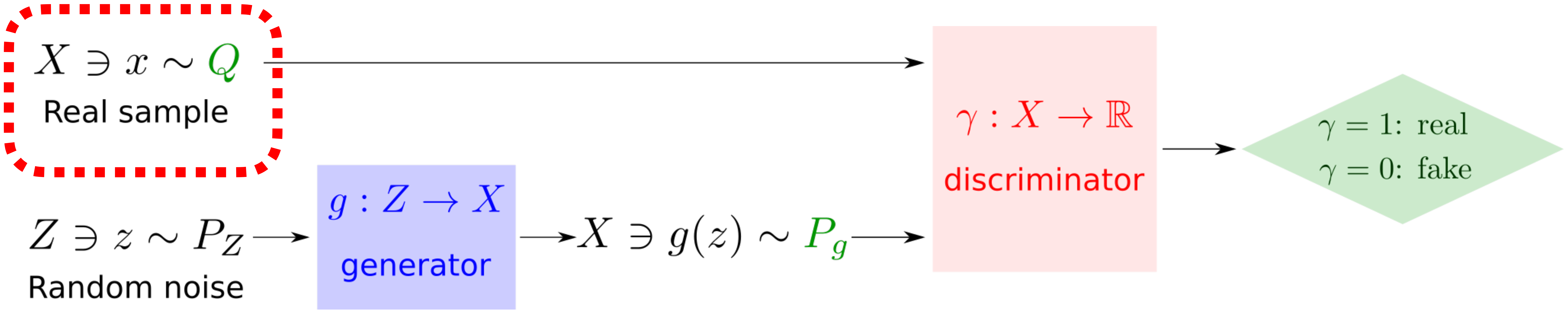


Figure: Repecka et al., *Nature Machine Intelligence* 2021

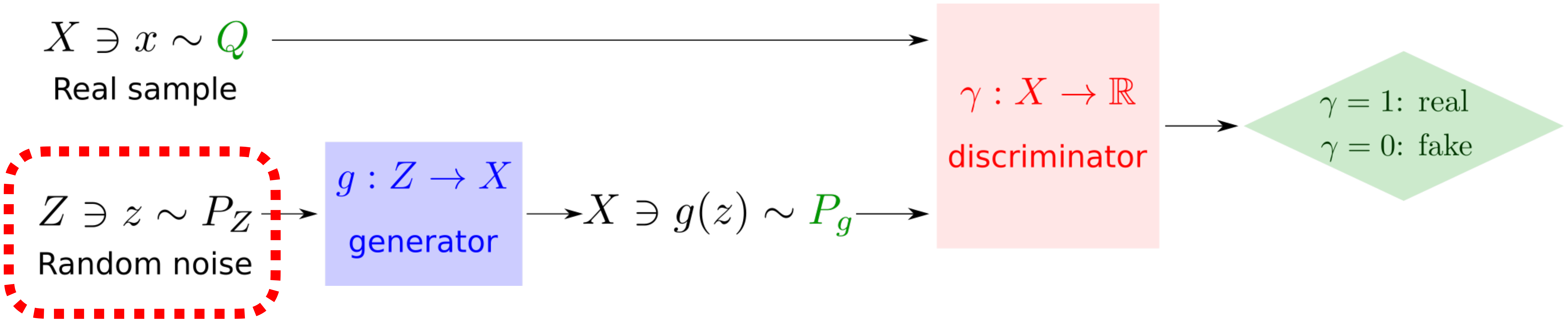
Generative adversarial networks (GANs)



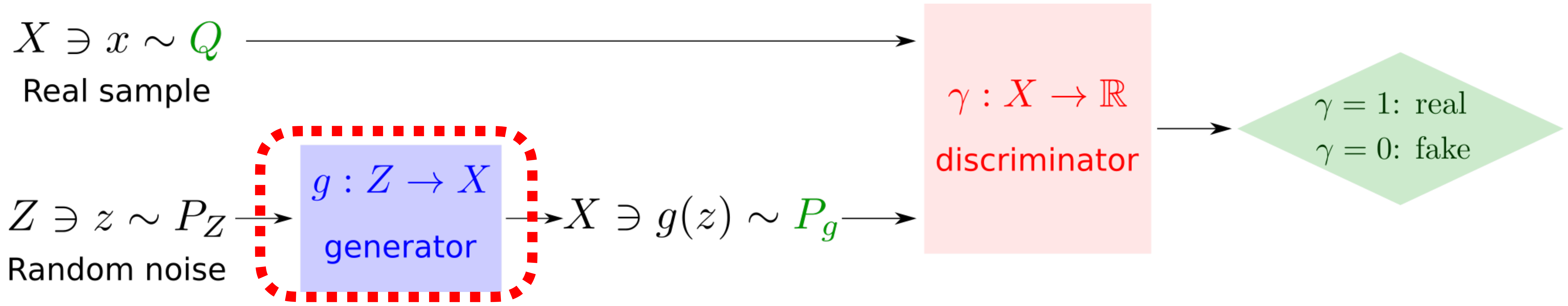
Generative adversarial networks (GANs)



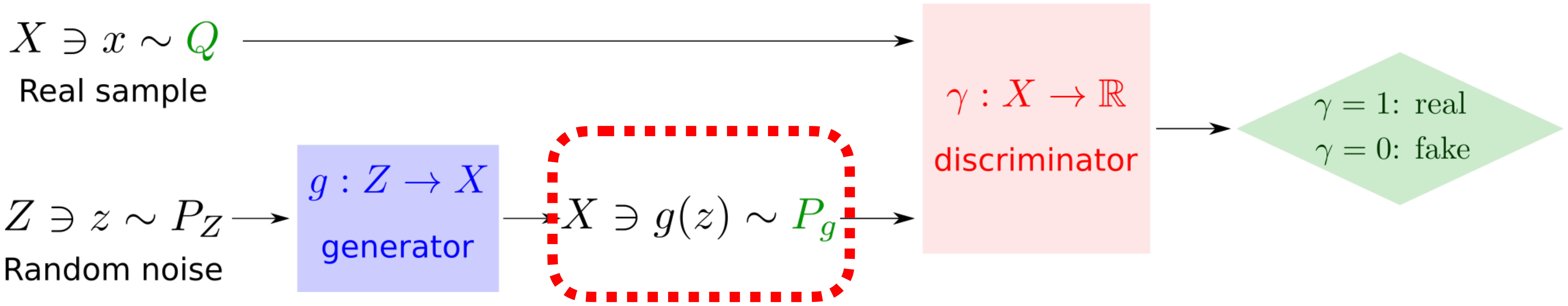
Generative adversarial networks (GANs)



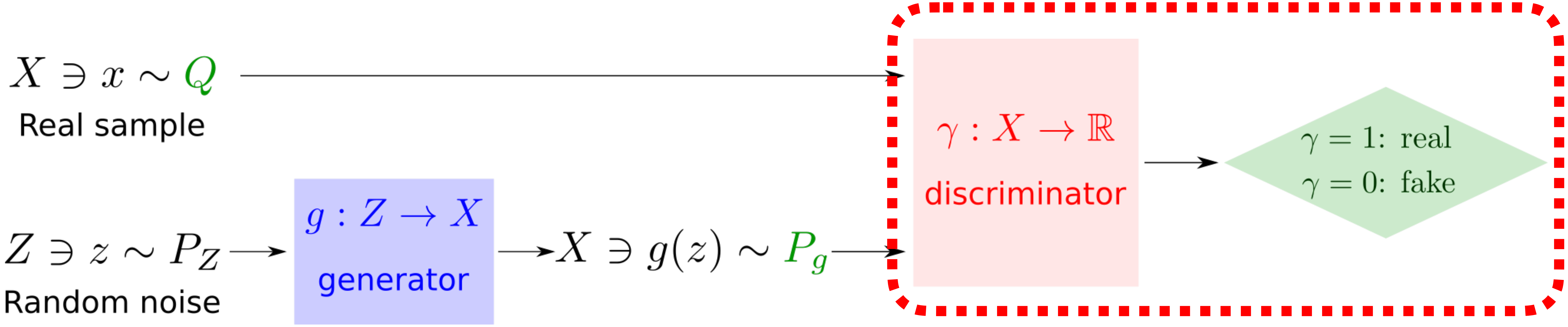
Generative adversarial networks (GANs)



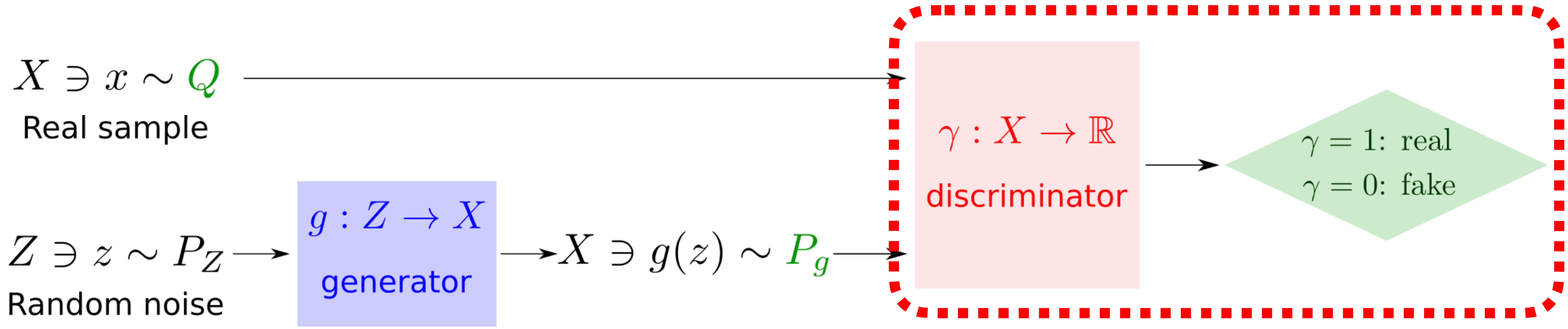
Generative adversarial networks (GANs)



Generative adversarial networks (GANs)



Generative adversarial networks (GANs)



- Mathematically, GAN is minimizing some **divergence**, $D_H^\Gamma(Q \| P_g)$, between Q and P_g.
- $D_H^\Gamma(Q \| P_g) = \max_{\gamma \in \Gamma} H(\gamma; Q, P_g)$ is determined by H and discriminators $\gamma \in \Gamma$.

$$\min_{g \in G} D_H^\Gamma(Q \| P_g) = \min_{g \in G} \max_{\gamma \in \Gamma} H(\gamma; Q, P_g).$$

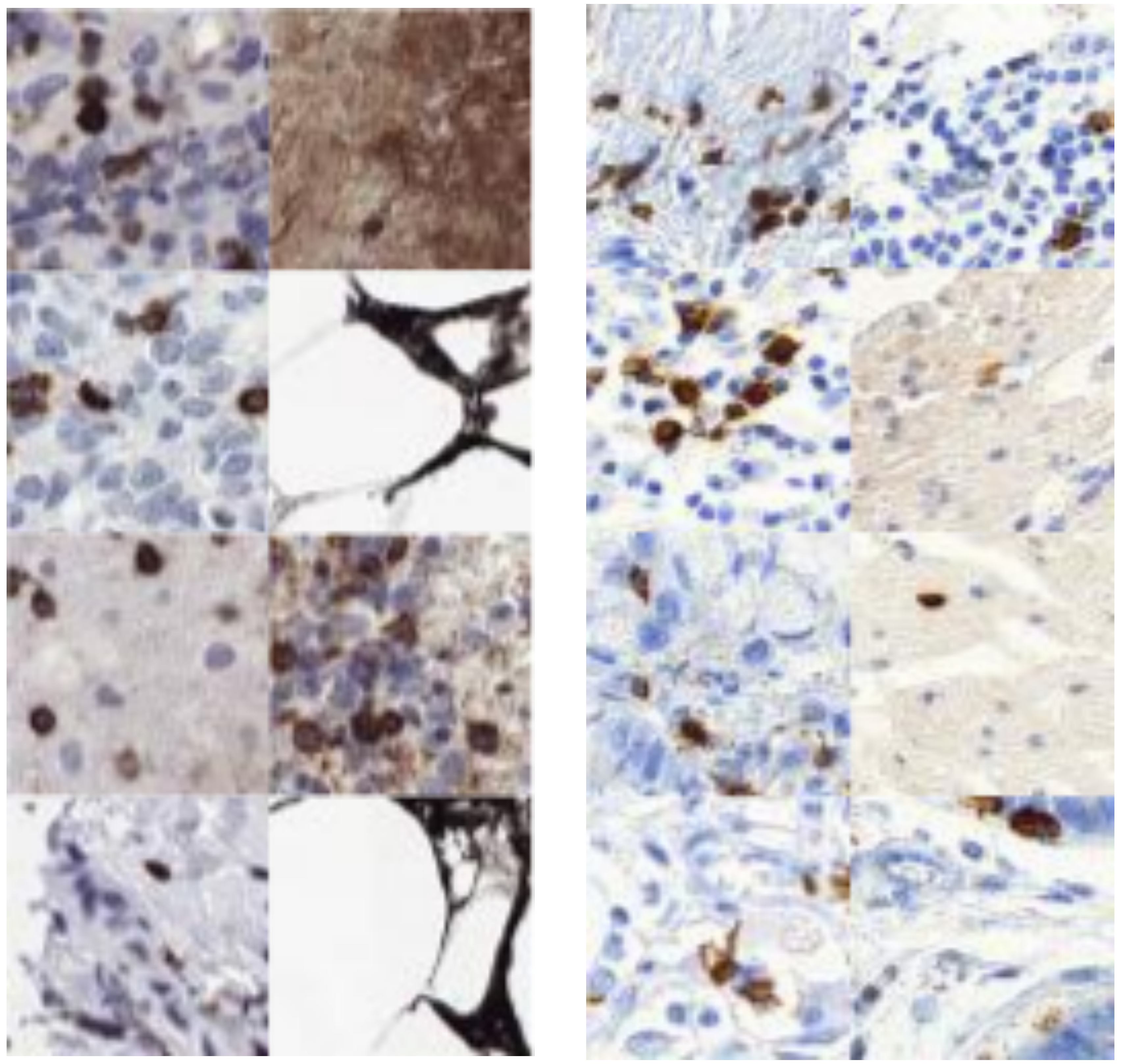
GAN is "probability divergence" minimization

$$\min_{g\in G} D_H^\Gamma(Q\|P_g) = \min_{g\in G}\max_{\gamma\in\Gamma} H(\gamma; Q, P_g).$$

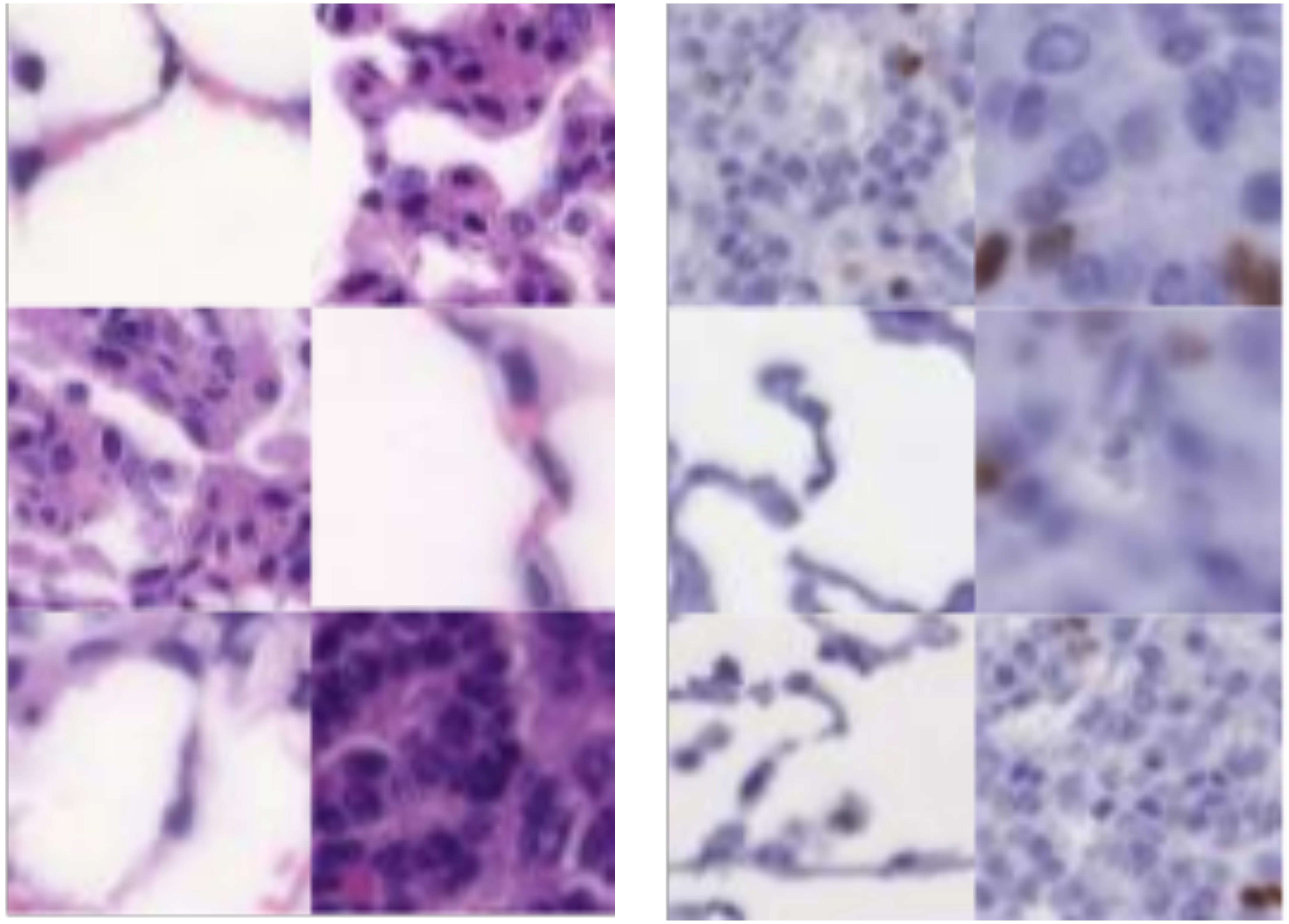
- The original GAN [Goodfellow et al., 2014]: Jensen–Shannon divergence (JSD).
- f-divergences: $D_f(Q\|P) = \sup_{\gamma\in\mathcal{M}_b(X)} \{\mathbb{E}_Q[\gamma] - \mathbb{E}_P[f^*(\gamma)]\}$. (KL, JSD, etc.)
- Γ-IPM: $W^\Gamma(Q\|P) = \sup_{\gamma\in\Gamma}\{\mathbb{E}_Q[\gamma] - \mathbb{E}_P[\gamma]\}$. (TV, Dudley metric, Wasserstein-1, MMD)
- Wasserstein metric and Sinkhorn divergence.

Structured target data & distribution Q

Q



LYSTO[1]



ANHIR[2]

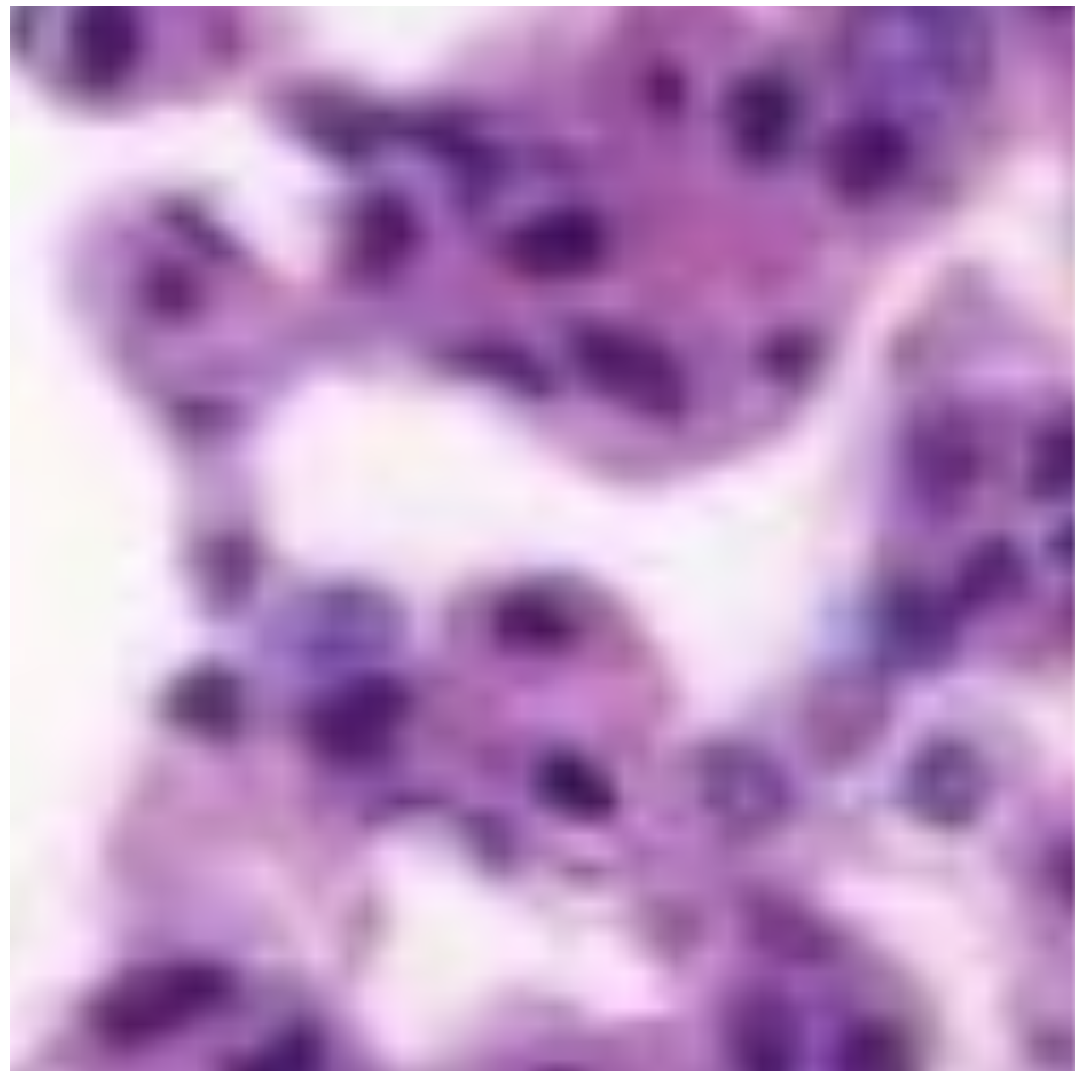
1. Ciompi et al., Zenodo 2019

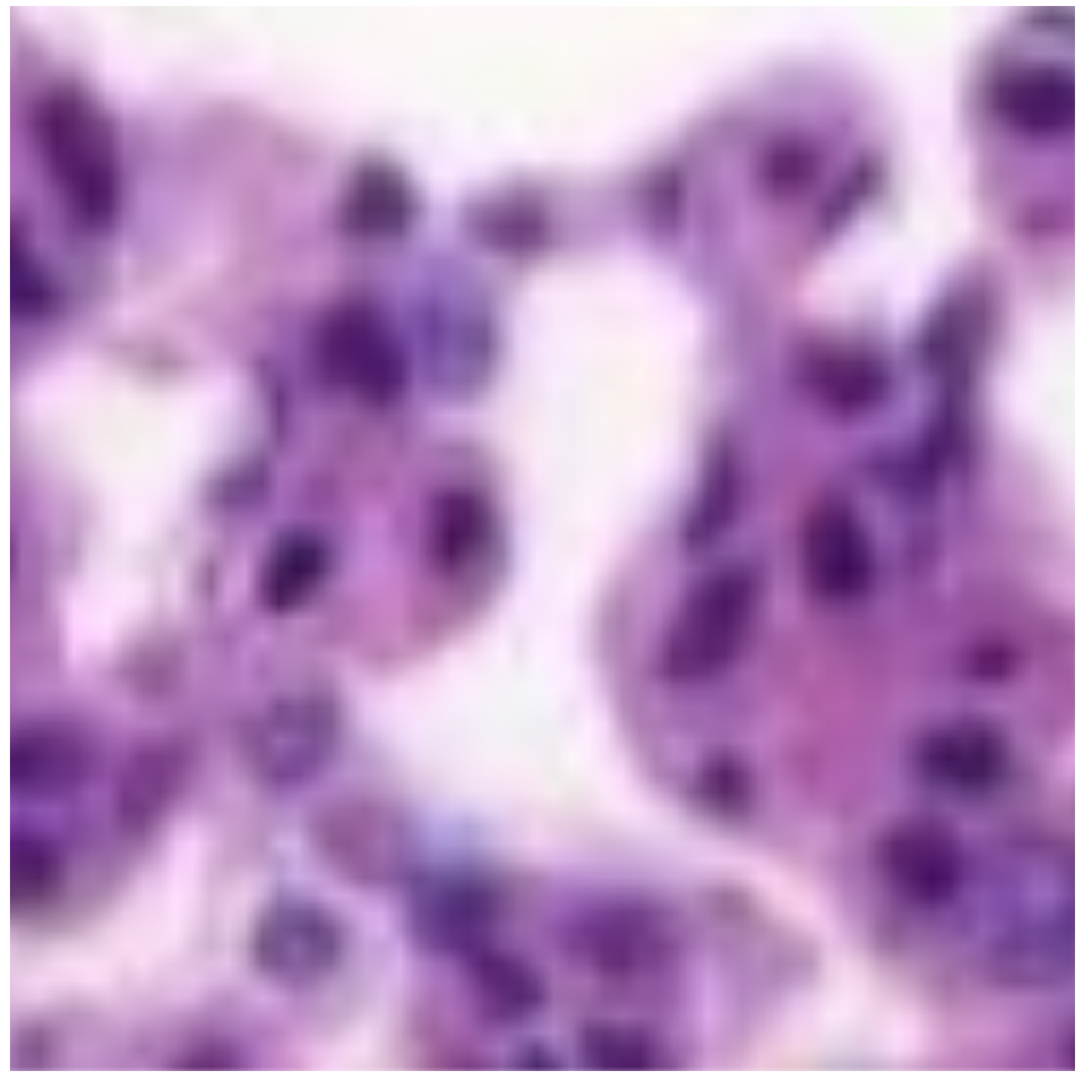
2. Borovec et al., IEEE Transactions on Medical Imaging 2020

Structured target data & distribution Q

equiprobable

Q

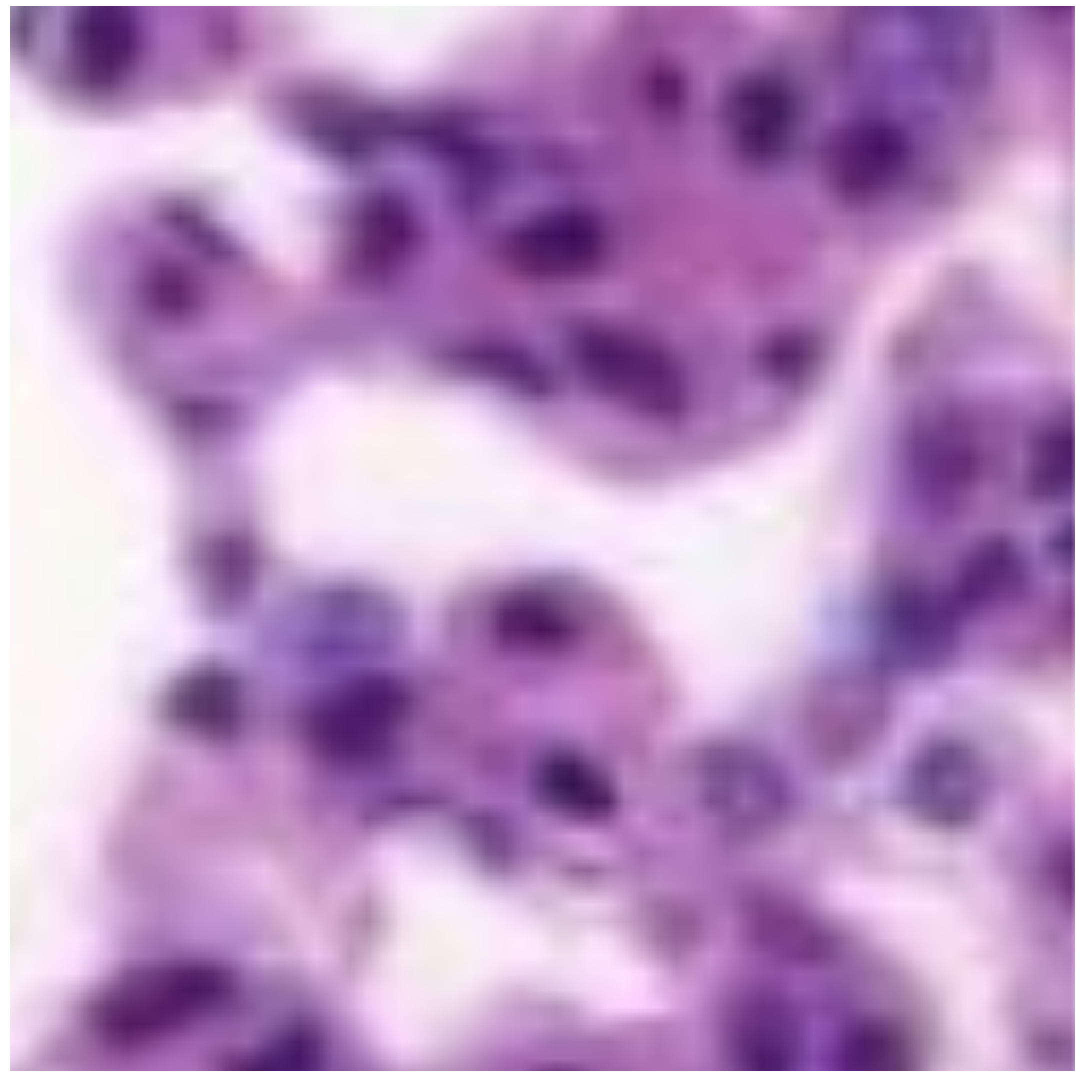


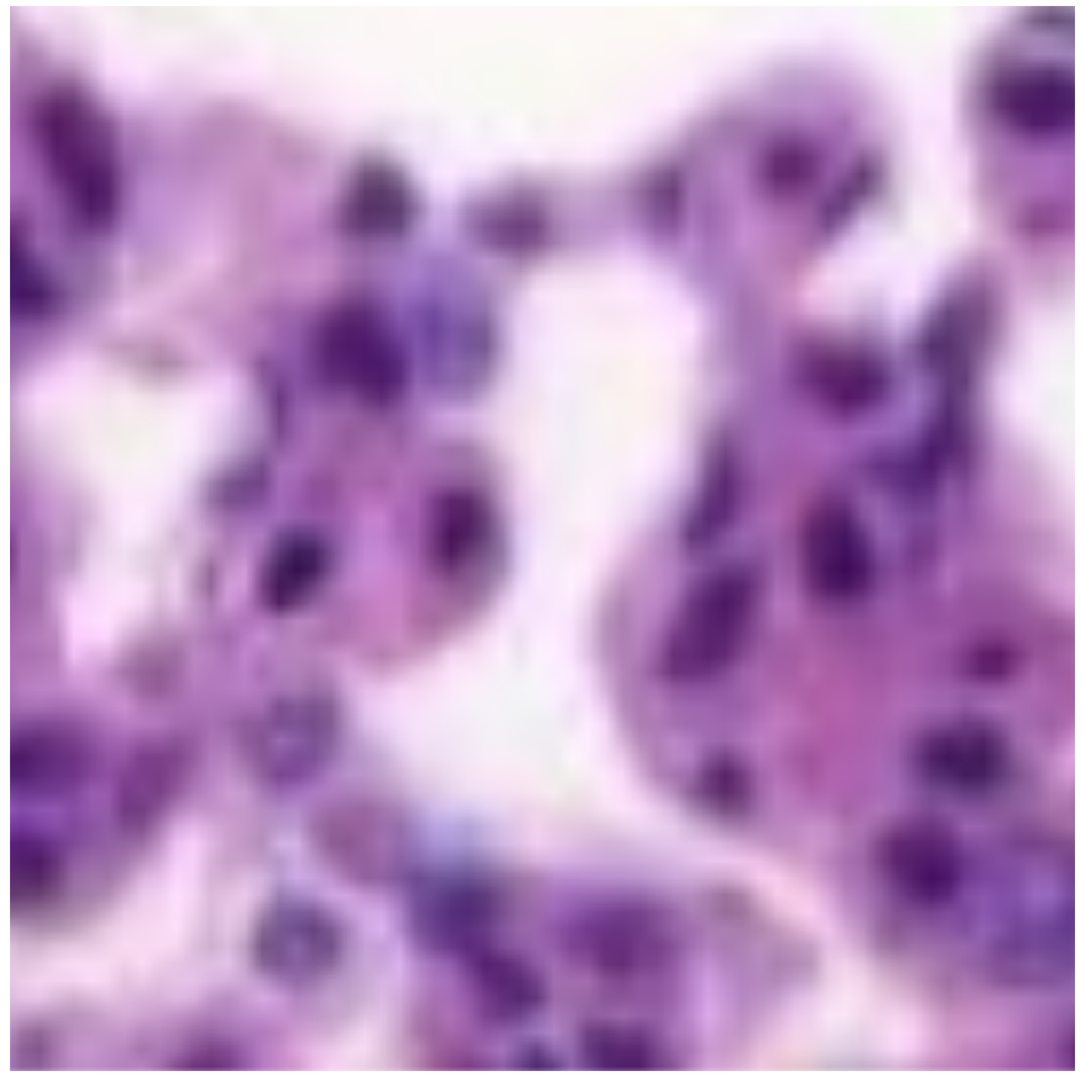


Structured target data & distribution Q

equiprobable

Q





Question: how to build **embedded structure** into GAN players (generators and discriminators) for data-efficient distribution learning?

GAN with embedded structure

$X \ni x \sim Q$ $\longrightarrow$

Real sample

$\gamma : X \to \mathbb{R}$

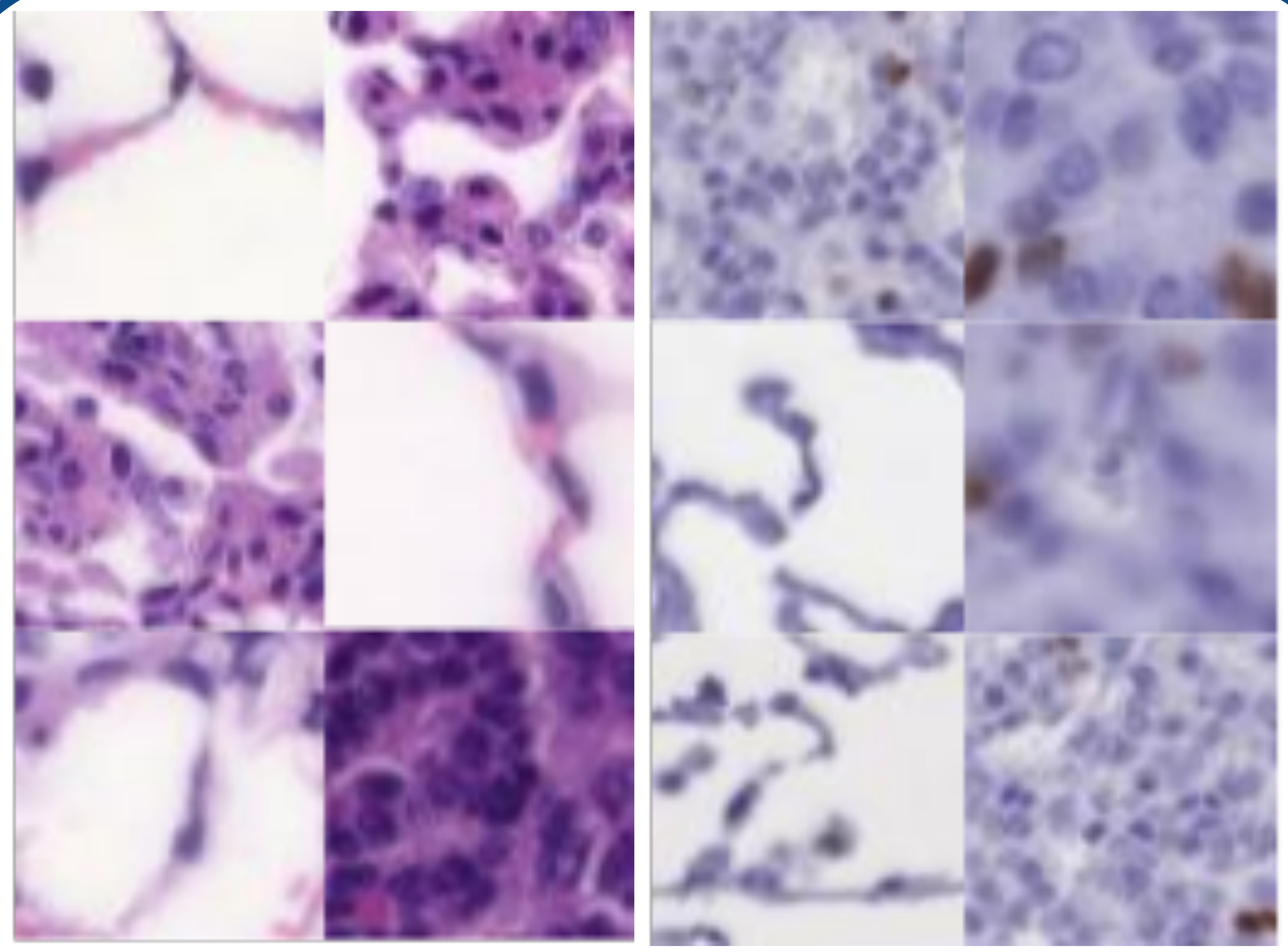
discriminator

$\longrightarrow$ $\gamma = 1$: real, $\gamma = 0$: fake

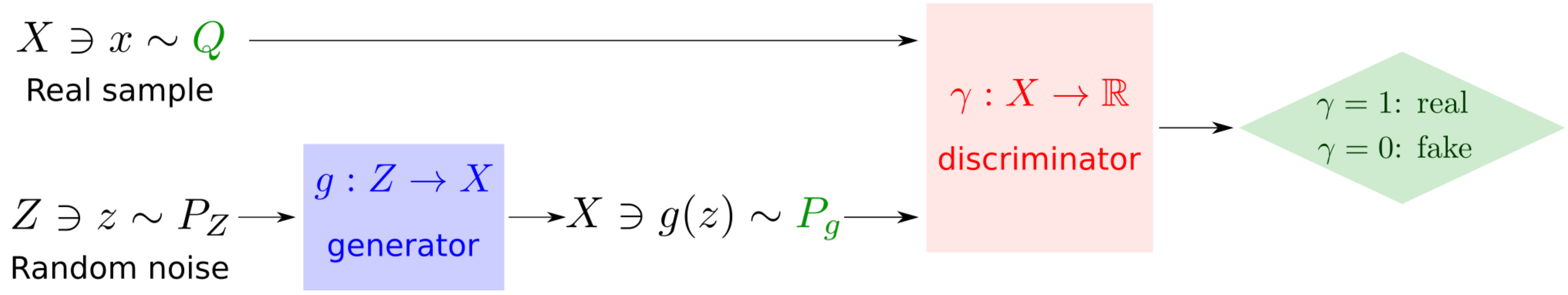
$Z \ni z \sim P_Z \longrightarrow$ $g : Z \to X$ generator $\longrightarrow X \ni g(z) \sim P_g \longrightarrow$

Random noise

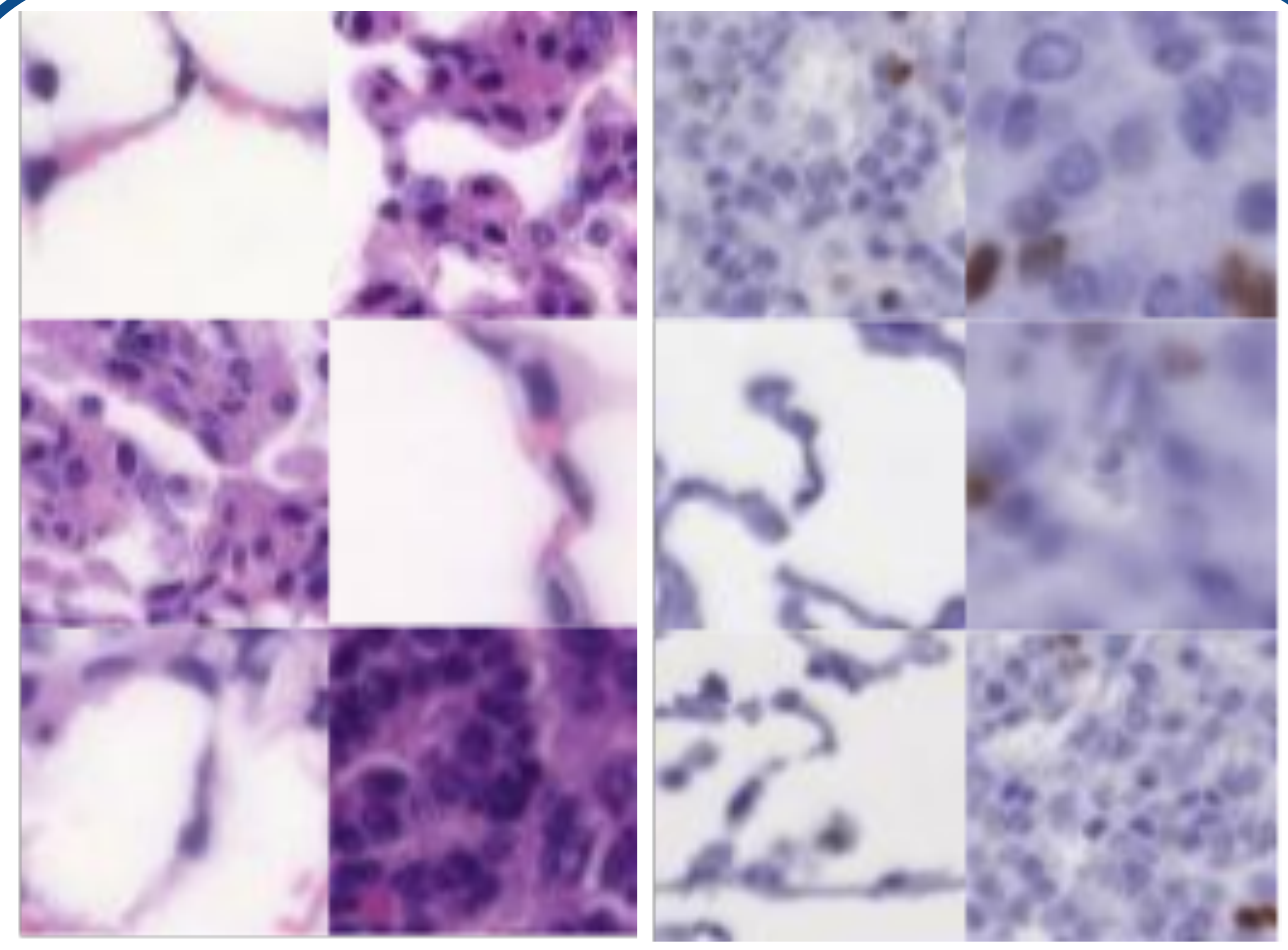
$$\min_{g \in G} D^{\Gamma}(Q \| P_g) = \min_{g \in G} \max_{\gamma \in \Gamma} H(\gamma; Q, P_g), \quad \underline{Q \text{ is } \Sigma\text{-invariant}}$$



GAN with embedded structure

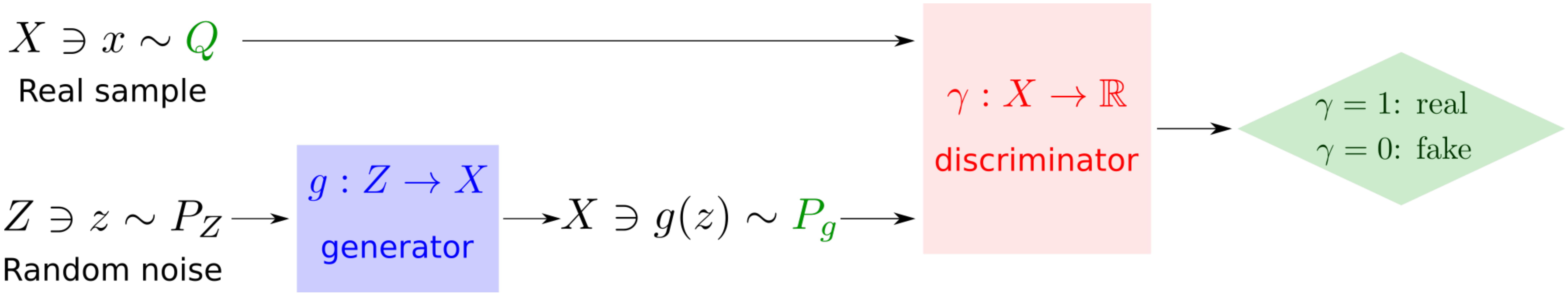


$$\min_{g \in G} D^{\Gamma}(Q \| P_g) = \min_{g \in G} \max_{\gamma \in \Gamma} H(\gamma; Q, P_g), \quad Q \text{ is } \Sigma\text{-invariant}$$

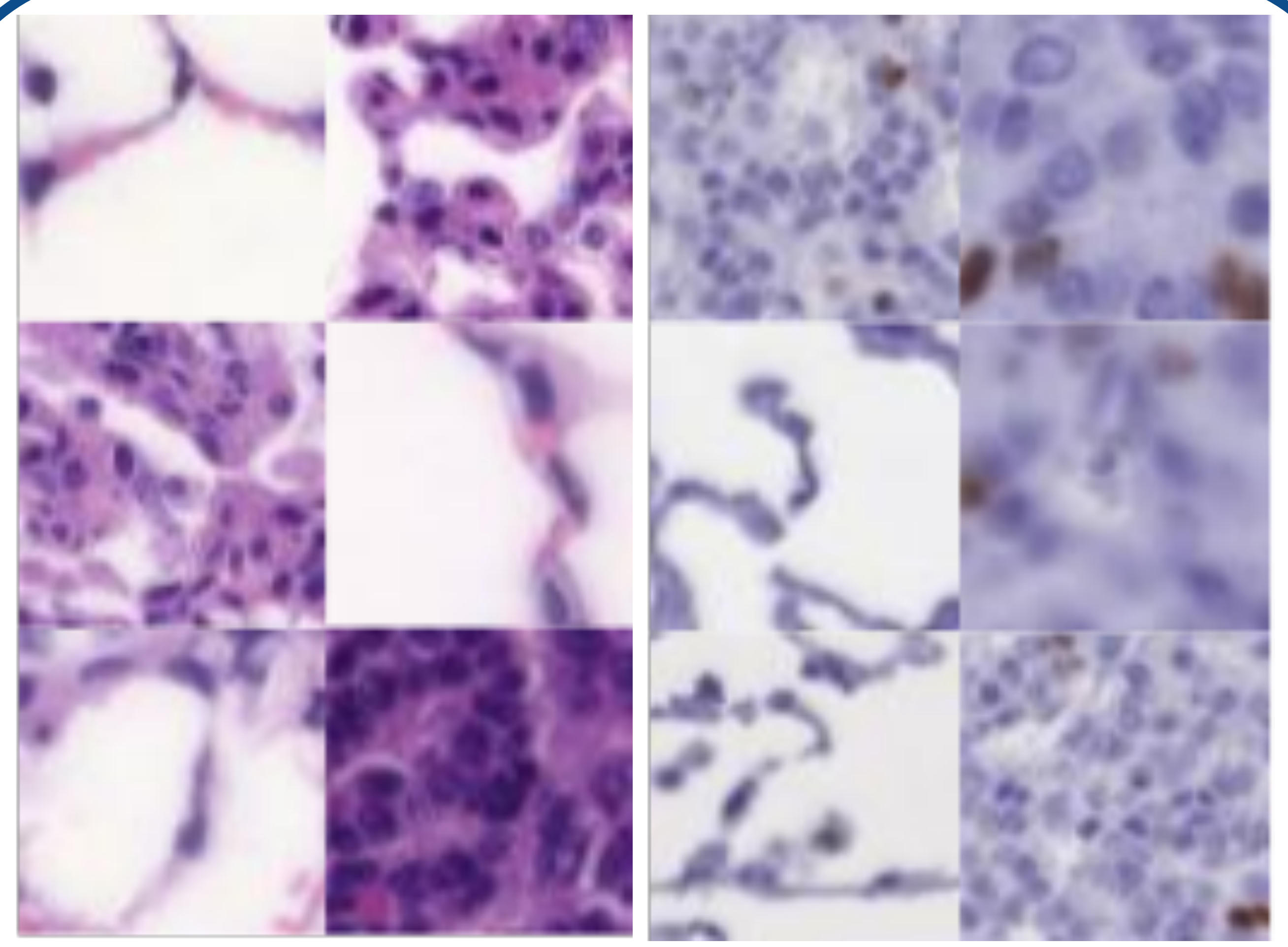


- Target distribution Q is invariant under a group Σ.

GAN with embedded structure

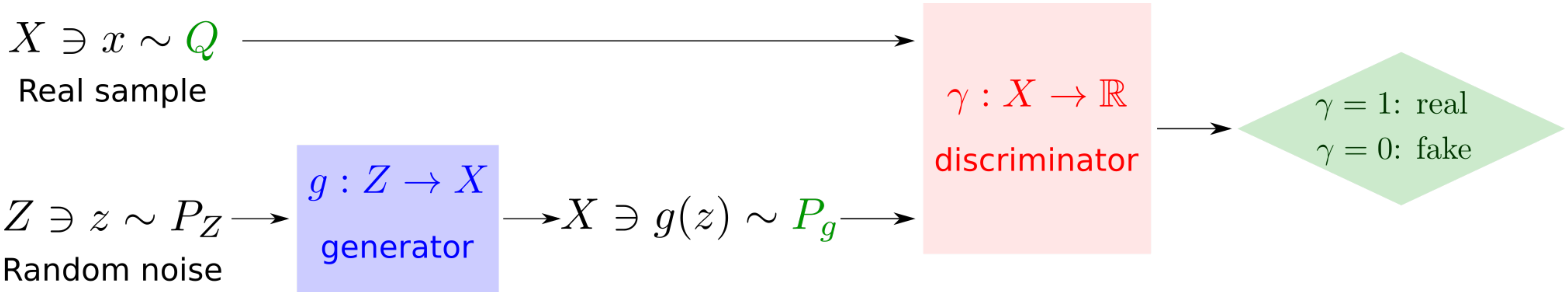


$$\min_{g \in G} D^{\Gamma}(Q \| P_g) = \min_{g \in G} \max_{\gamma \in \Gamma} H(\gamma; Q, P_g), \quad \underline{Q \text{ is } \Sigma\text{-invariant}}$$



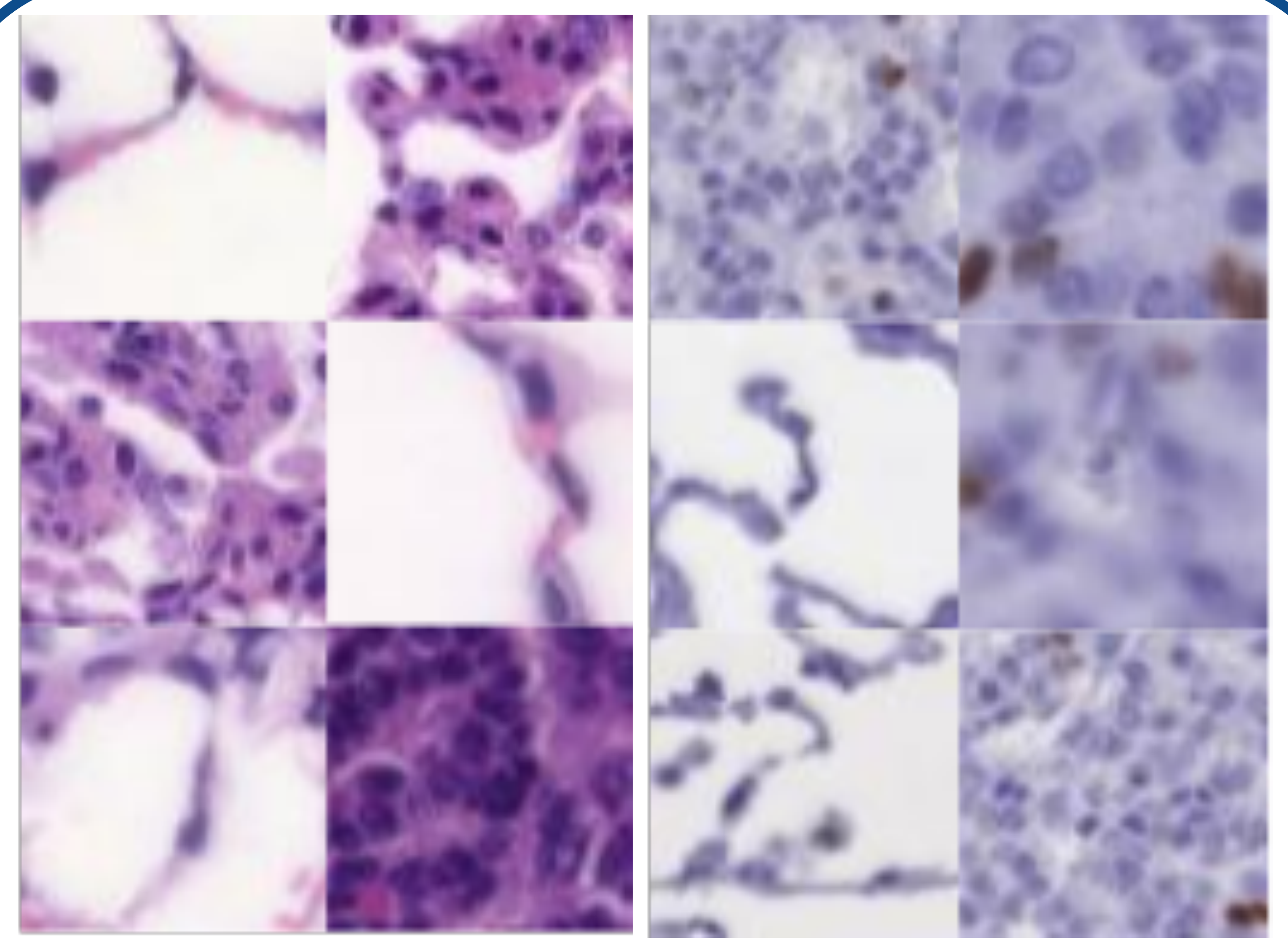
- Target distribution Q is invariant under a group Σ.
- Σ: rotation, reflection, permutation, etc.

GAN with embedded structure



$$\min_{g \in G} D^{\Gamma}(Q \| P_g) = \min_{g \in G} \max_{\gamma \in \Gamma} H(\gamma; Q, P_g),$$

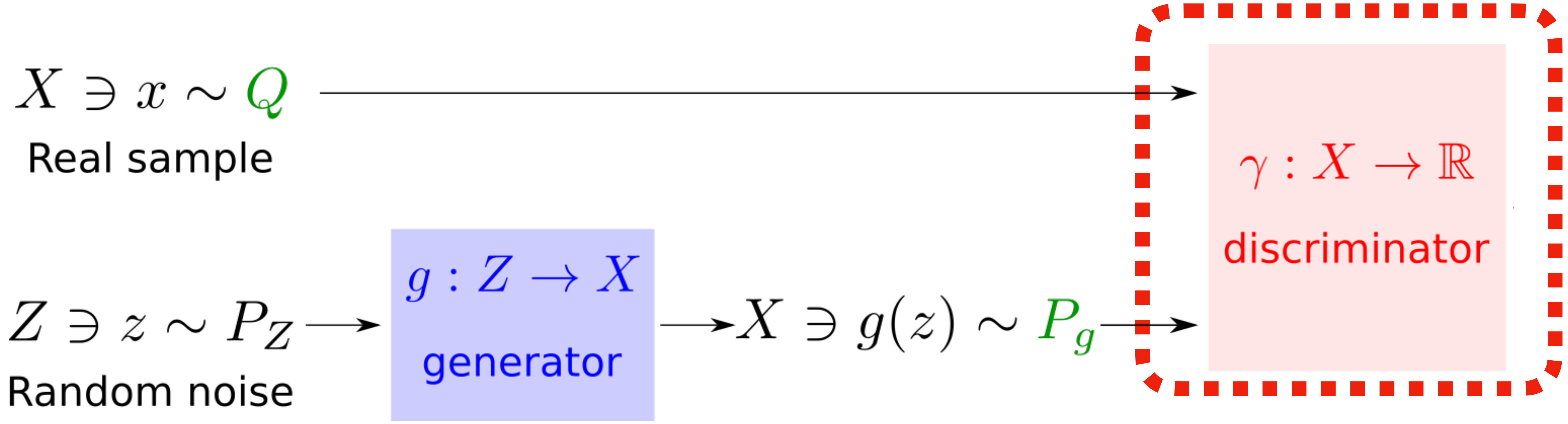
Q is Σ-invariant



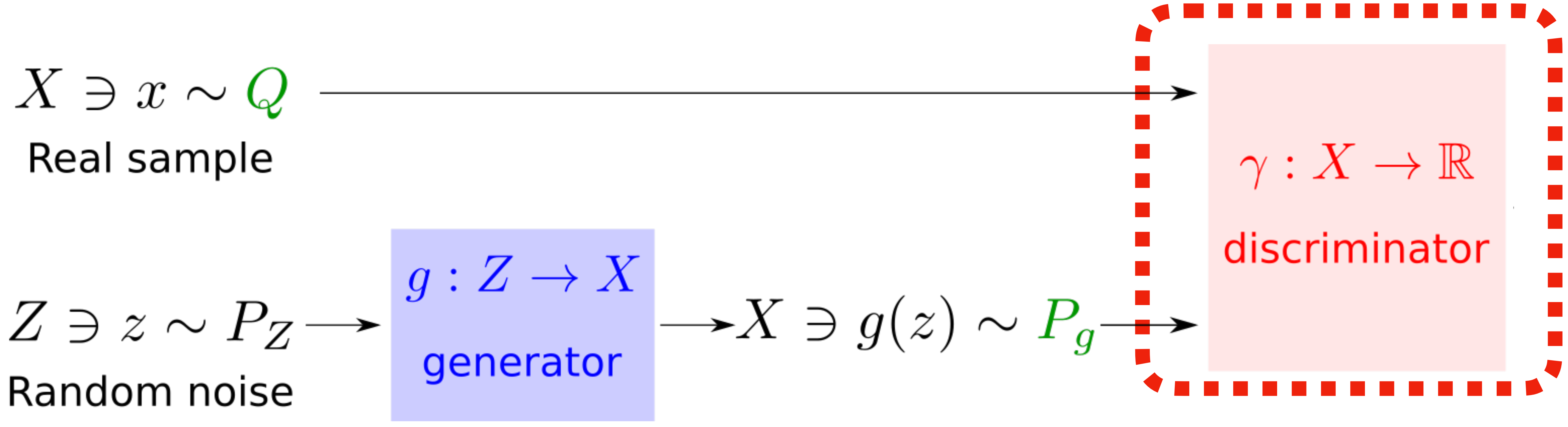
- Target distribution Q is invariant under a group Σ.
- Σ: rotation, reflection, permutation, etc.
- **How to incorporate structure into g and γ?**

Theorem 1: “smarter” discriminator

Theorem 1: “smarter” discriminator



Theorem 1: “smarter” discriminator



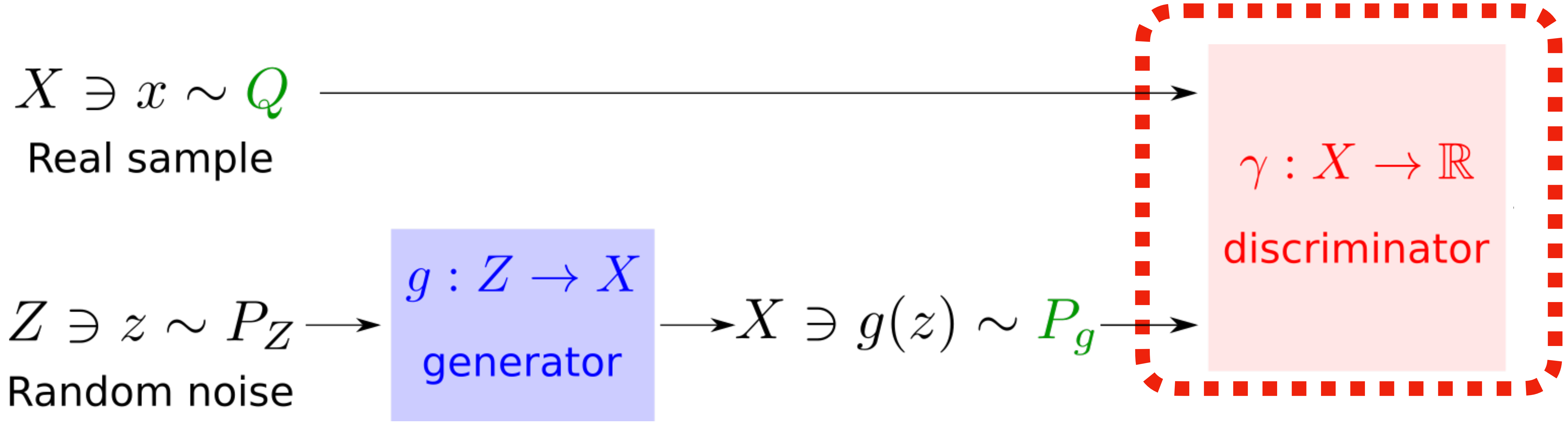
Theorem [Birrell, Katsoulakis, Rey-Bellet, **Z.**, *ICML* 2022]

Under mild assumptions on Σ and Γ, if the distributions P, Q are **Σ-invariant**, then

$$D^{\Gamma}(Q\|P) = D^{\Gamma_\Sigma^{\text{inv}}}(Q\|P) \ = \sup_{\gamma \in \Gamma_\Sigma^{\text{inv}}} H(\gamma; Q, P),$$

- $\Gamma_\Sigma^{\text{inv}} \subset \Gamma$ is the subset of **Σ-invariant** “smarter” discriminators

Theorem 1: “smarter” discriminator



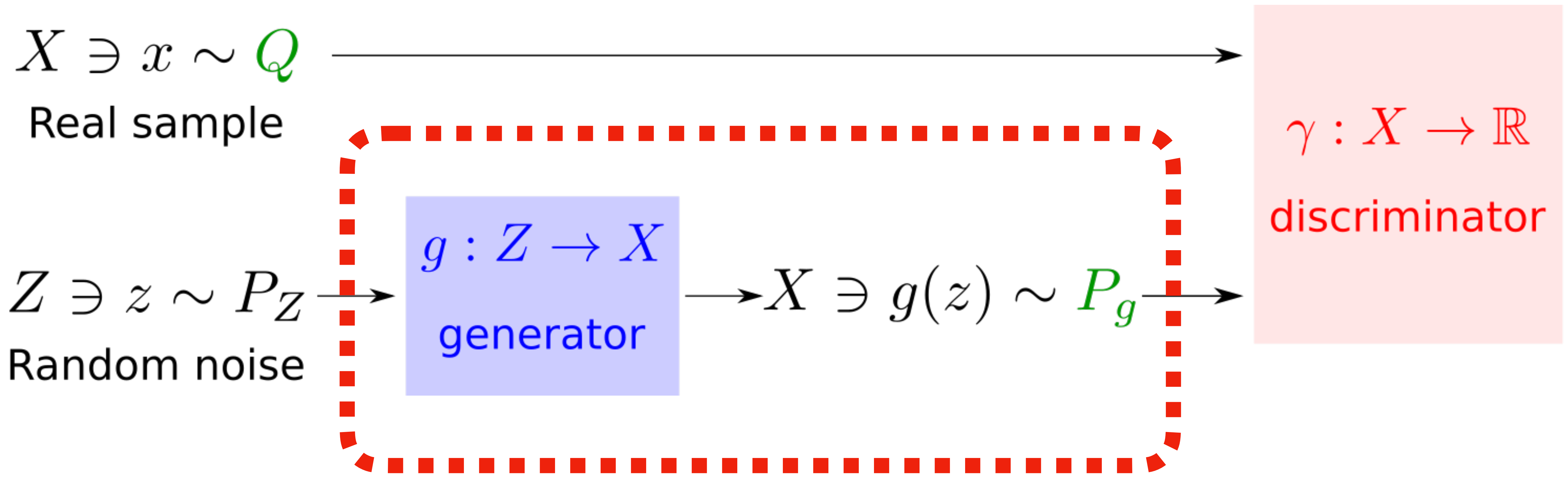
Theorem [Birrell, Katsoulakis, Rey-Bellet, **Z.**, *ICML* 2022]

Under mild assumptions on Σ and Γ, if the distributions P, Q are **Σ-invariant**, then

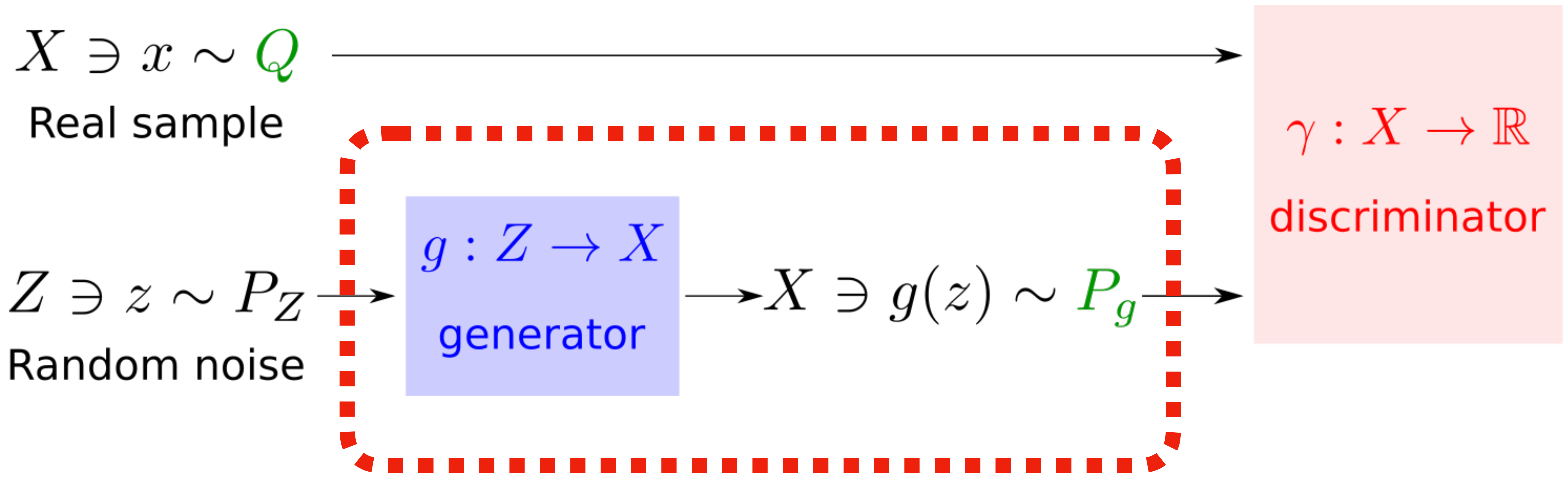
$$D^{\Gamma}(Q\|P) = D^{\Gamma_{\Sigma}^{\mathrm{inv}}}(Q\|P) = \sup_{\gamma\in\Gamma_{\Sigma}^{\mathrm{inv}}} H(\gamma; Q, P),$$

- $\Gamma_{\Sigma}^{\mathrm{inv}} \subset \Gamma$ is the subset of **Σ-invariant** “smarter” discriminators
- $\Gamma_{\Sigma}^{\mathrm{inv}}$ serves as an **unbiased regularization** to prevent discriminator **overfitting**.

Theorem 2: "smarter" generator



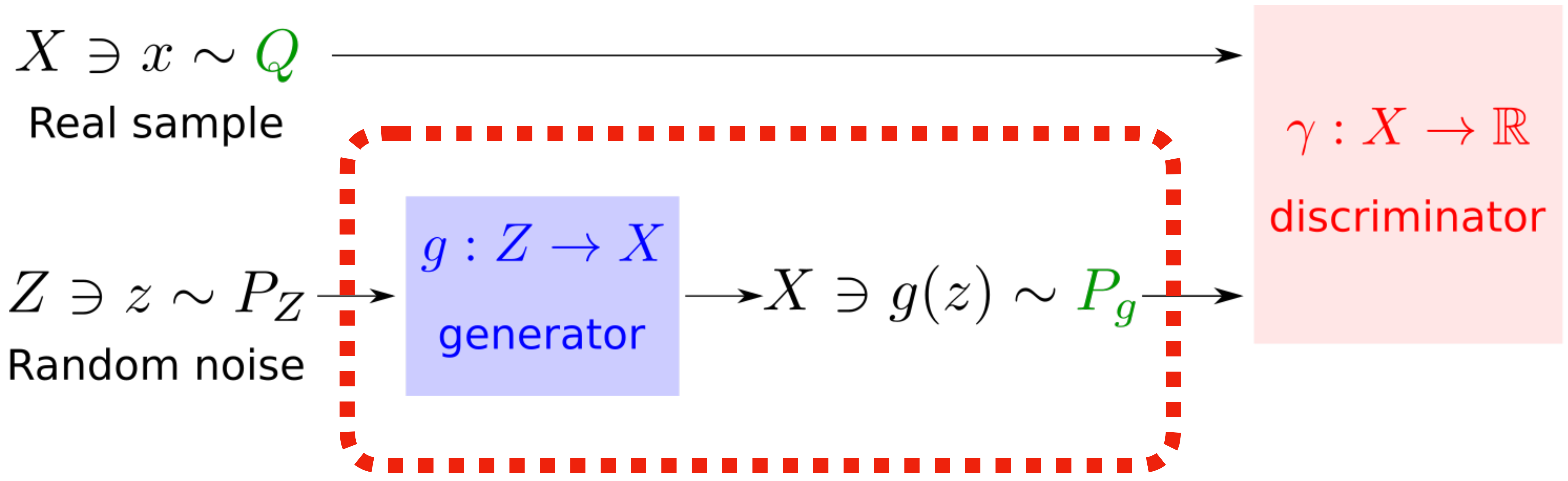
Theorem 2: “smarter” generator



Theorem [Birrell, Katsoulakis, Rey-Bellet, **Z.**, *ICML* 2022]

If P_Z is Σ-invariant and $g : Z \to X$ is Σ-equivariant, the generated measure P_g is Σ-invariant.

Theorem 2: “smarter” generator

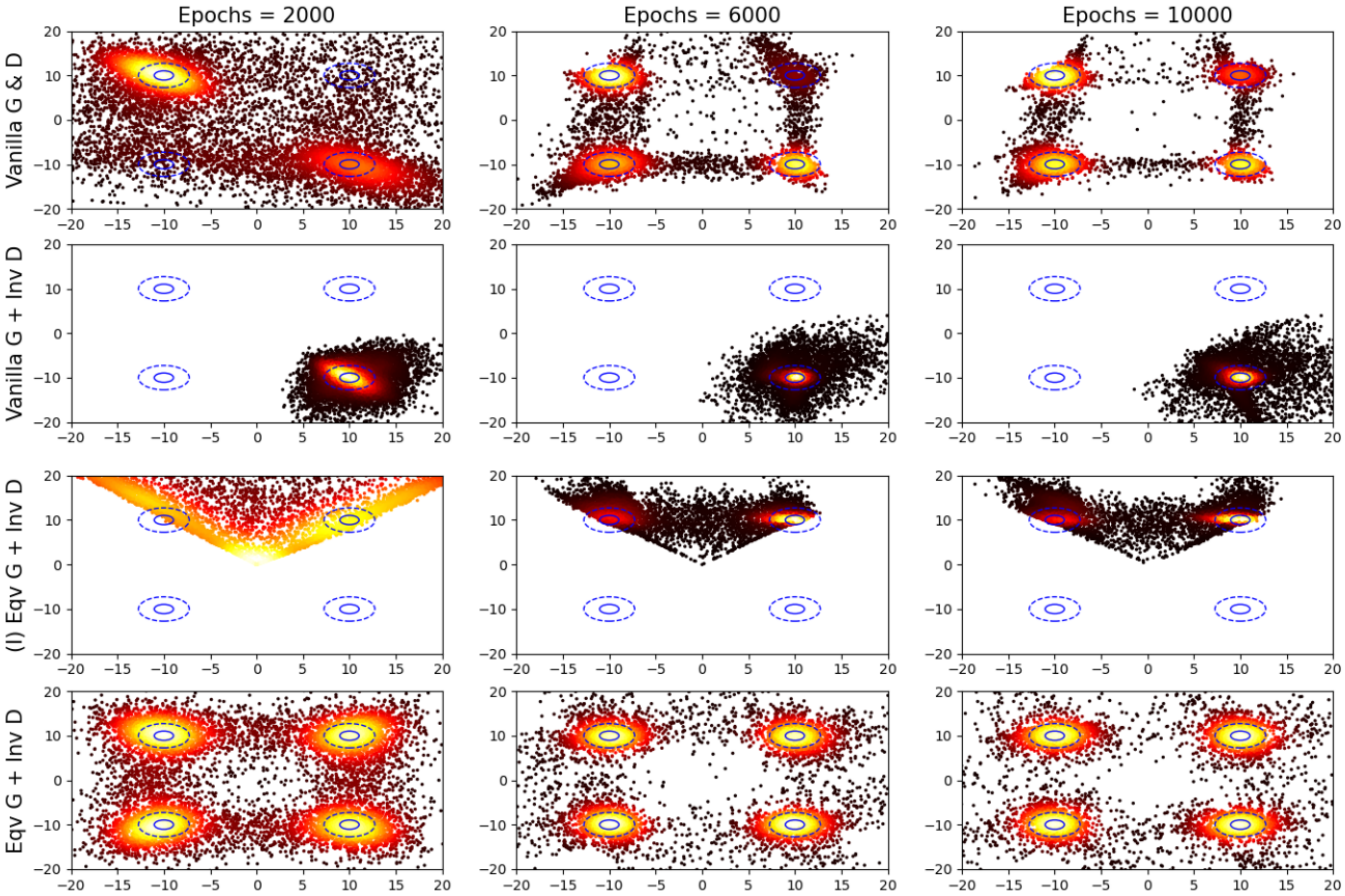


Theorem [Birrell, Katsoulakis, Rey-Bellet, **Z.**, *ICML* 2022]

If P_Z is Σ-invariant and $g : Z \to X$ is Σ-equivariant, the generated measure P_g is Σ-invariant.

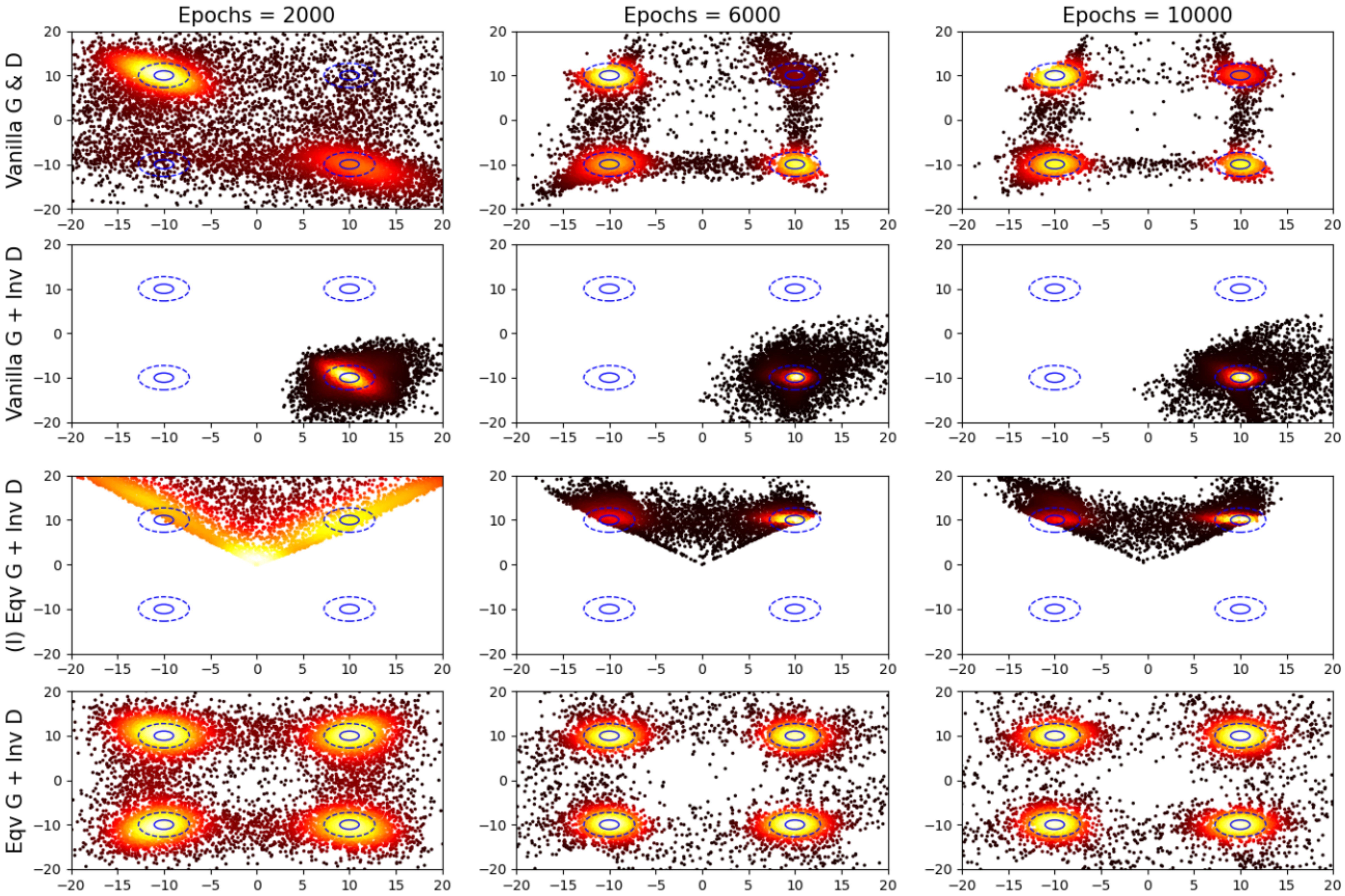
- Structure information embedded in the “smarter” generator **and** noise source.

Two “smart” players



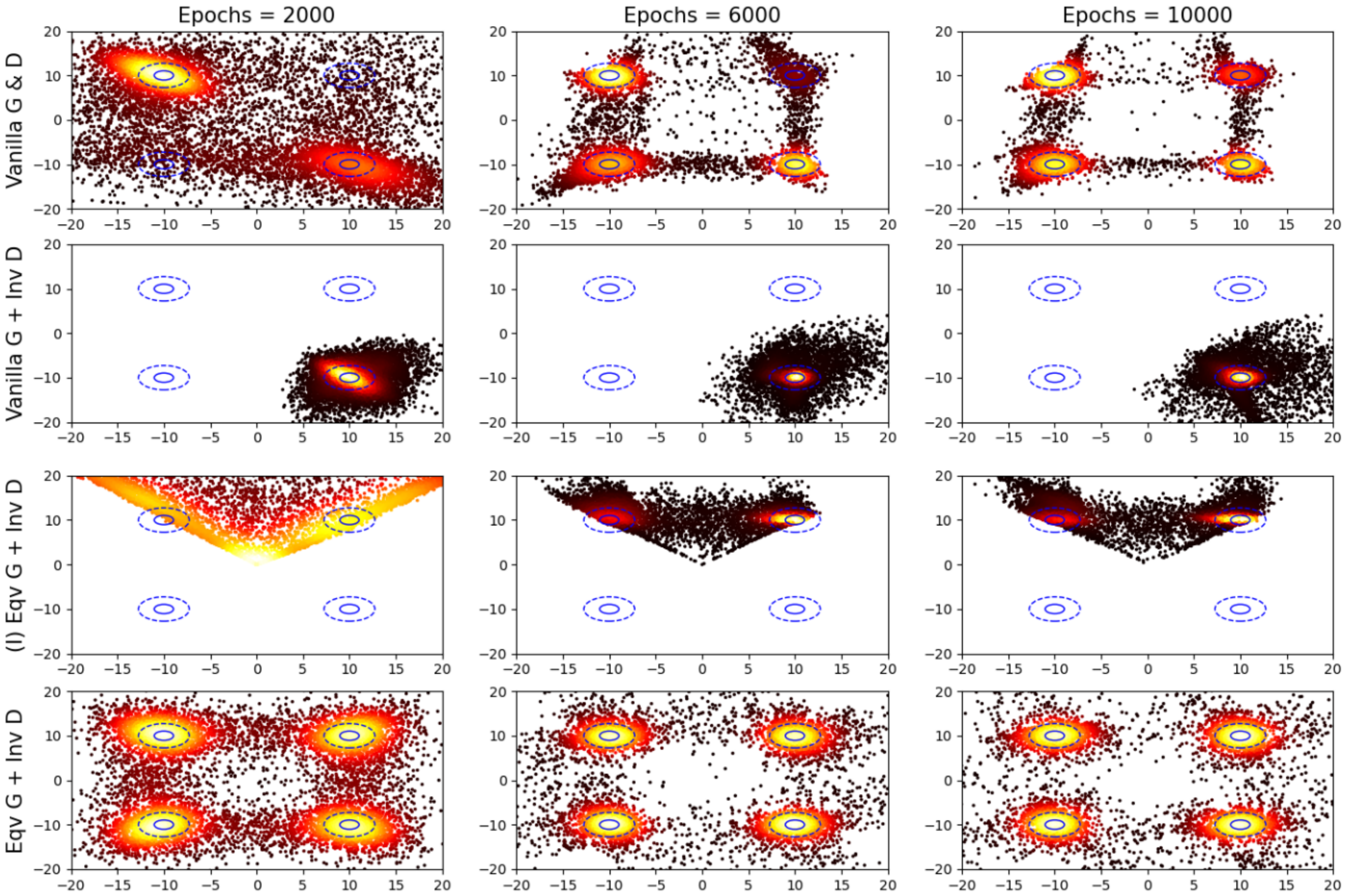
Two “smart” players

“Ignorant”
players need lots of
data, lots of time…
(the usual GANs)



Two “smart” players

“Ignorant” players need lots of data, lots of time… (the usual GANs)



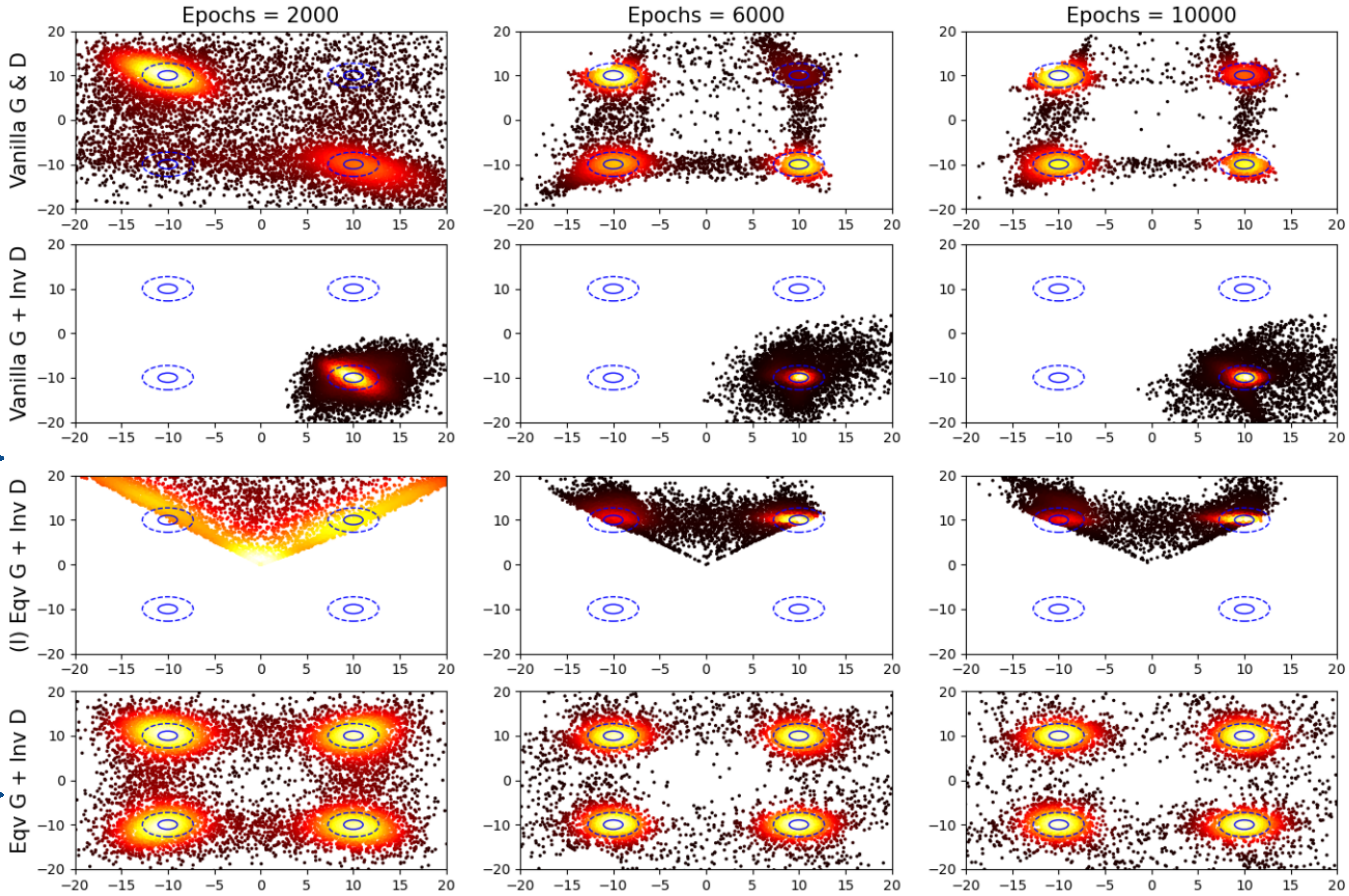
“Smart” players learn faster and better (our GANs)

Two “smart” players

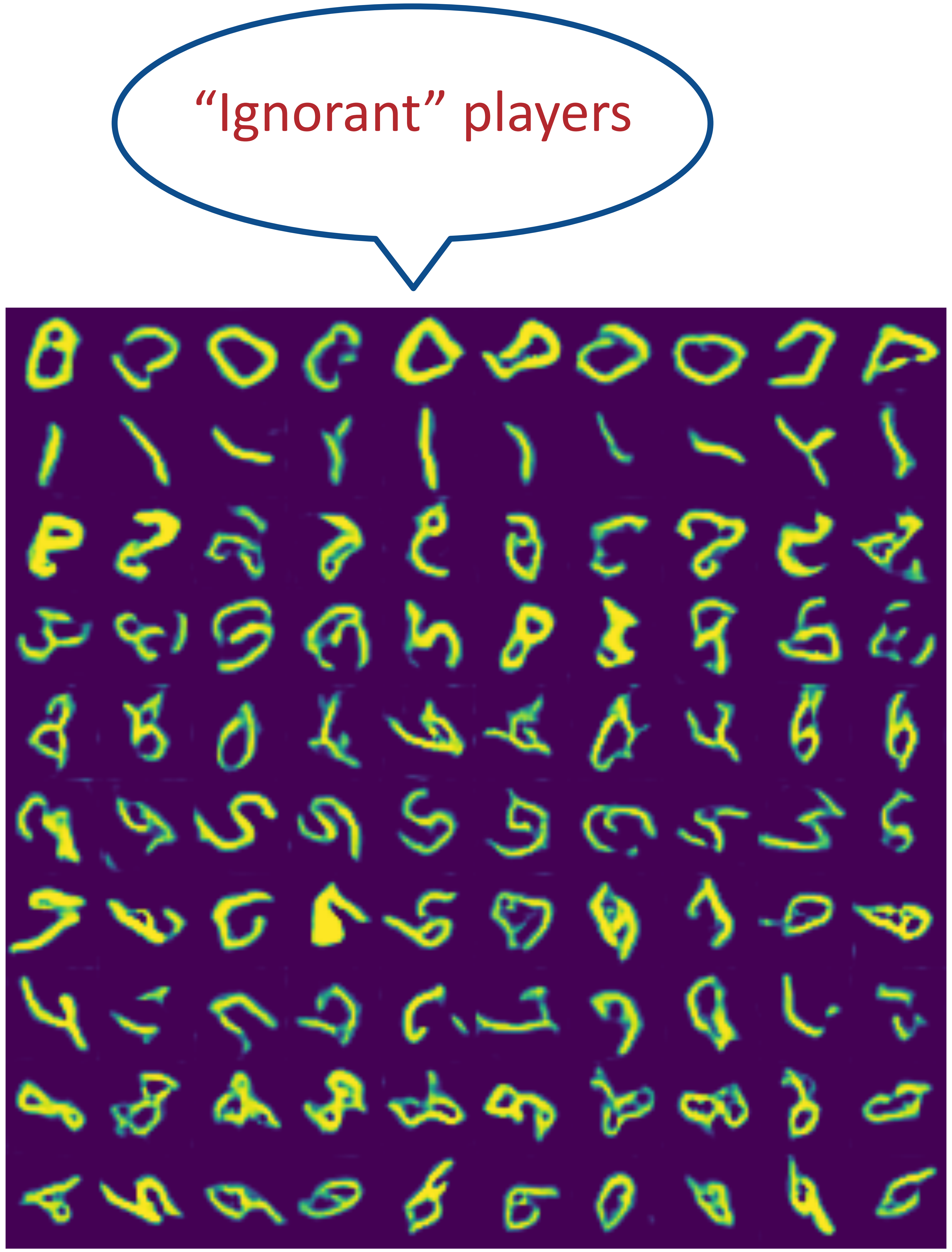
“Ignorant” players need lots of data, lots of time… (the usual GANs)

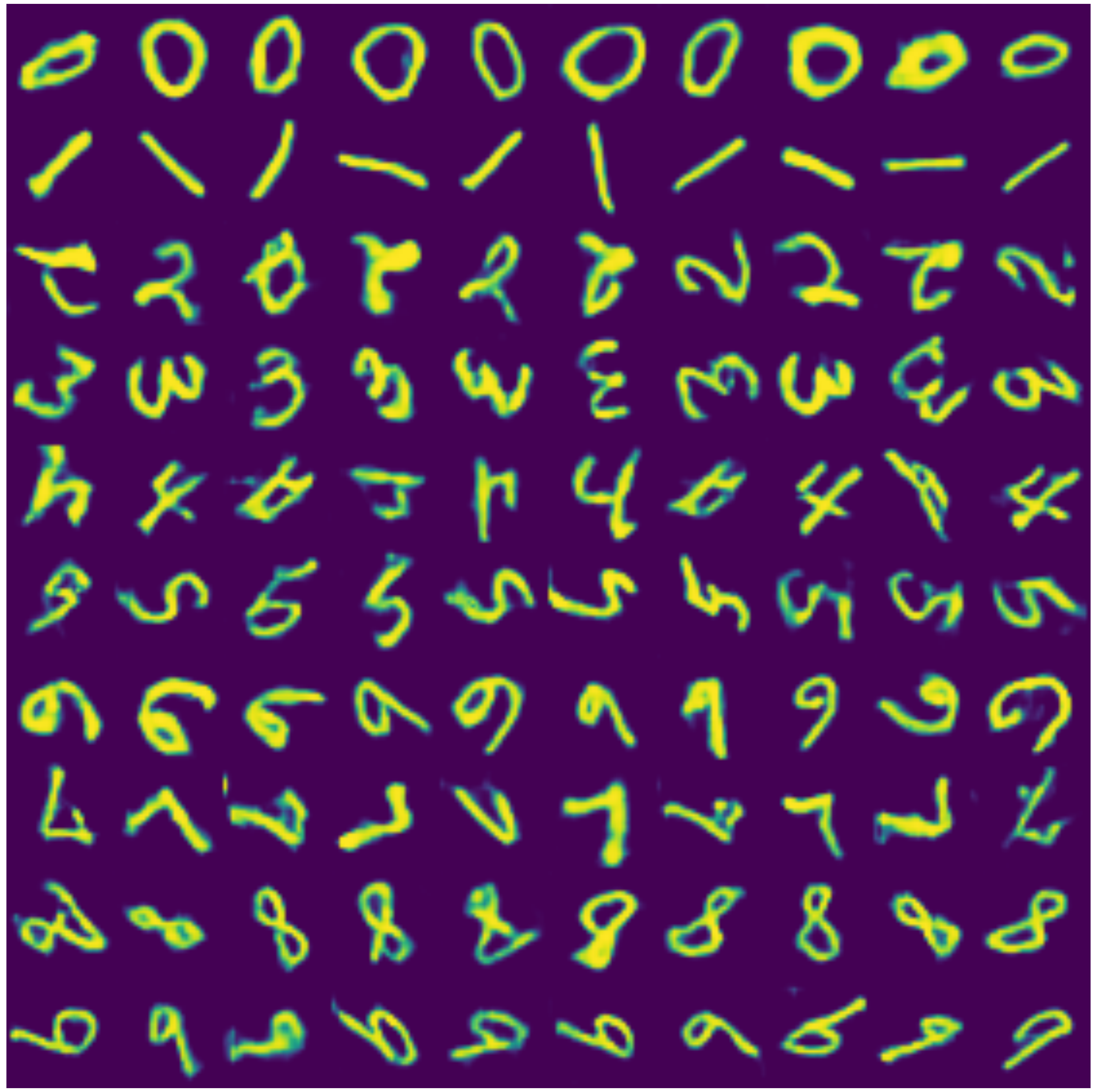
Players need to be “equally smart”: no weak links!

“Smart” players learn faster and better (our GANs)



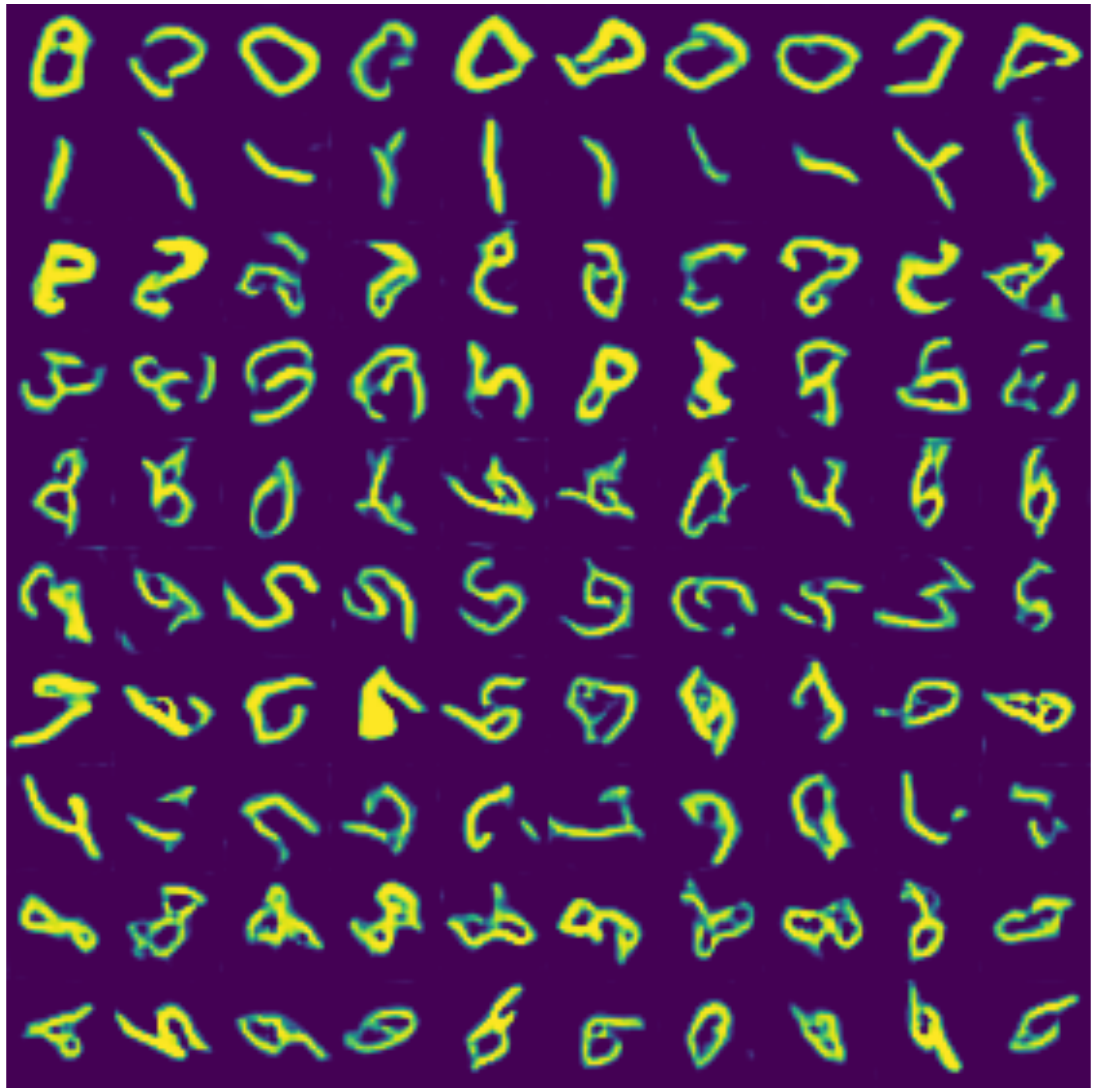
RotMNIST with 1% training samples



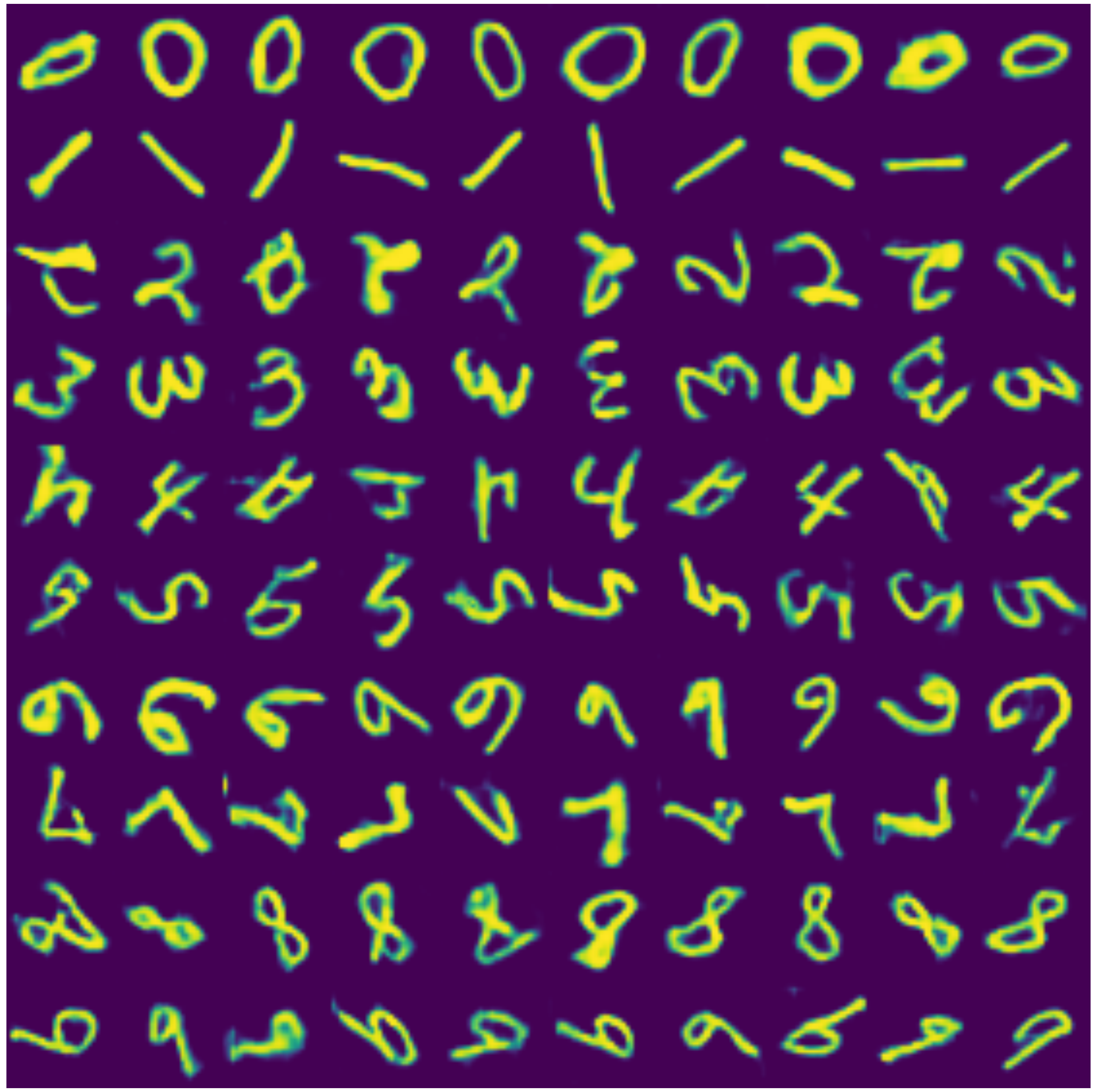


RotMNIST with 1% training samples

“Ignorant” players



“Smart” players

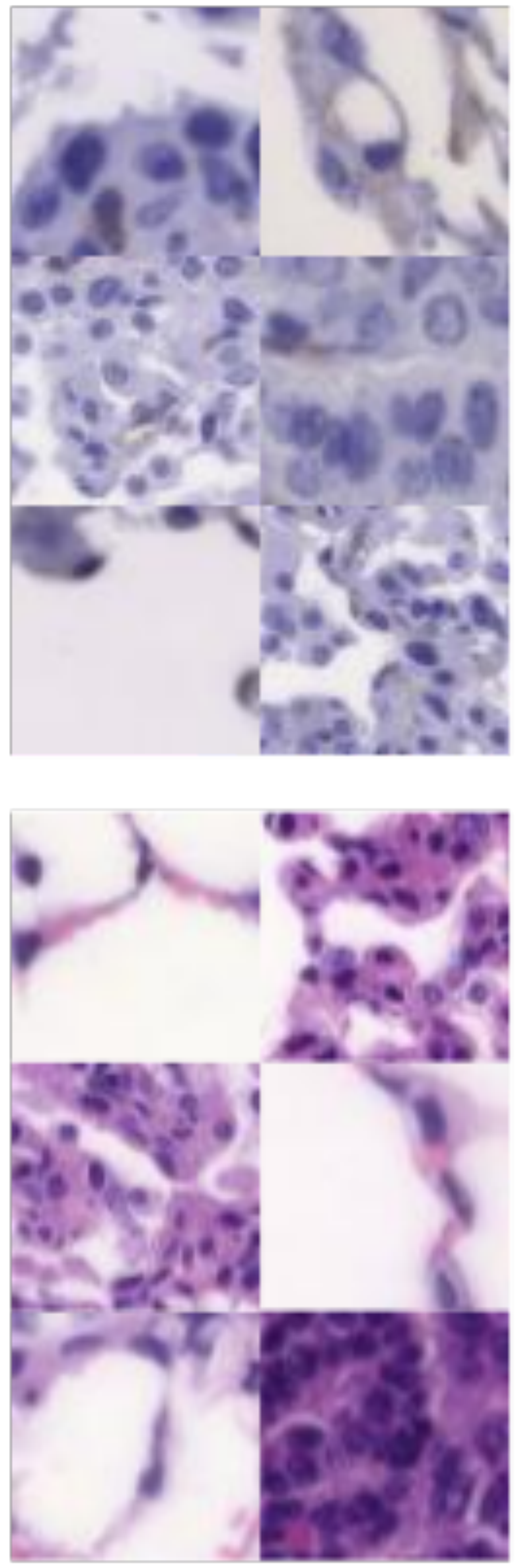


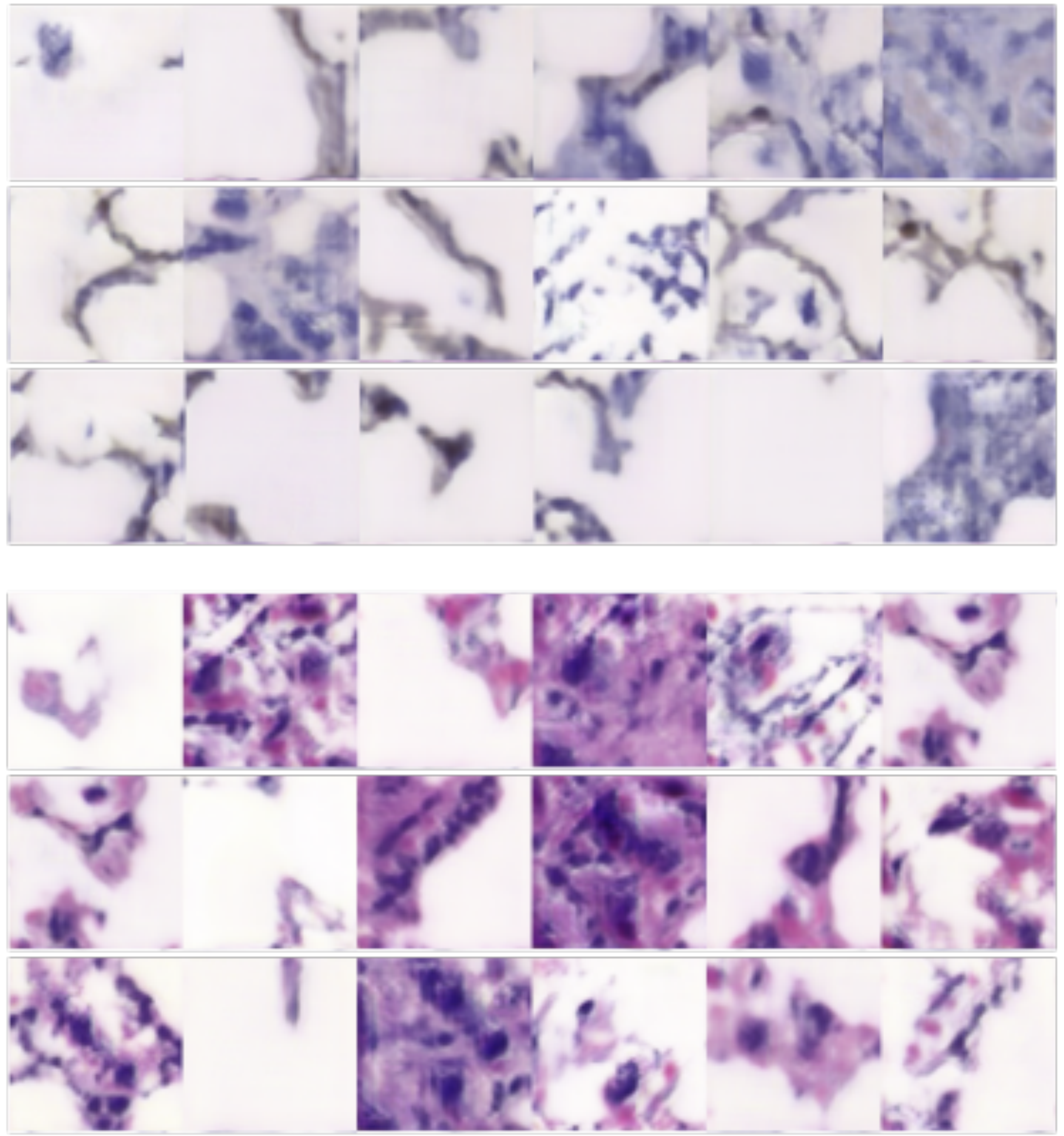
Medical images (ANHIR)

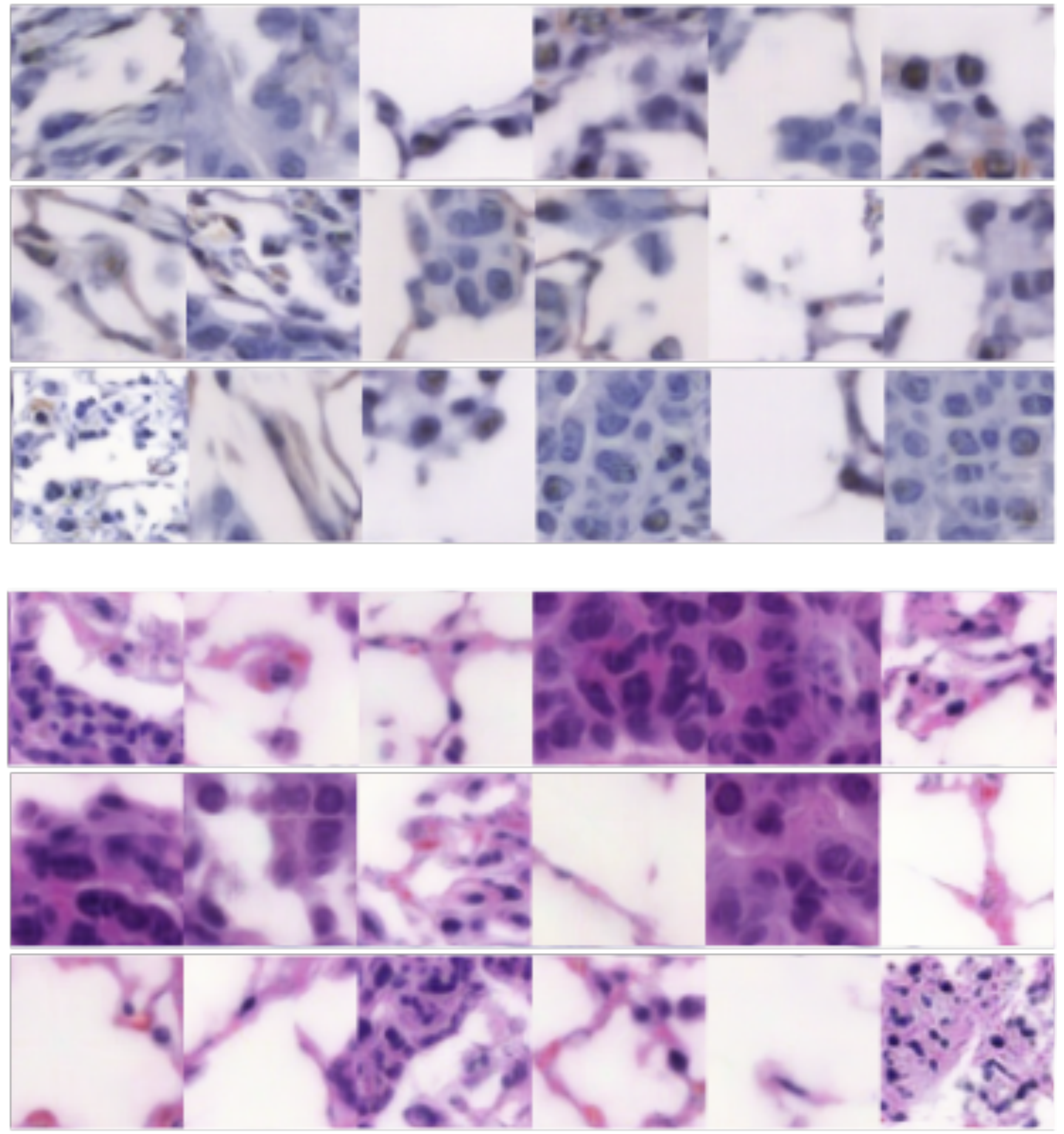
“Ignorant” players

“Smart” players

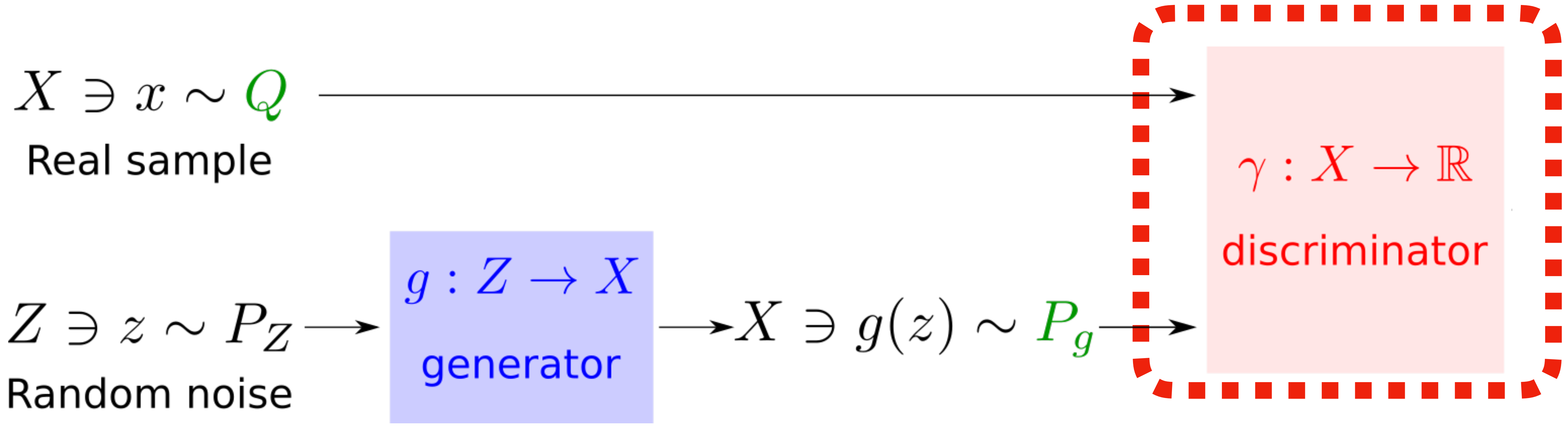
Real Samples



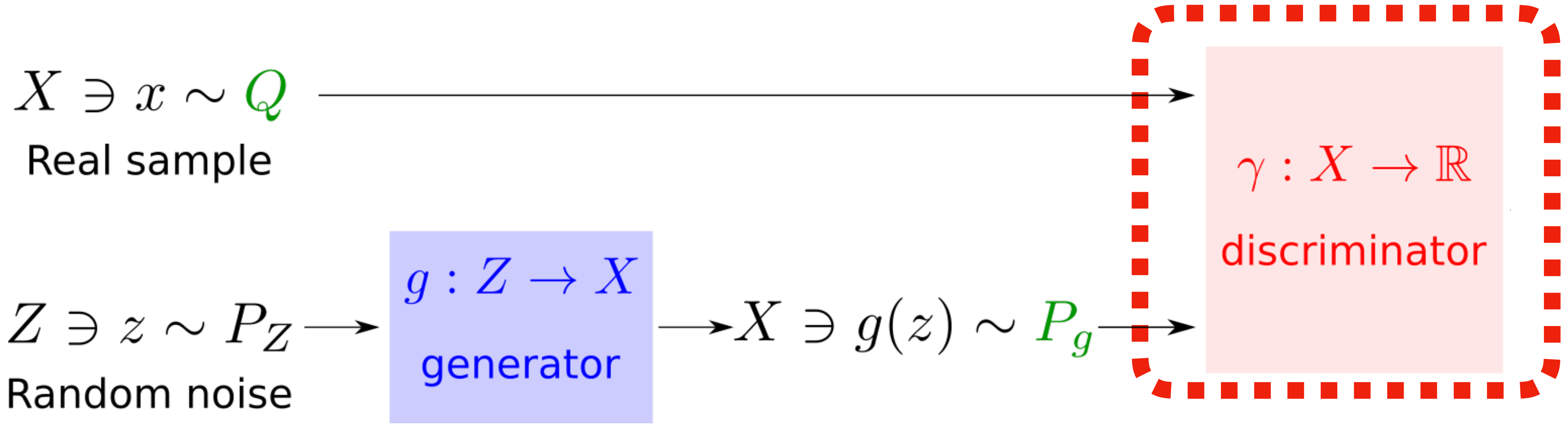




What is the reason behind the improvement?



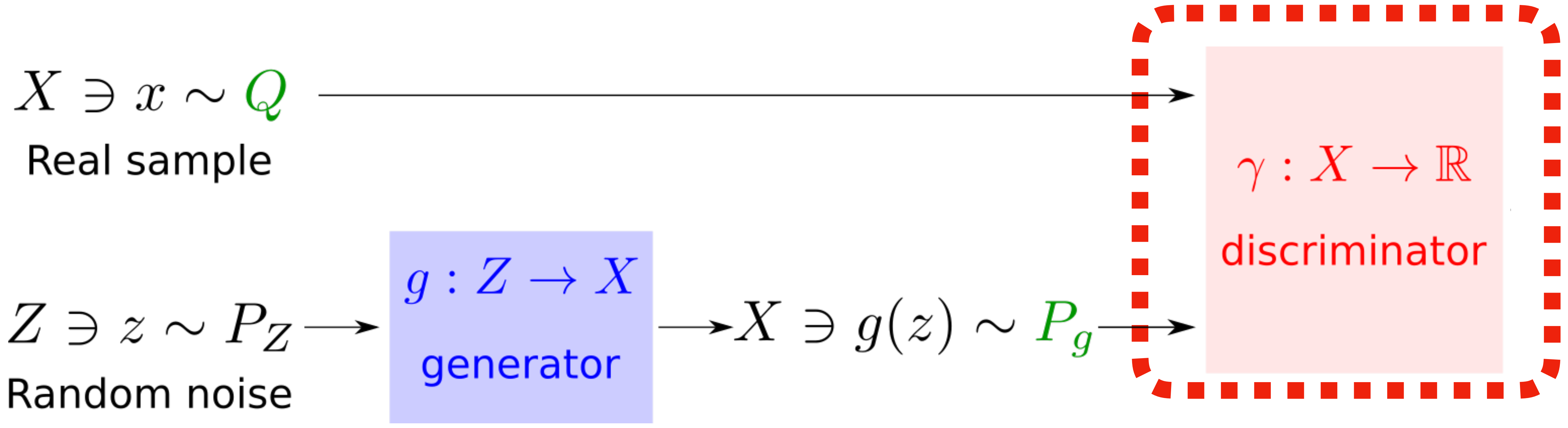
What is the reason behind the improvement?



P, Q are Σ**-invariant** $\Longrightarrow D^{\Gamma}(Q\|P) = D^{\Gamma_{\Sigma}^{\text{inv}}}(Q\|P)$,

- Reducing Γ to $\Gamma_{\Sigma}^{\text{inv}}$ provides a better **empirical estimation** for $D^{\Gamma}(Q\|P)$.

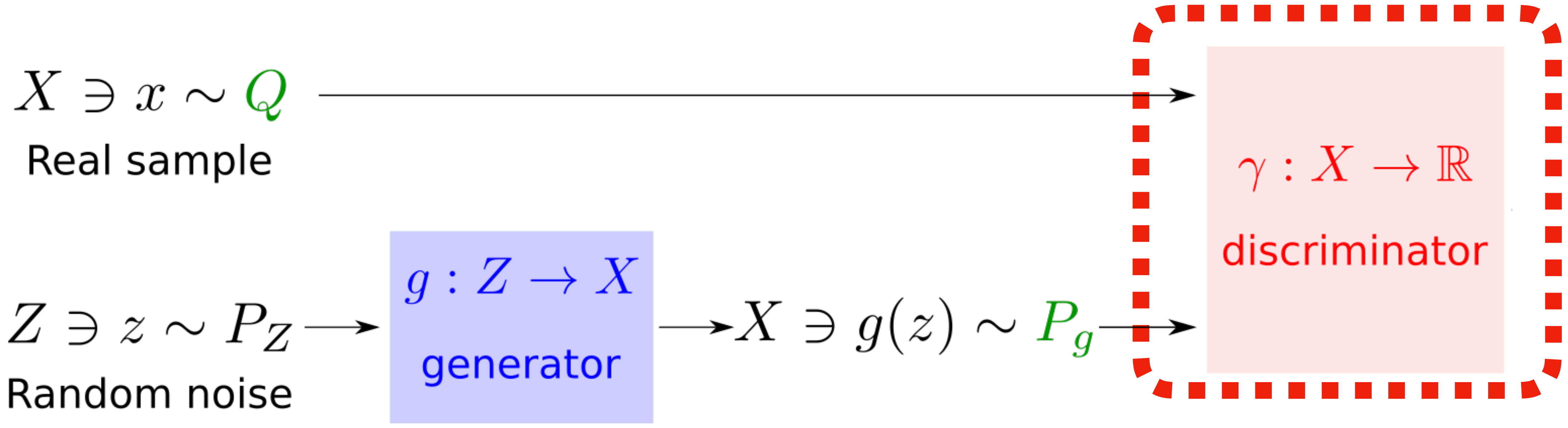
What is the reason behind the improvement?



P, Q are **Σ-invariant** $\Longrightarrow D^{\Gamma}(Q \| P) = D^{\Gamma_{\Sigma}^{\text{inv}}}(Q \| P)$,

- Reducing Γ to $\Gamma_{\Sigma}^{\text{inv}}$ provides a better **empirical estimation** for $D^{\Gamma}(Q \| P)$.
- $(x_1, \cdots, x_m) \sim P$, $(y_1, \cdots, y_n) \sim Q \Longrightarrow$ **Empirical measures** $P_m = \frac{1}{m} \sum_{i=1}^{m} \delta_{x_i}$, $Q_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{y_i}$

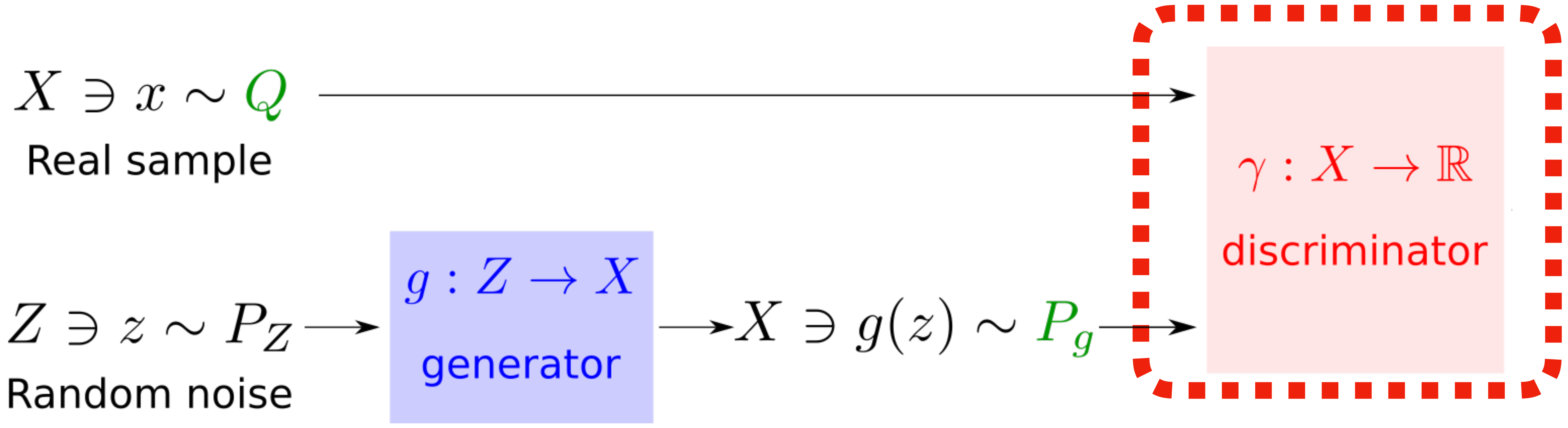
What is the reason behind the improvement?



P, Q are **Σ-invariant** $\Longrightarrow D^{\Gamma}(Q\|P) = D^{\Gamma_{\Sigma}^{\text{inv}}}(Q\|P)$,

- Reducing Γ to $\Gamma_{\Sigma}^{\text{inv}}$ provides a better **empirical estimation** for $D^{\Gamma}(Q\|P)$.
- $(x_1, \cdots, x_m) \sim P$, $(y_1, \cdots, y_n) \sim Q \Longrightarrow$ **Empirical measures** $P_m = \dfrac{1}{m}\sum_{i=1}^{m} \delta_{x_i}$, $Q_n = \dfrac{1}{n}\sum_{i=1}^{n} \delta_{y_i}$
- $D^{\Gamma}(Q\|P) \approx \cancel{D^{\Gamma}(Q_n\|P_m)} = D^{\Gamma_{\Sigma}^{\text{inv}}}(Q_n\|P_m)$

What is the reason behind the improvement?



P, Q are **Σ-invariant** $\Longrightarrow D^{\Gamma}(Q\|P) = D^{\Gamma_\Sigma^{\text{inv}}}(Q\|P),$

- Reducing Γ to $\Gamma_\Sigma^{\text{inv}}$ provides a better **empirical estimation** for $D^{\Gamma}(Q\|P)$.
- $(x_1, \cdots, x_m) \sim P, (y_1, \cdots, y_n) \sim Q \Longrightarrow$ **Empirical measures** $P_m = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \delta_{x_i}, Q_n = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \delta_{y_i}$
- $D^{\Gamma}(Q\|P) \approx \cancel{D^{\Gamma}(Q_n\|P_m)} = D^{\Gamma_\Sigma^{\text{inv}}}(Q_n\|P_m)$

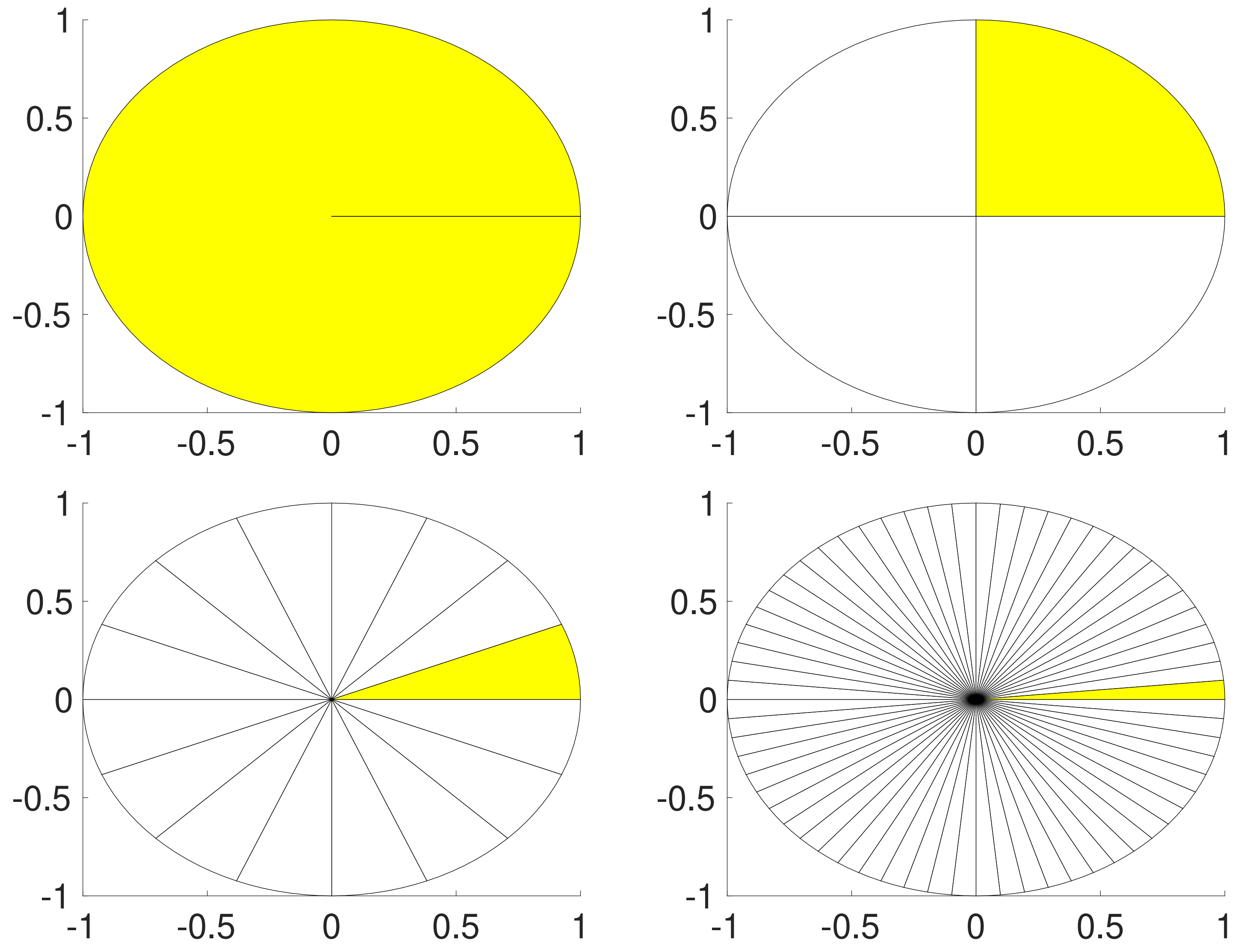
Question: How much **more accurate** is the new estimation?

Wasserstein-1 metric

- $W(Q, P) = \sup_{\gamma \in \Gamma} \{\mathbb{E}_Q[\gamma] - \mathbb{E}_P[\gamma]\}$. $\Gamma = \text{Lip}_1(X)$
- **Estimator:** $W^{\Sigma}(Q_n, P_m) = \sup_{\gamma \in \Gamma_{\Sigma}^{\text{inv}}} \{\mathbb{E}_{Q_n}[\gamma] - \mathbb{E}_{P_m}[\gamma]\}$

Wasserstein-1 metric

- $W(Q,P) = \sup_{\gamma\in\Gamma}\{\mathbb{E}_Q[\gamma] - \mathbb{E}_P[\gamma]\}$. $\Gamma = \mathrm{Lip}_1(X)$
- **Estimator:** $W^{\Sigma}(Q_n, P_m) = \sup_{\gamma\in\Gamma_{\Sigma}^{\mathrm{inv}}}\{\mathbb{E}_{Q_n}[\gamma] - \mathbb{E}_{P_m}[\gamma]\}$



Theorem [Chen, Katsoulakis, Rey-Bellet, **Z.**, *ICML* 2023]

$X = \Sigma \times X_0$ bounded in $\mathbb{R}^d$, and $P, Q \in \mathscr{P}_{\Sigma}(X)$ are Σ-invariant. With high probability,

- when $d \geq 2$: $\forall s > 0, \left| W(Q,P) - W^{\Sigma}(Q_n, P_m) \right| \leq C\left(\left(\frac{1}{|\Sigma|m}\right)^{\frac{1}{d+s}} + \left(\frac{1}{|\Sigma|n}\right)^{\frac{1}{d+s}}\right)$
- when $d = 1$: $|W(Q,P) - W^{\Sigma}(Q_n, P_m)| \leq C \cdot \mathrm{diam}(X_0)\left(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}}\right)$

Maximum Mean Discrepancy (MMD)

- $\text{MMD}(Q, P) = \sup_{\gamma \in \Gamma} \{\mathbb{E}_Q[\gamma] - \mathbb{E}_P[\gamma]\}$. Γ is the unit ball in some **RKHS** $\mathscr{H}$ with kernel $k(x, y)$.
- **Estimator:** $\text{MMD}^{\Sigma}(Q_n, P_m) = \sup_{\gamma \in \Gamma_\Sigma^{\text{inv}}} \{\mathbb{E}_{Q_n}[\gamma] - \mathbb{E}_{P_m}[\gamma]\}$

Maximum Mean Discrepancy (MMD)

- $\mathrm{MMD}(Q, P) = \sup_{\gamma \in \Gamma} \{\mathbb{E}_Q[\gamma] - \mathbb{E}_P[\gamma]\}$. Γ is the unit ball in some **RKHS** $\mathscr{H}$ with kernel $k(x, y)$.
- **Estimator:** $\mathrm{MMD}^{\Sigma}(Q_n, P_m) = \sup_{\gamma \in \Gamma_\Sigma^{\mathrm{inv}}} \{\mathbb{E}_{Q_n}[\gamma] - \mathbb{E}_{P_m}[\gamma]\}$

Theorem [Chen, Katsoulakis, Rey-Bellet, **Z.**, *ICML* 2023]

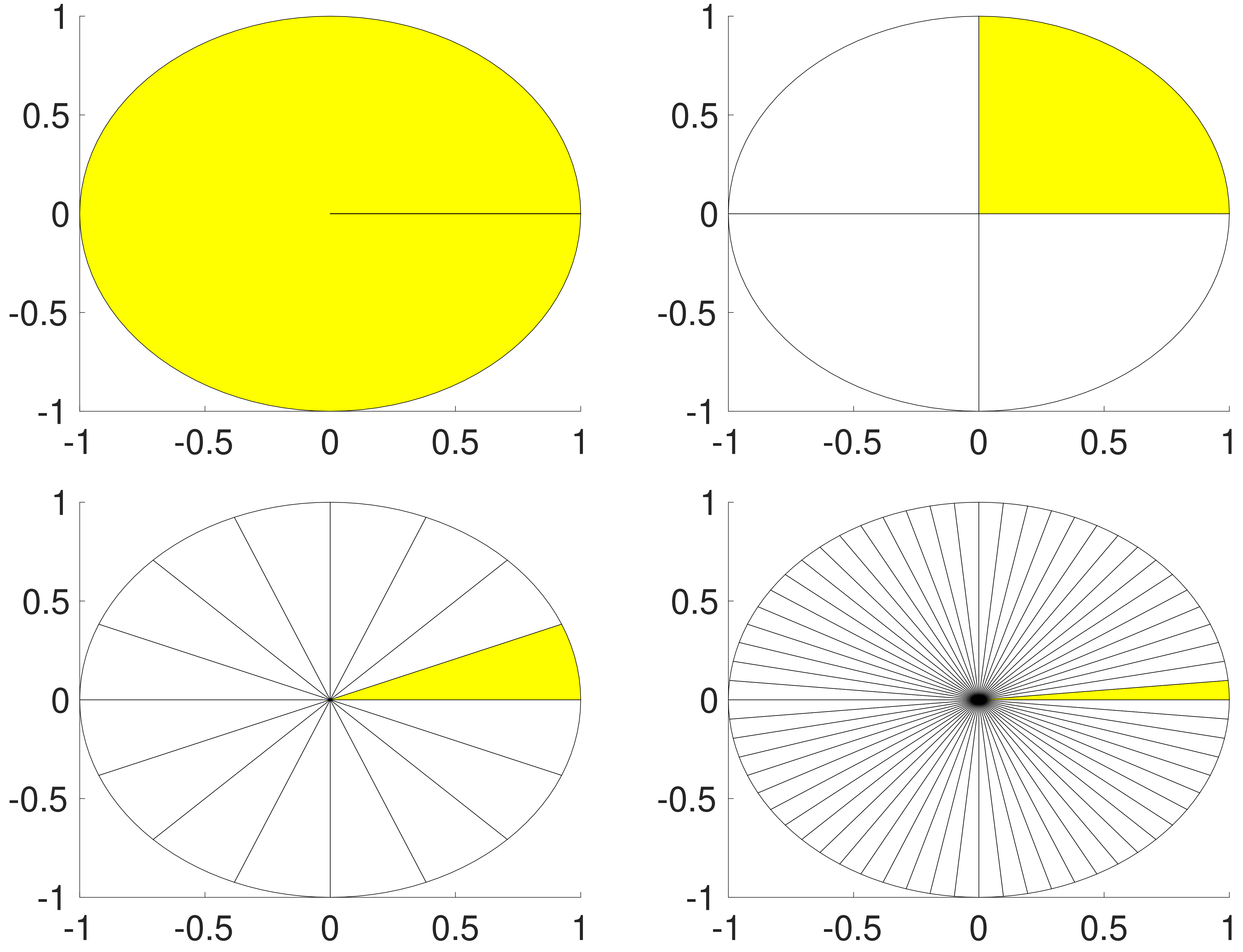
$X = \Sigma \times X_0$ bounded in $\mathbb{R}^d$, and $P, Q \in \mathscr{P}_\Sigma(X)$ are Σ-invariant. With high probability,

$$\left| \mathrm{MMD}(Q, P) - \mathrm{MMD}^{\Sigma}(Q_n, P_m) \right| = O\left(C_{\Sigma,k} \left(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}} \right) \right),$$

where $C_{\Sigma,k} = \sqrt{a_{\Sigma,k} + \frac{1 - a_{\Sigma,k}}{|\Sigma|}}$, and $a_{\Sigma,k} \in (0,1)$ depends on Σ and the **kernel** $k(x, y)$.

Maximum Mean Discrepancy (MMD)

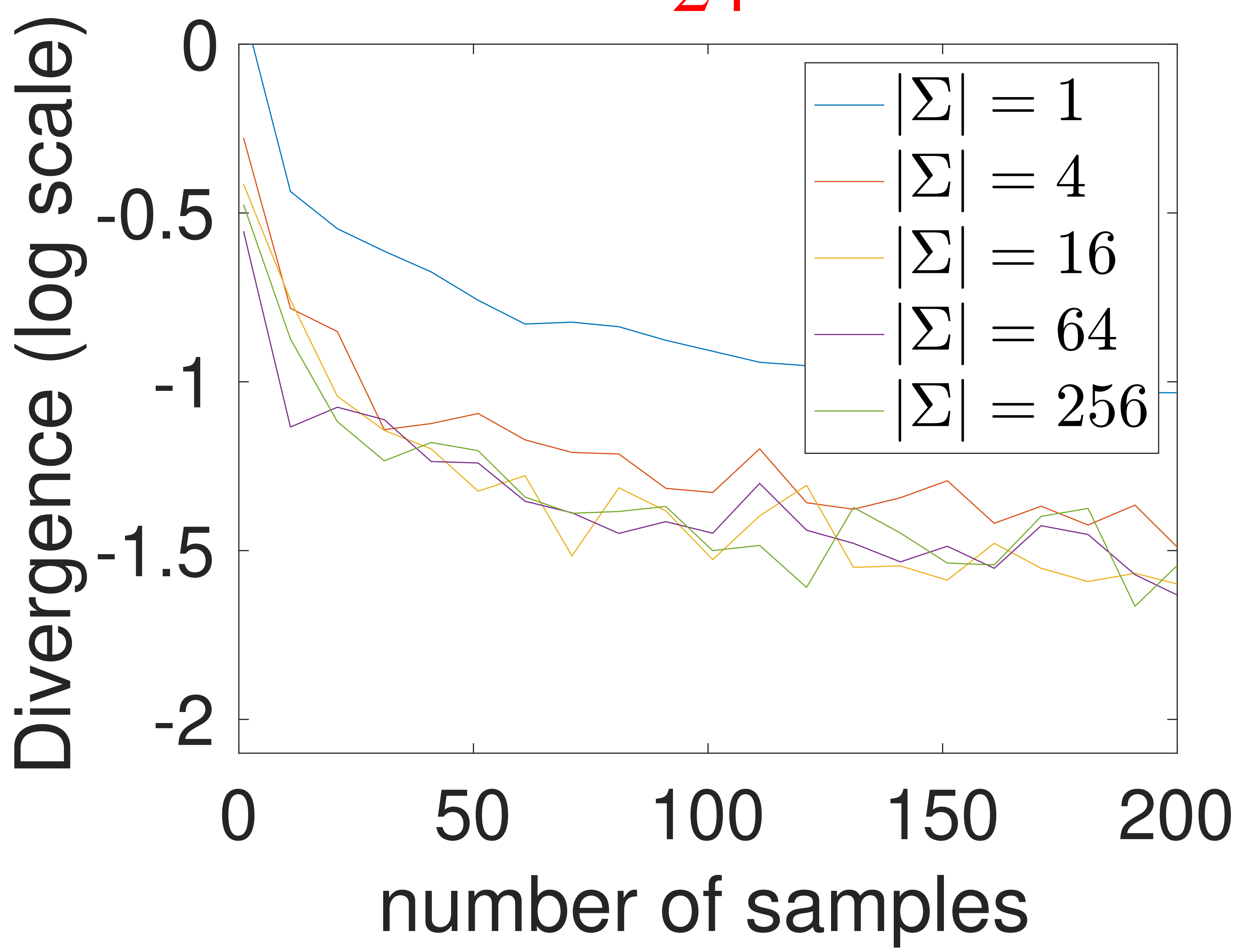
$$\left|\mathrm{MMD}(Q,P) - \mathrm{MMD}^{\Sigma}(Q_n,P_m)\right| = O\left(C_{\Sigma,k}\left(\frac{1}{\sqrt{m}}+\frac{1}{\sqrt{n}}\right)\right), \qquad C_{\Sigma,k} = \sqrt{a_{\Sigma,k} + \frac{1-a_{\Sigma,k}}{|\Sigma|}}$$



Maximum Mean Discrepancy (MMD)

$$\left| \mathrm{MMD}(Q, P) - \mathrm{MMD}^{\Sigma}(Q_n, P_m) \right| = O\left(C_{\Sigma,k} \left(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}} \right) \right), \quad C_{\Sigma,k} = \sqrt{a_{\Sigma,k} + \frac{1 - a_{\Sigma,k}}{|\Sigma|}}$$

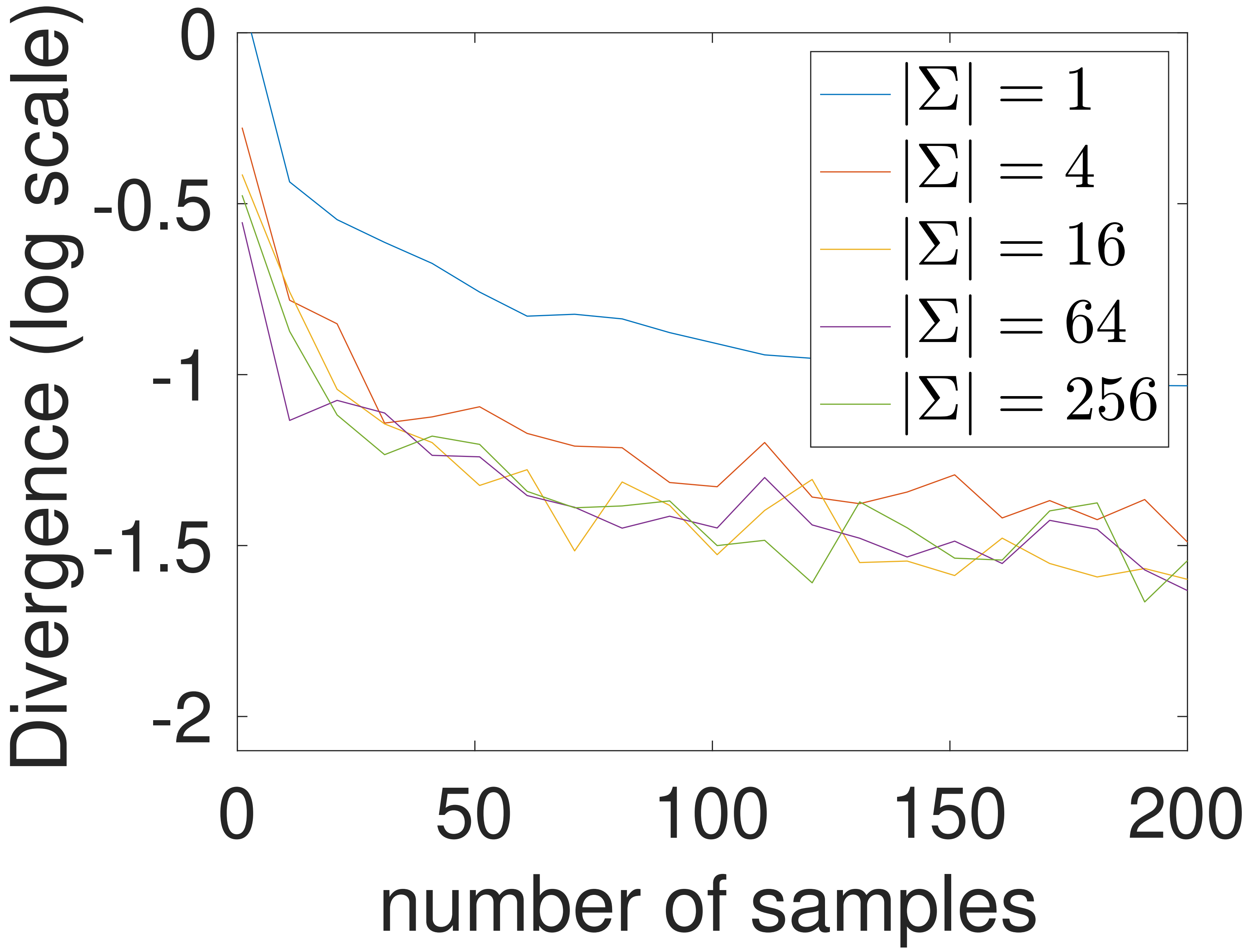
$$s = \frac{2\pi}{24}$$



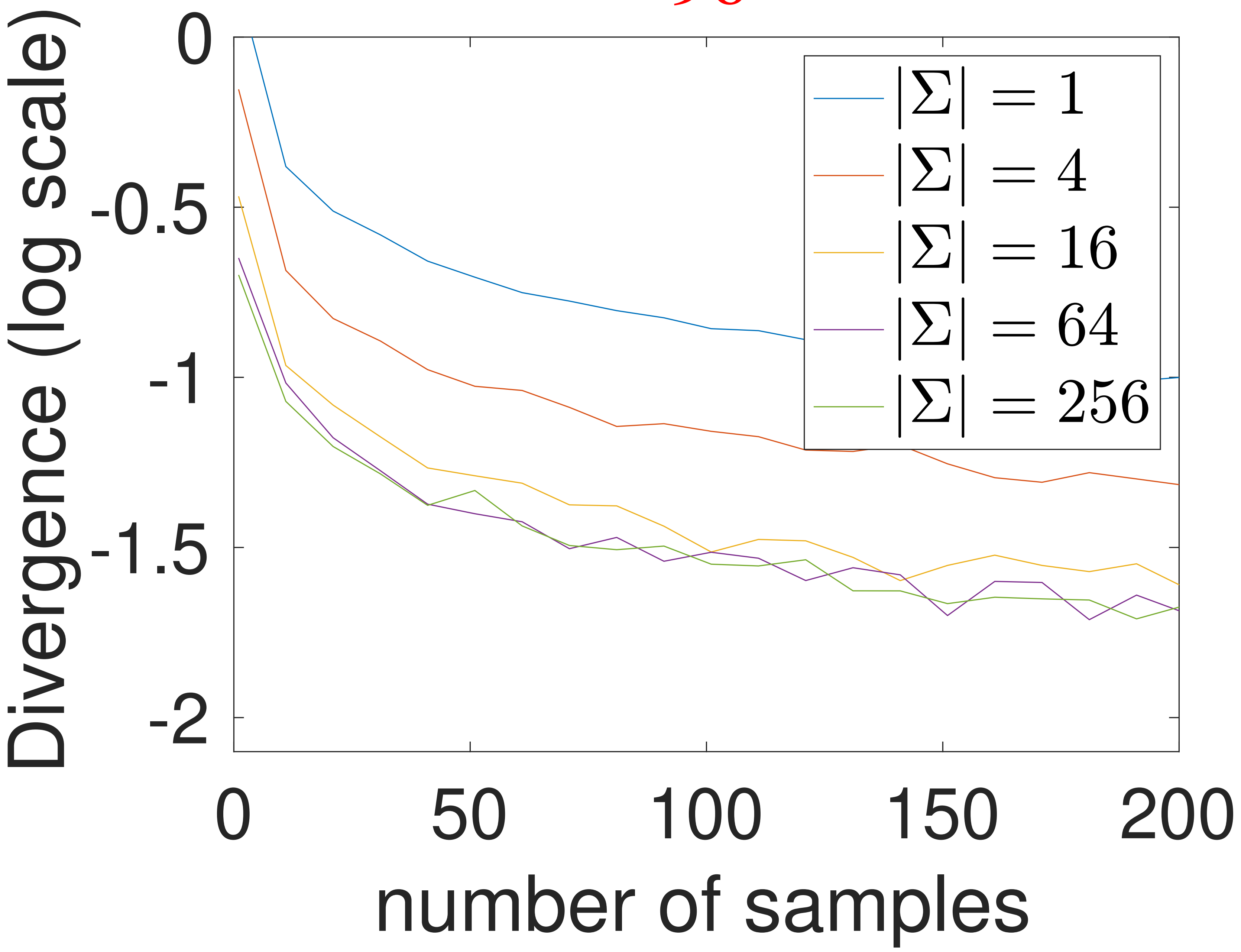
Maximum Mean Discrepancy (MMD)

$$\left|\mathrm{MMD}(Q,P) - \mathrm{MMD}^{\Sigma}(Q_n,P_m)\right| = O\left(C_{\Sigma,k}\left(\frac{1}{\sqrt{m}}+\frac{1}{\sqrt{n}}\right)\right), \qquad C_{\Sigma,k} = \sqrt{a_{\Sigma,k} + \frac{1-a_{\Sigma,k}}{|\Sigma|}}$$

$s = \frac{2\pi}{24}$



$s = \frac{2\pi}{96}$

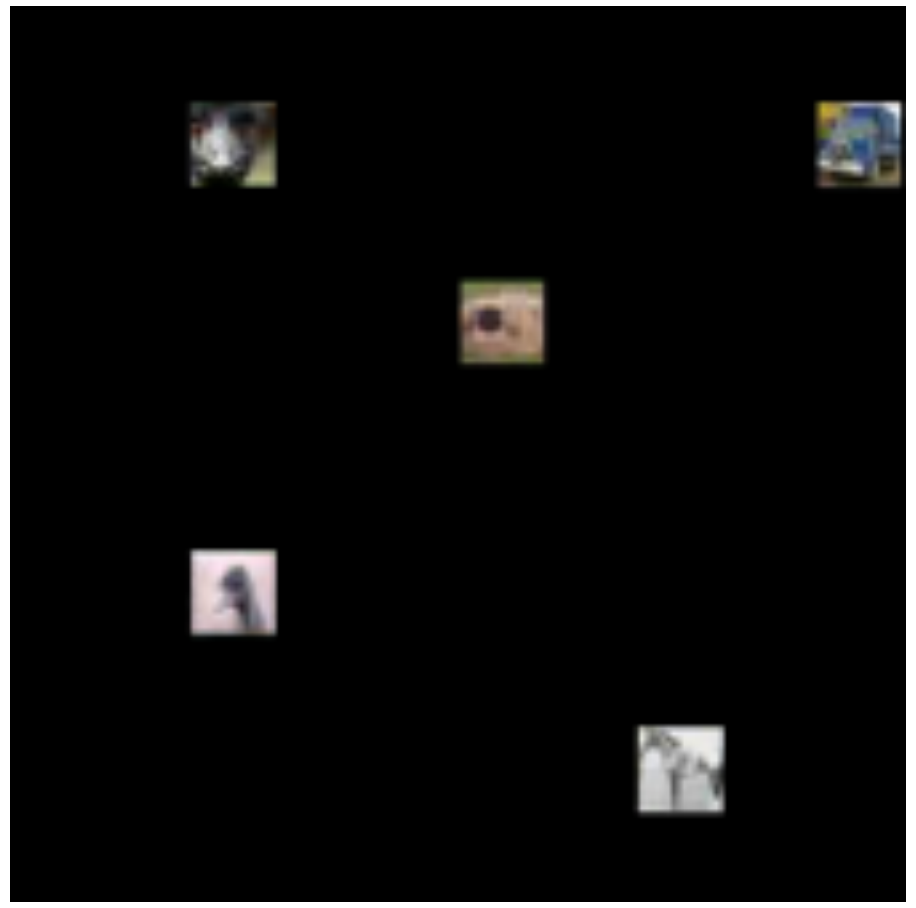


Maximum Mean Discrepancy (MMD)

$$\left|\mathrm{MMD}(Q,P) - \mathrm{MMD}^{\Sigma}(Q_n,P_m)\right| = O\left(C_{\Sigma,k}\left(\frac{1}{\sqrt{m}}+\frac{1}{\sqrt{n}}\right)\right), \qquad C_{\Sigma,k} = \sqrt{a_{\Sigma,k} + \frac{1-a_{\Sigma,k}}{|\Sigma|}}$$

Missing pieces

- **Exact quantification of the improvement**
  - **Sample complexity** and **error bound**.



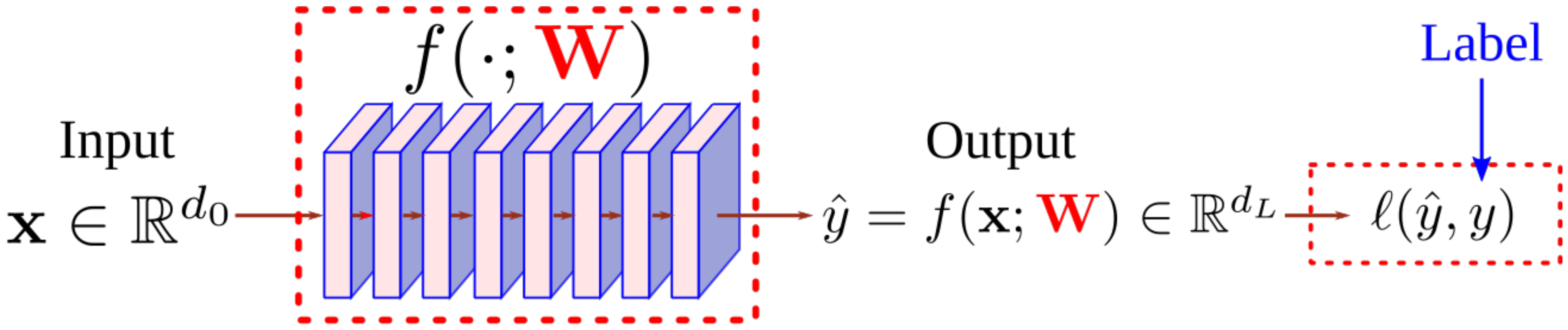
- **Does it converge? To what solution?**
  - **Training dynamics** of equivariant models

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathscr{L}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \mathbf{W}), y_i)$$

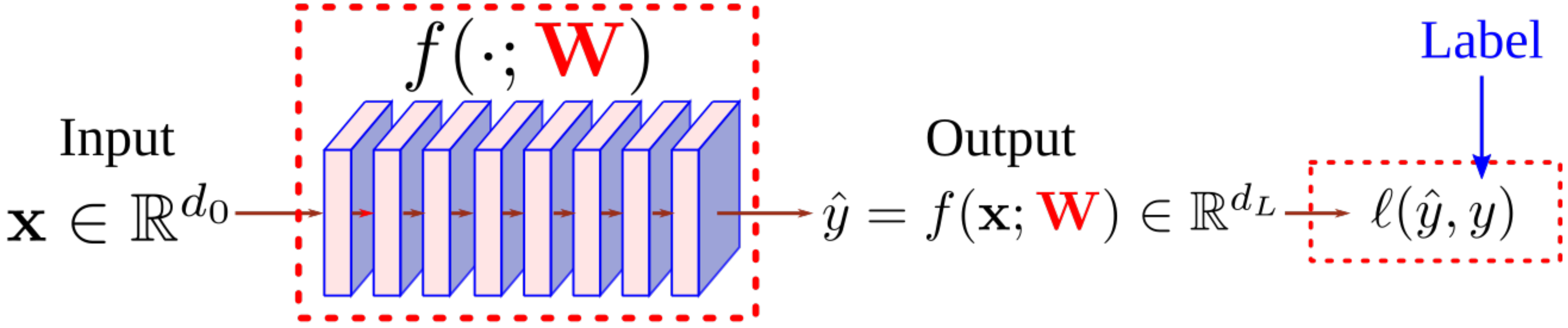
Implicit bias of linear equivariant networks

- Z. Chen and **W. Zhu**. “On the implicit bias of linear equivariant steerable networks”. *NeurIPS* (2023)
- Inspired by [Lawrence et al., *ICML* 2022]

Optimization (training) of G-CNN



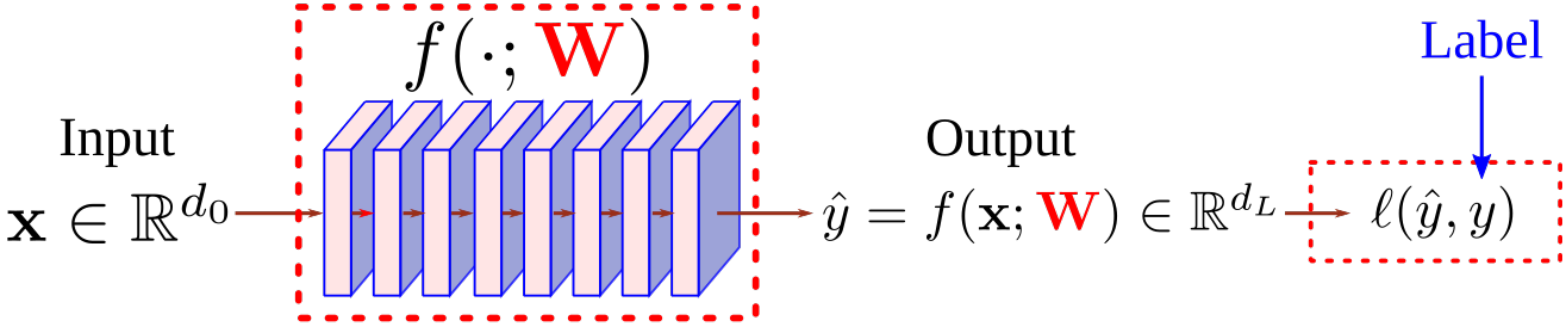
Optimization (training) of G-CNN



- Training a DNN on the a (labeled) data set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$:

$$\min_{\mathbf{W}} \mathscr{L}(\mathbf{W}; S) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \mathbf{W}), y_i)$$

Optimization (training) of G-CNN

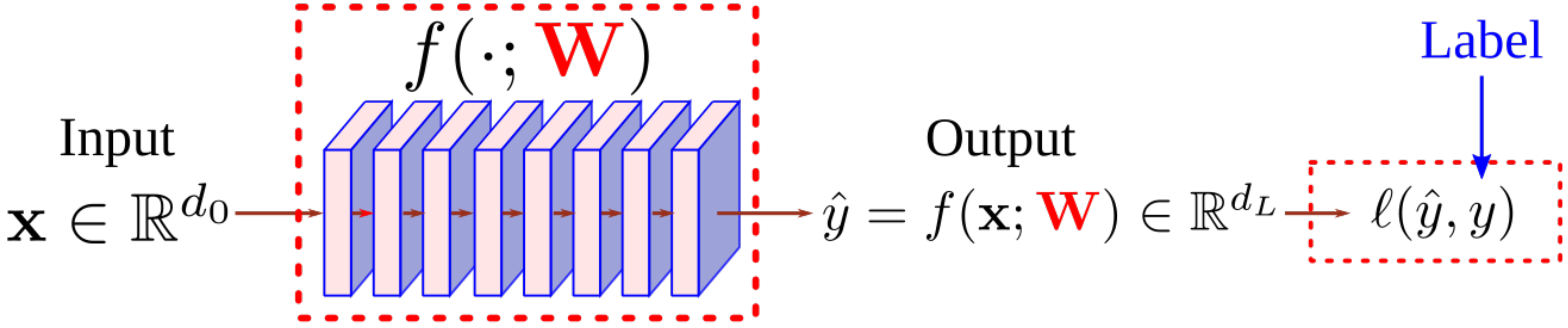


- Training a DNN on the a (labeled) data set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$:

$$\min_{\mathbf{W}} \mathscr{L}(\mathbf{W}; S) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \mathbf{W}), y_i)$$

- Design $f(\,\cdot\,; \mathbf{W})$ to respect group symmetry—**explicit regularization**.

Optimization (training) of G-CNN



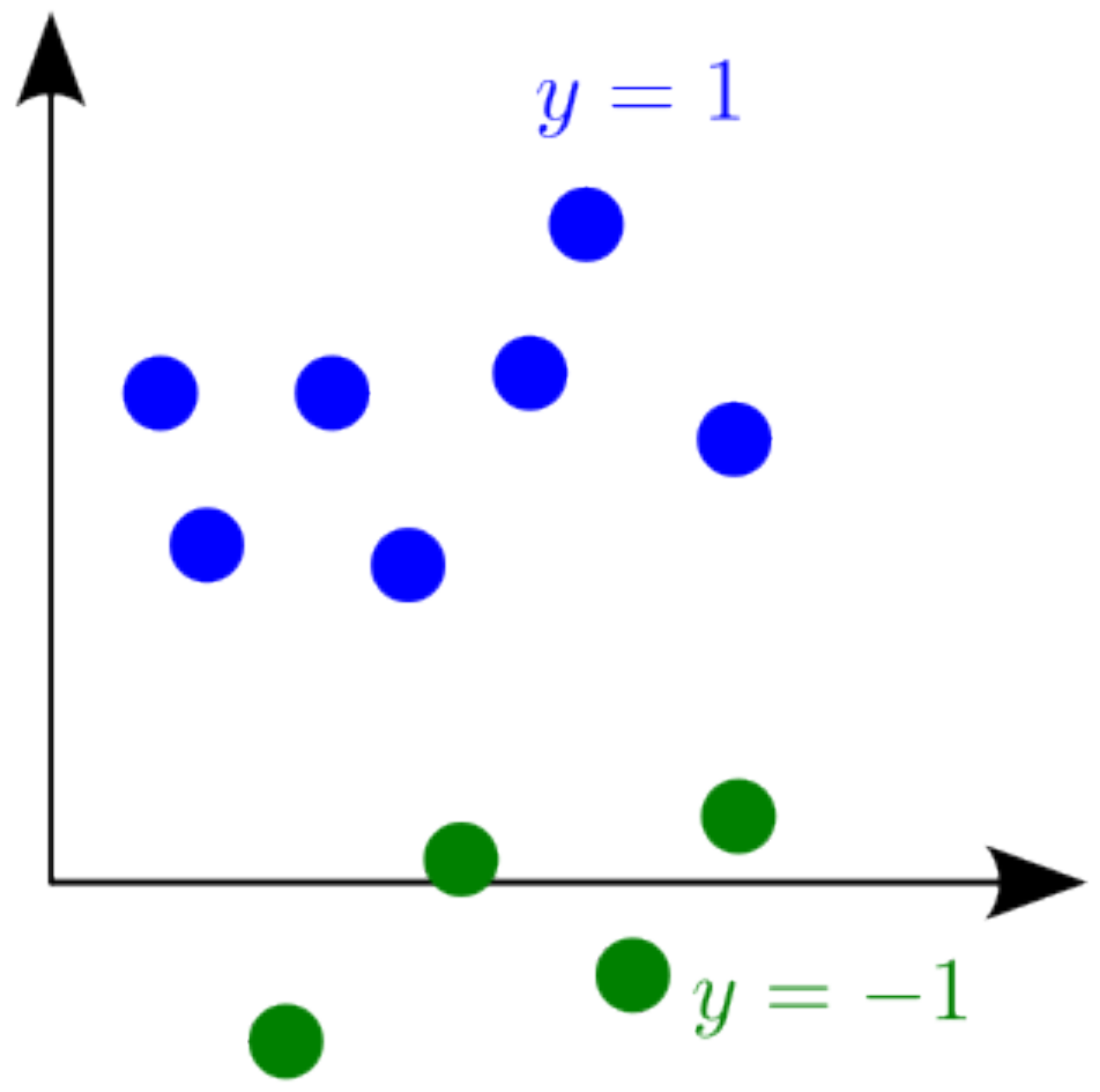
- Training a DNN on the a (labeled) data set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$:

$$\boxed{\min_{\mathbf{W}} \mathscr{L}(\mathbf{W}; S) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \mathbf{W}), y_i)}$$

- Design $f(\,\cdot\,;\mathbf{W})$ to respect group symmetry—**explicit regularization**.
- **Question**: when trained with gradient-based methods,
  - **which** solution does it converge to?
  - is it **really** better than non-equivariant models?

Implicit regularization of training algorithms

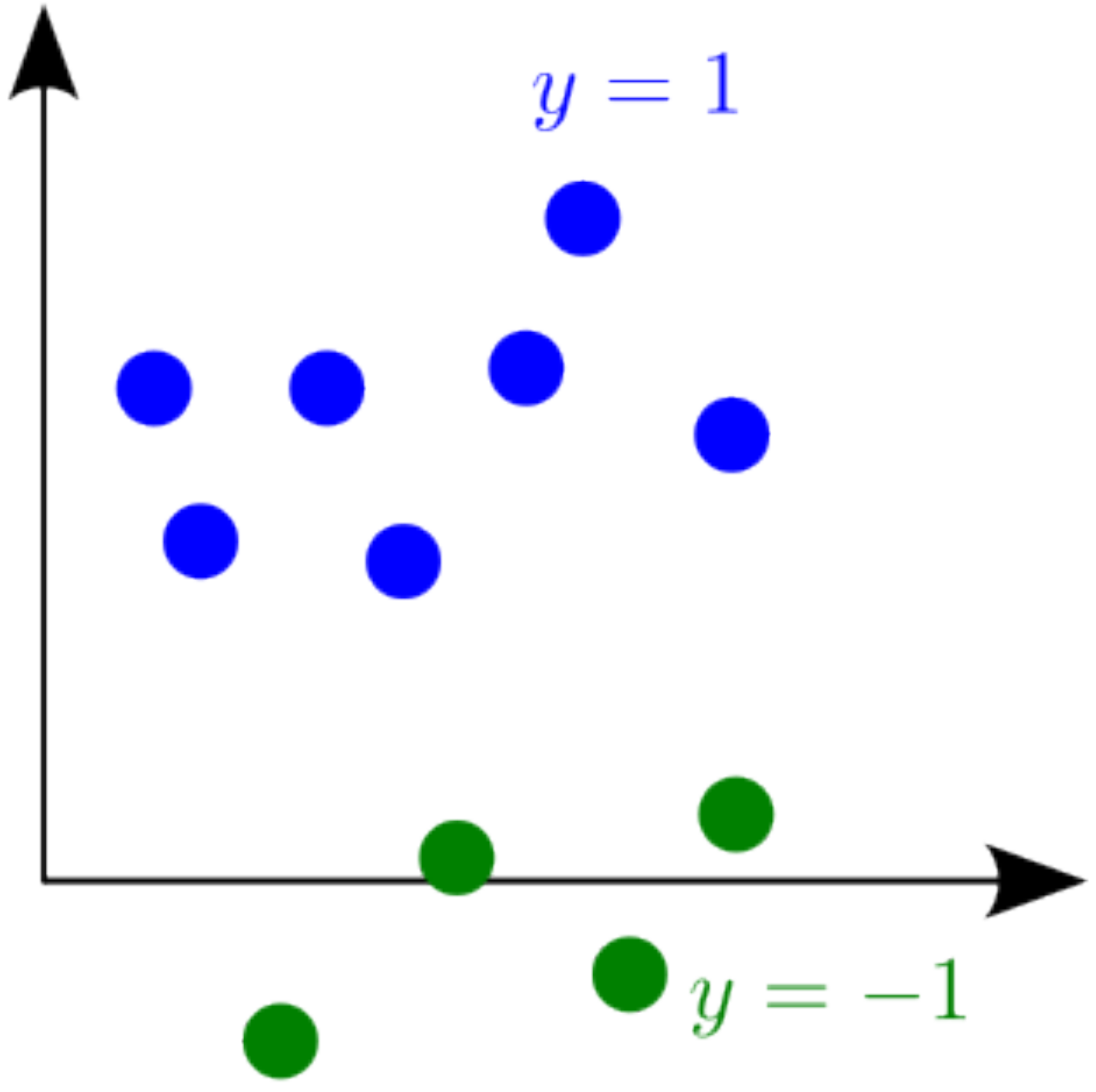
Linear binary classification



Implicit regularization of training algorithms

- $S = \{(\mathbf{x}_i, y_i) : i \in [n]\}$, $\mathbf{x}_i \in \mathbb{R}^{d_0}$ and $y_i \in \{\pm 1\}$.

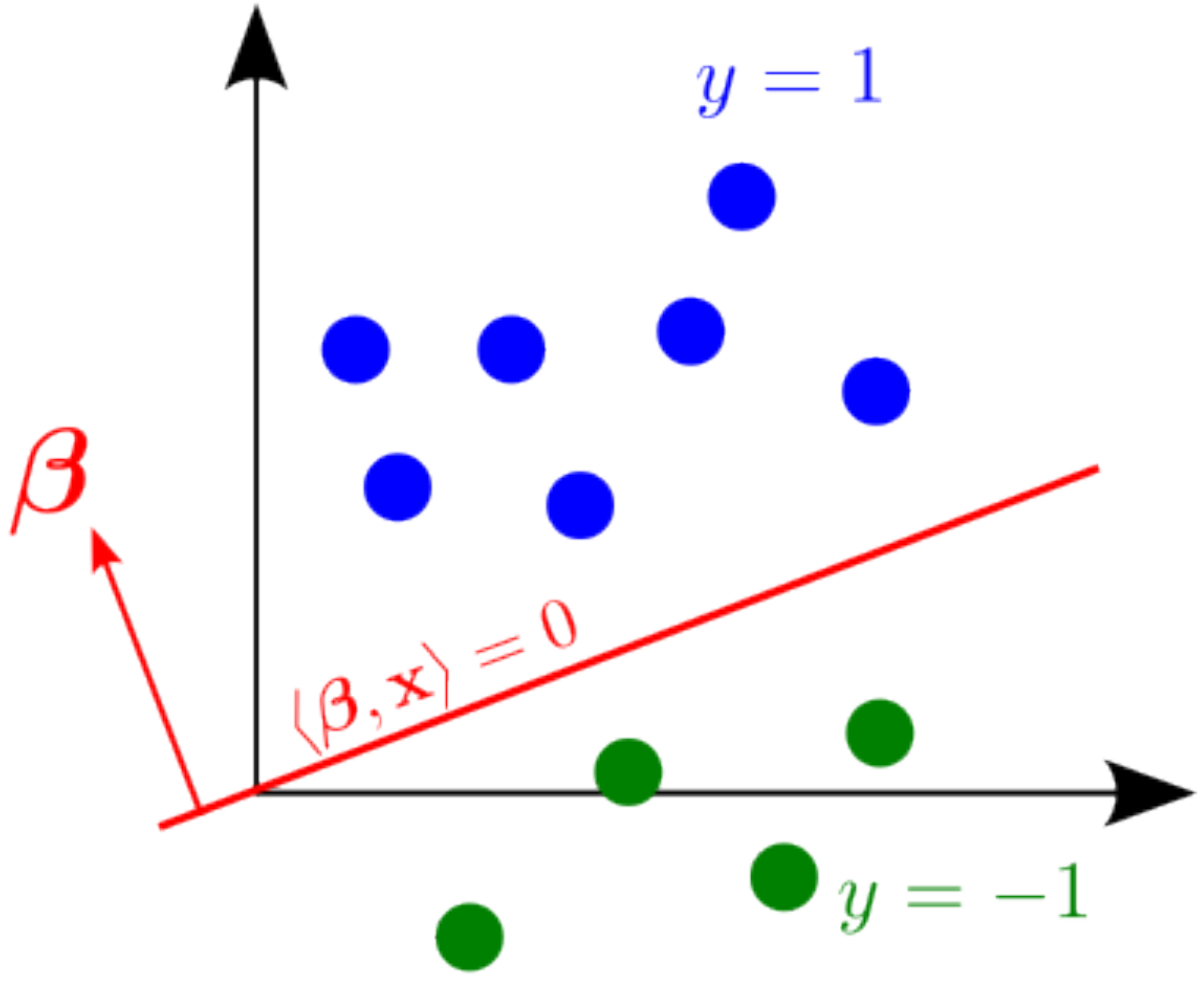
Linear binary classification



Implicit regularization of training algorithms

- $S = \{(\mathbf{x}_i, y_i) : i \in [n]\}$, $\mathbf{x}_i \in \mathbb{R}^{d_0}$ and $y_i \in \{\pm 1\}$.
- **Linearly separable**: $\exists \beta^* \in \mathbb{R}^{d_0}$, s.t $y_i \left\langle \mathbf{x}_i, \beta^* \right\rangle \geq 1, \forall i \in [n]$.

Linear binary classification



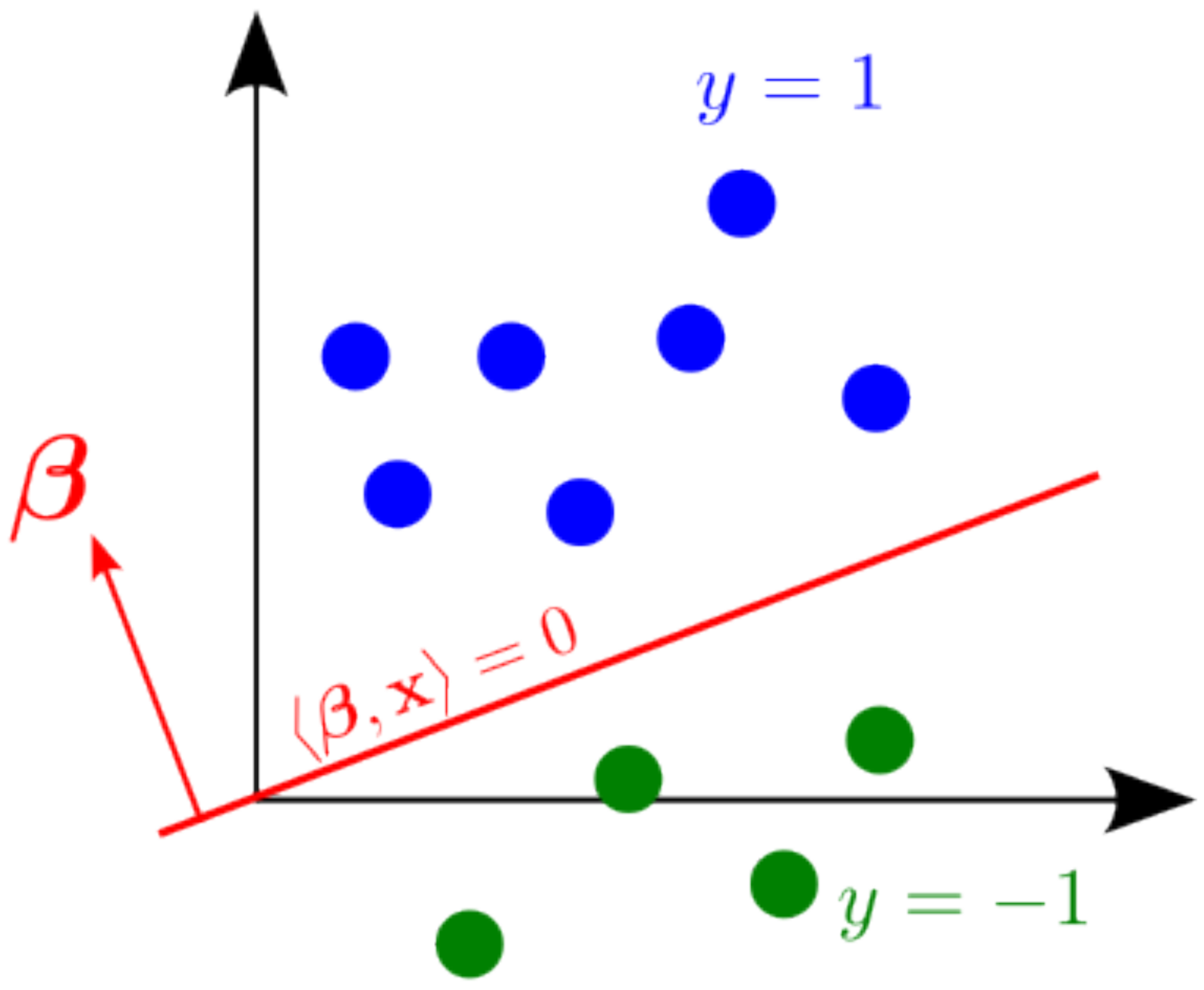
Implicit regularization of training algorithms

- $S = \{(\mathbf{x}_i, y_i) : i \in [n]\}$, $\mathbf{x}_i \in \mathbb{R}^{d_0}$ and $y_i \in \{\pm 1\}$.
- **Linearly separable**: $\exists \beta^* \in \mathbb{R}^{d_0}$, s.t $y_i \left\langle \mathbf{x}_i, \beta^* \right\rangle \geq 1, \forall i \in [n]$.
- Use **linear fully-connected (fc)** network to parameterize $\langle \mathbf{x}, \beta^* \rangle$

$$f_{\mathrm{fc}}(\mathbf{x}; \mathbf{W}) = \mathbf{w}_L^\top \mathbf{w}_{L-1}^\top \cdots \mathbf{w}_1^\top \mathbf{x} = \left\langle \mathbf{x}, \mathscr{P}_{\mathrm{fc}}(\mathbf{W}) \right\rangle \overset{?}{\approx} \langle \mathbf{x}, \beta^* \rangle$$

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L], \quad \mathscr{P}_{\mathrm{fc}}(\mathbf{W}) = \mathbf{w}_1 \mathbf{w}_2 \cdots \mathbf{w}_L$$

Linear binary classification



Implicit regularization of training algorithms

- $S = \{(\mathbf{x}_i, y_i) : i \in [n]\}$, $\mathbf{x}_i \in \mathbb{R}^{d_0}$ and $y_i \in \{\pm 1\}$.
- **Linearly separable**: $\exists \beta^* \in \mathbb{R}^{d_0}$, s.t $y_i \left\langle \mathbf{x}_i, \beta^* \right\rangle \geq 1, \forall i \in [n]$.
- Use **linear fully-connected (fc)** network to parameterize $\langle \mathbf{x}, \beta^* \rangle$

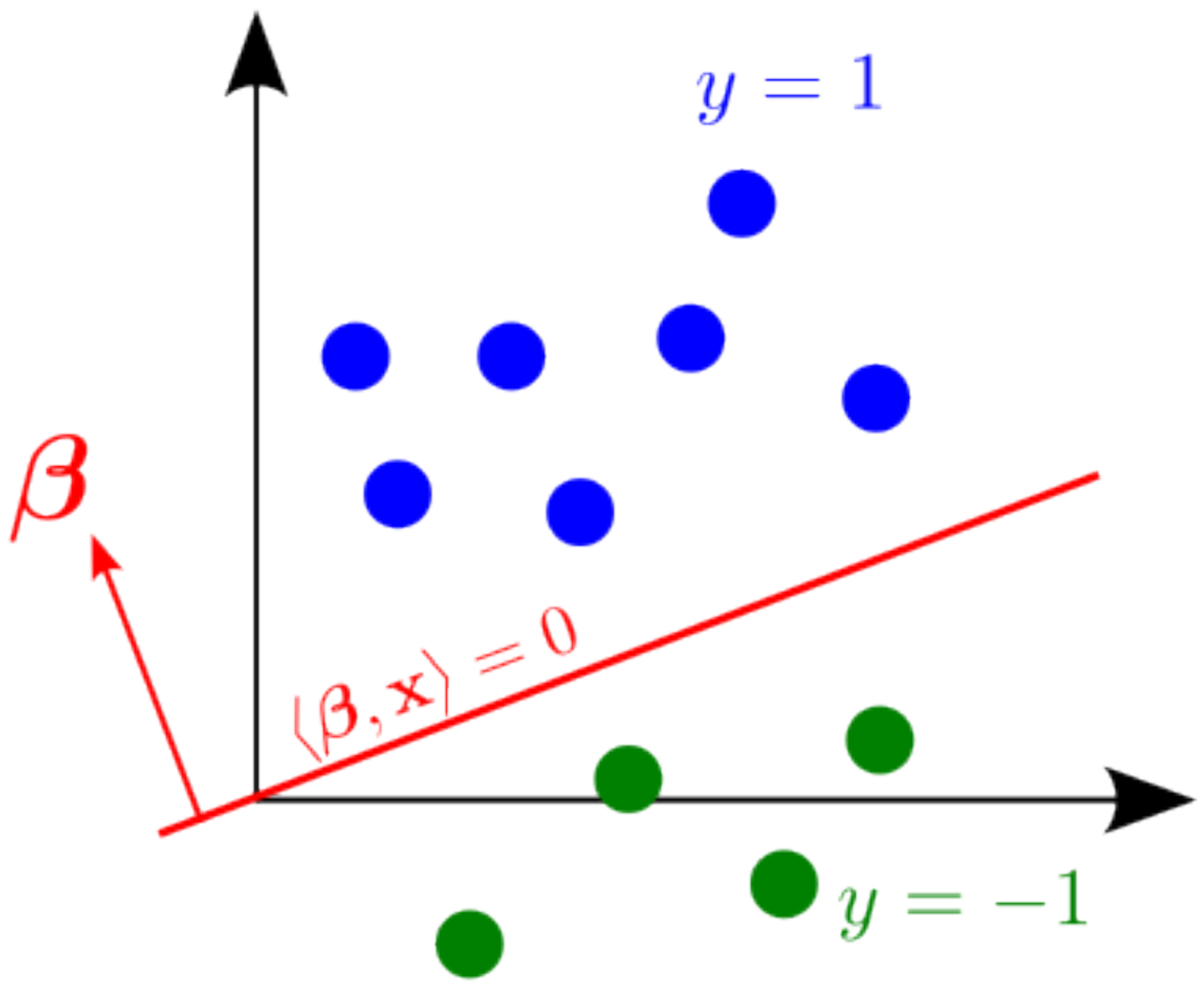
$$f_{\text{fc}}(\mathbf{x}; \mathbf{W}) = \mathbf{w}_L^\top \mathbf{w}_{L-1}^\top \cdots \mathbf{w}_1^\top \mathbf{x} = \left\langle \mathbf{x}, \mathscr{P}_{\text{fc}}(\mathbf{W}) \right\rangle \overset{?}{\approx} \langle \mathbf{x}, \beta^* \rangle$$

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L], \quad \mathscr{P}_{\text{fc}}(\mathbf{W}) = \mathbf{w}_1 \mathbf{w}_2 \cdots \mathbf{w}_L$$

- Regression based on $\ell_{\exp}(\hat{y}, y) = \exp(-\hat{y}y)$

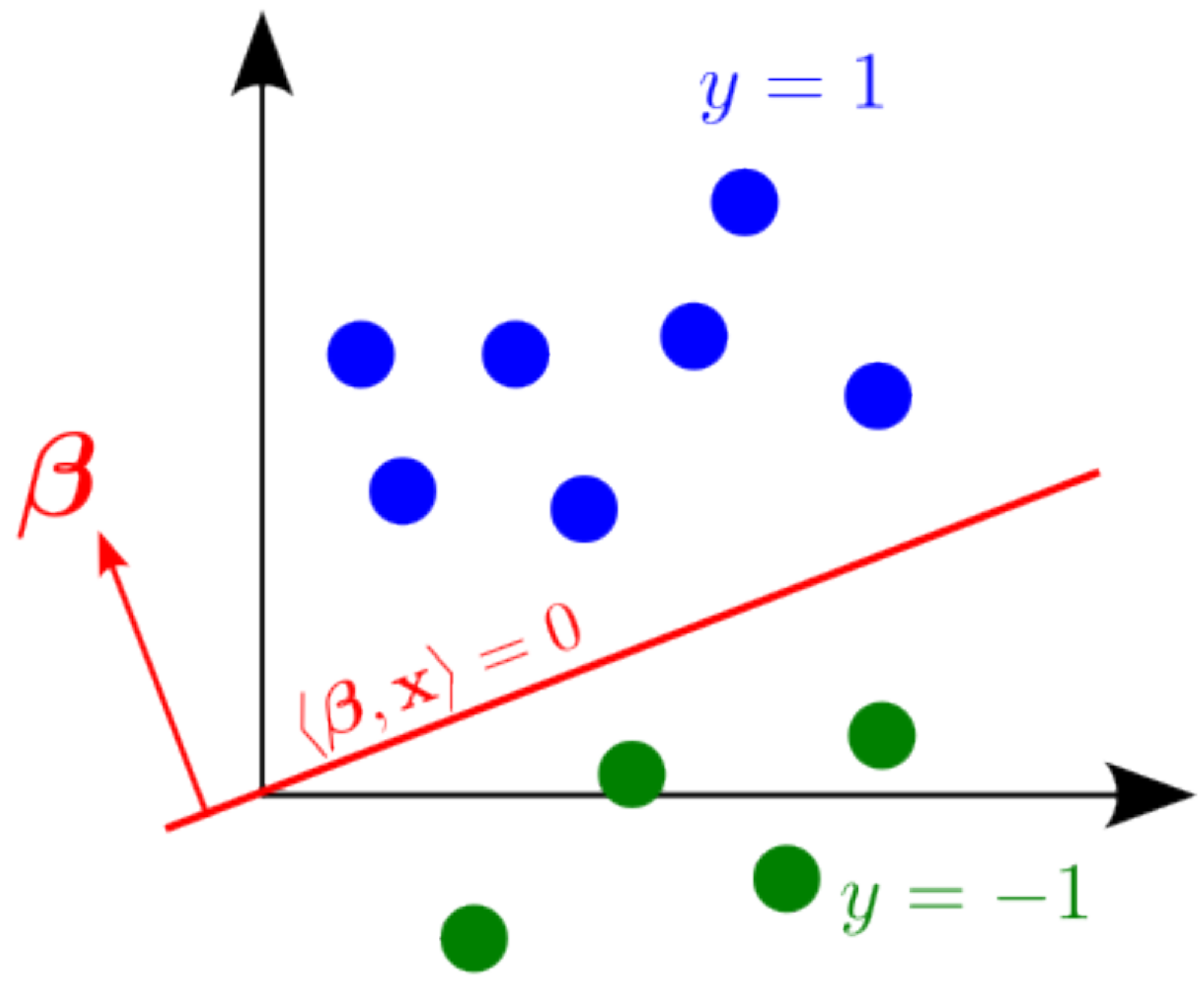
$$\min_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\text{fc}}}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell_{\exp}\left(\left\langle \mathbf{x}_i, \mathscr{P}_{\text{fc}}(\mathbf{W}) \right\rangle, y_i\right)$$

Linear binary classification



Implicit regularization of training algorithms

Linear binary classification



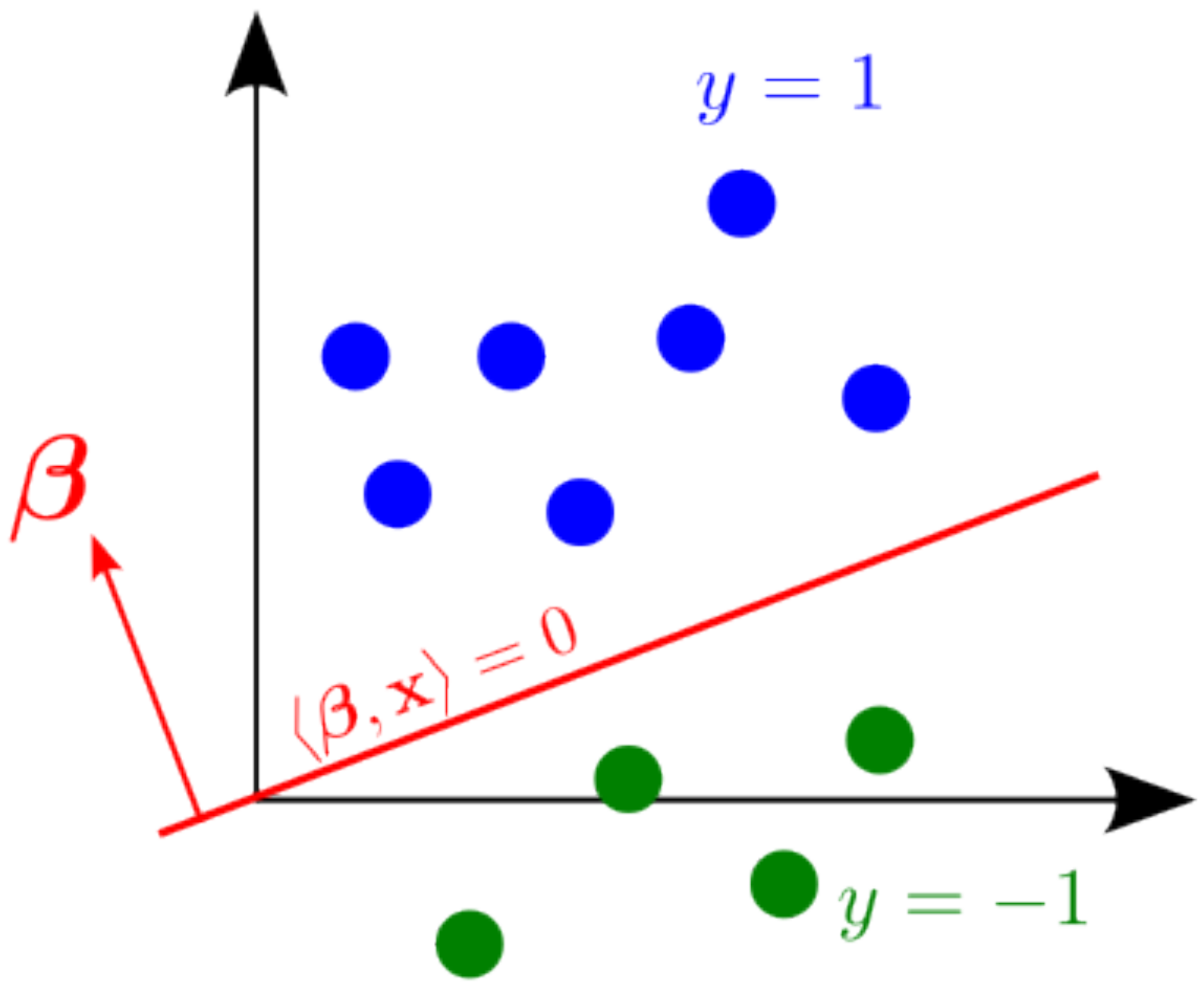
Implicit regularization of training algorithms

$$\min_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\text{fc}}}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell_{\exp}\left(\left\langle \mathbf{x}_i, \mathscr{P}_{\text{fc}}(\mathbf{W}) \right\rangle, y_i\right)$$

- Trained under **gradient flow (GF)**:

$$\frac{d\mathbf{W}}{dt} = -\nabla_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\text{fc}}}(\mathbf{W}; S)$$

Linear binary classification



Implicit regularization of training algorithms

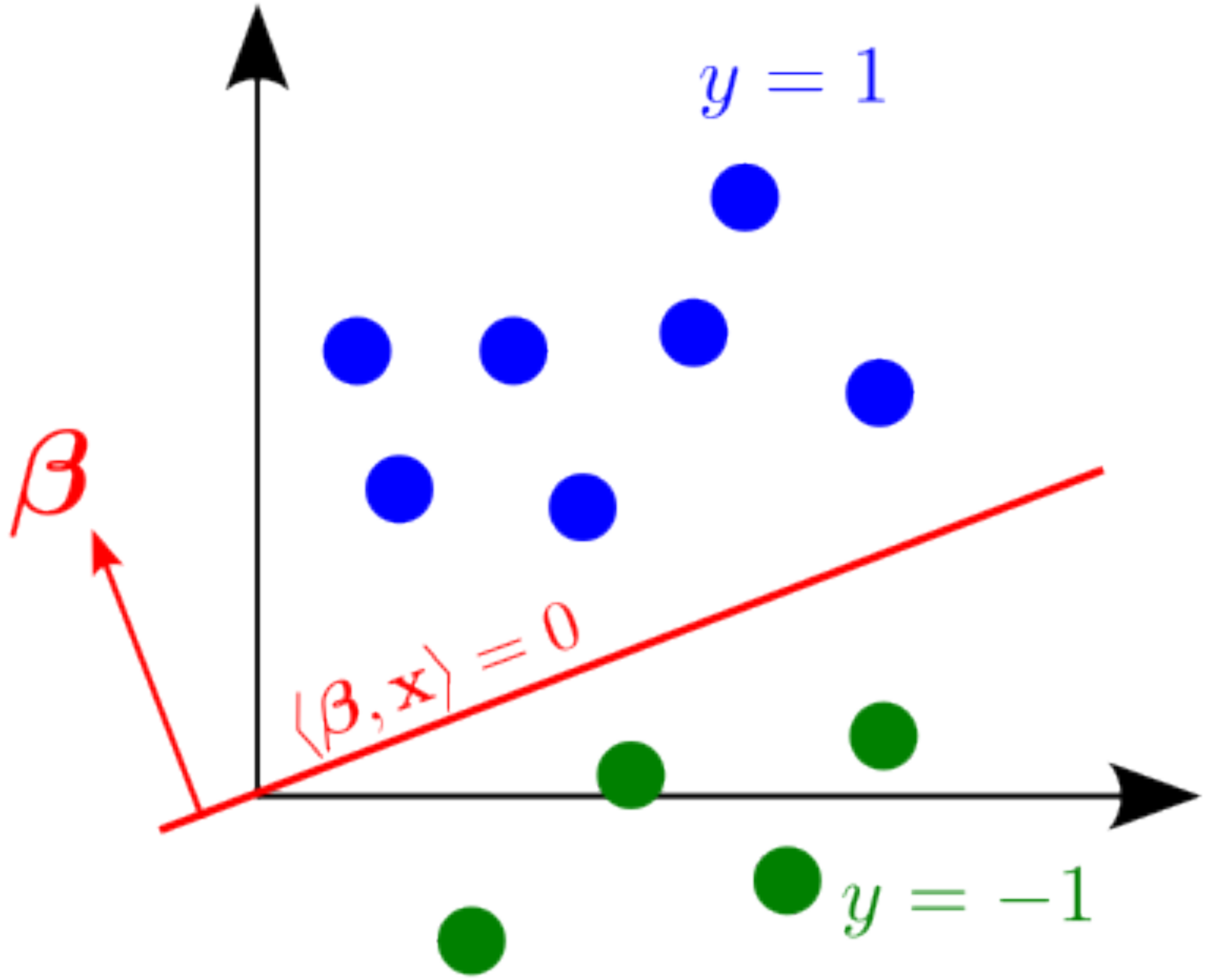
$$\min_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\mathsf{fc}}}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell_{\exp}\Big(\big\langle \mathbf{x}_i, \mathscr{P}_{\mathsf{fc}}(\mathbf{W})\big\rangle, y_i\Big)$$

- Trained under **gradient flow (GF)**:

$$\frac{d\mathbf{W}}{dt} = -\nabla_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\mathsf{fc}}}(\mathbf{W}; S)$$

Question: to what does $\beta_{\mathsf{fc}}(t) = \mathscr{P}_{\mathsf{fc}}(\mathbf{W}(t))$ converge?

Linear binary classification



Implicit regularization of training algorithms

$$\min_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\text{fc}}}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell_{\exp}\left(\left\langle \mathbf{x}_i, \mathscr{P}_{\text{fc}}(\mathbf{W}) \right\rangle, y_i\right)$$

- Trained under **gradient flow (GF)**:

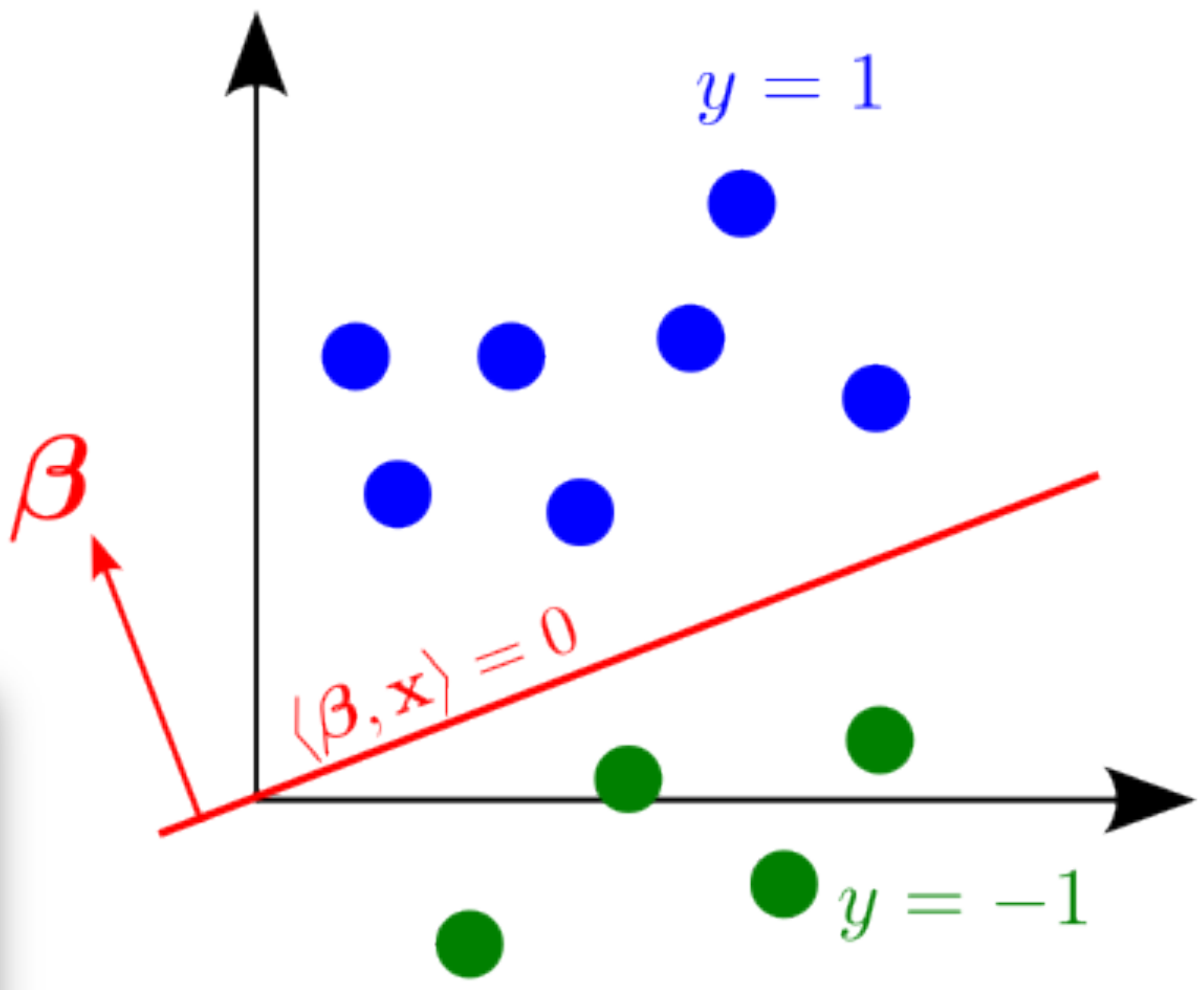
$$\frac{d\mathbf{W}}{dt} = -\nabla_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\text{fc}}}(\mathbf{W}; S)$$

Question: to what does $\beta_{\text{fc}}(t) = \mathscr{P}_{\text{fc}}(\mathbf{W}(t))$ converge?

Fact [Ji and Telgarsky, *ICLR* 2018], [Yun et al., *ICLR* 2021]

- $\beta_{\text{fc}}^{\infty} = \lim_{t \to \infty} \beta_{\text{fc}}(t) / \|\beta_{\text{fc}}(t)\|$ exists.

Linear binary classification



Implicit regularization of training algorithms

$$\min_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\mathsf{fc}}}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell_{\exp}\left(\langle \mathbf{x}_i, \mathscr{P}_{\mathsf{fc}}(\mathbf{W}) \rangle, y_i\right)$$

- Trained under **gradient flow (GF)**:

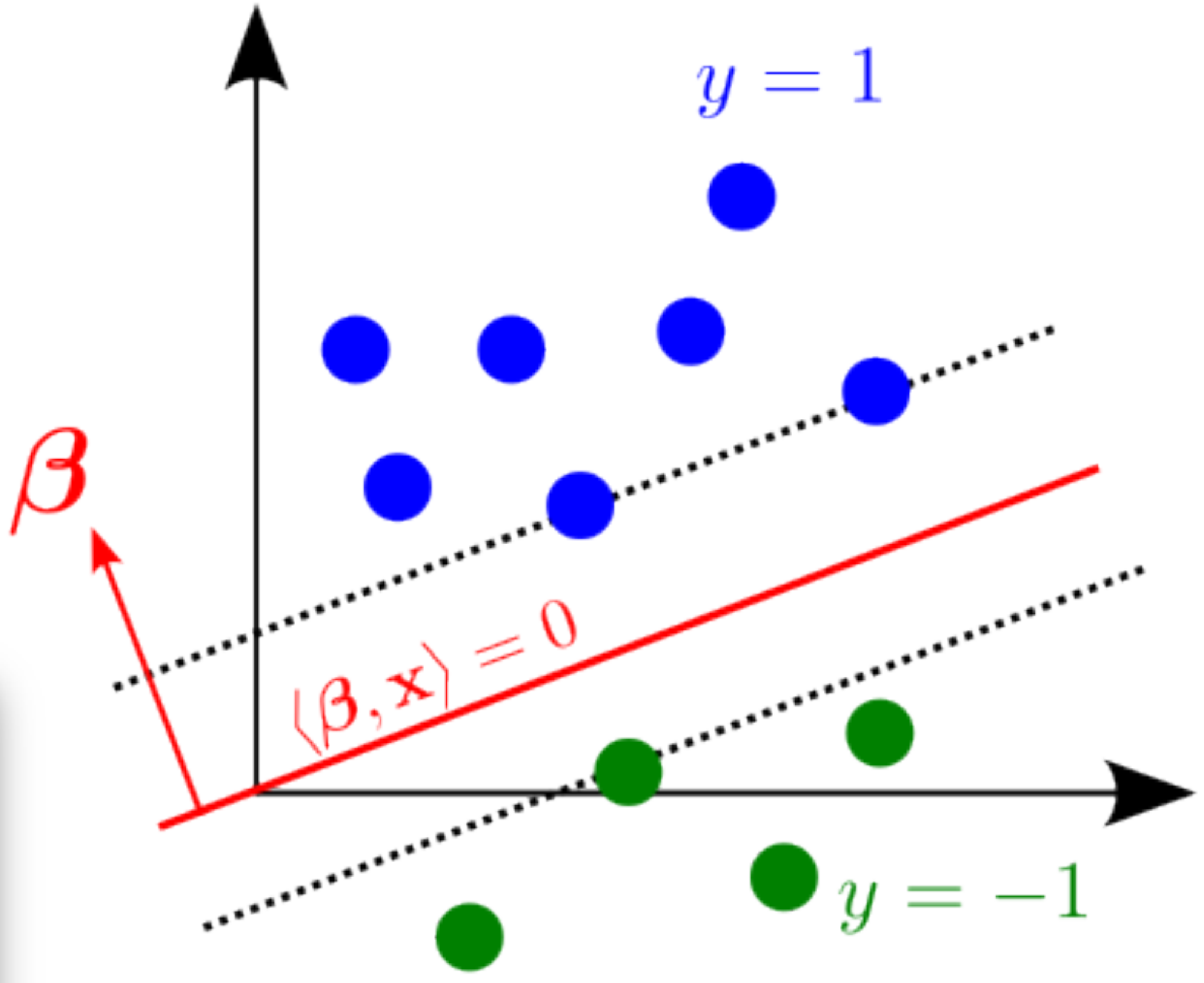
$$\frac{d\mathbf{W}}{dt} = -\nabla_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\mathsf{fc}}}(\mathbf{W}; S)$$

Question: to what does $\beta_{\mathsf{fc}}(t) = \mathscr{P}_{\mathsf{fc}}(\mathbf{W}(t))$ converge?

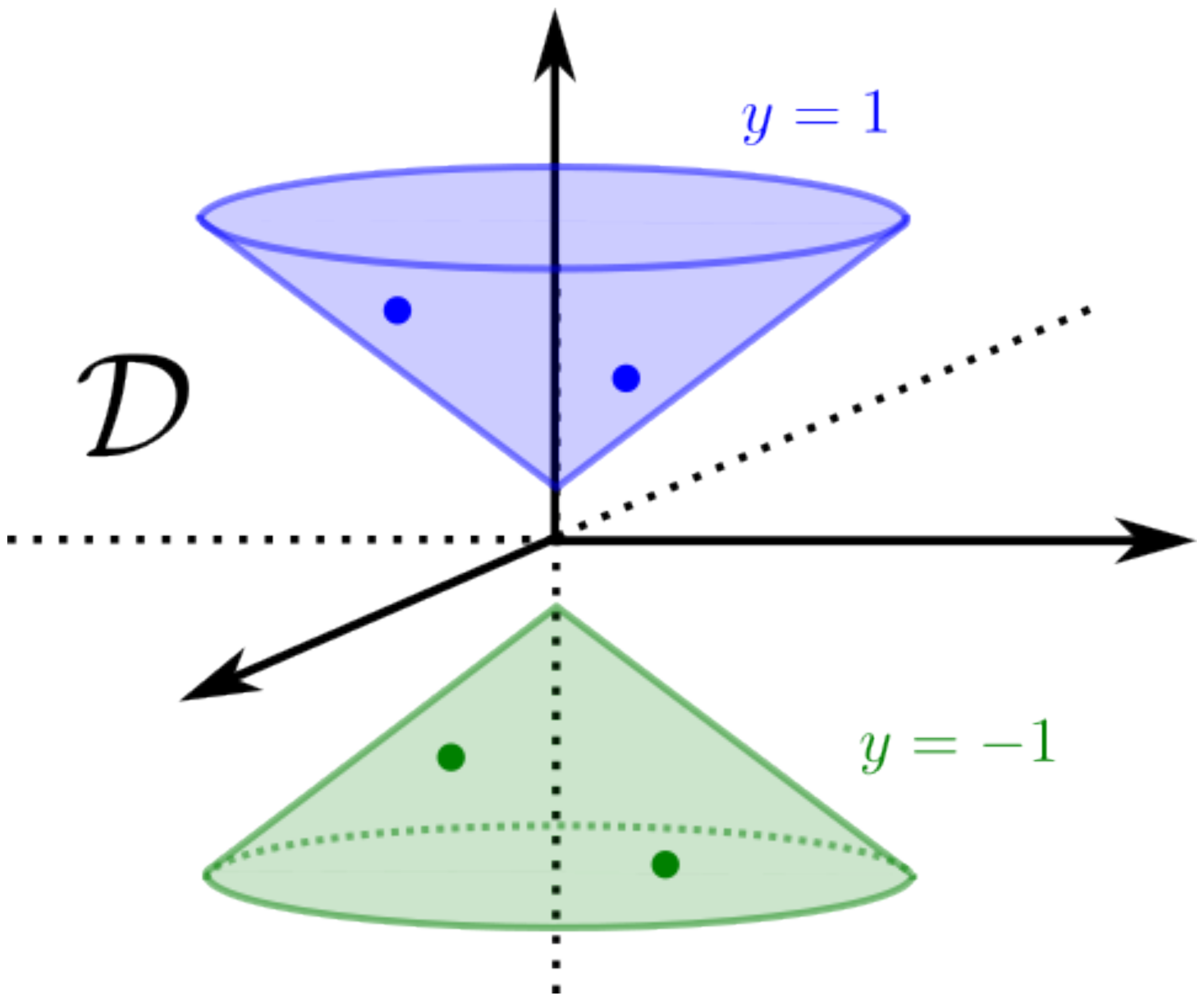
Fact [Ji and Telgarsky, *ICLR* 2018], [Yun et al., *ICLR* 2021]

- $\beta_{\mathsf{fc}}^{\infty} = \lim_{t\to\infty} \beta_{\mathsf{fc}}(t)/\|\beta_{\mathsf{fc}}(t)\|$ exists.
- $\beta_{\mathsf{fc}}^{\infty}$ is the the **max-L^2-margin** support vector machine (SVM).

Linear binary classification

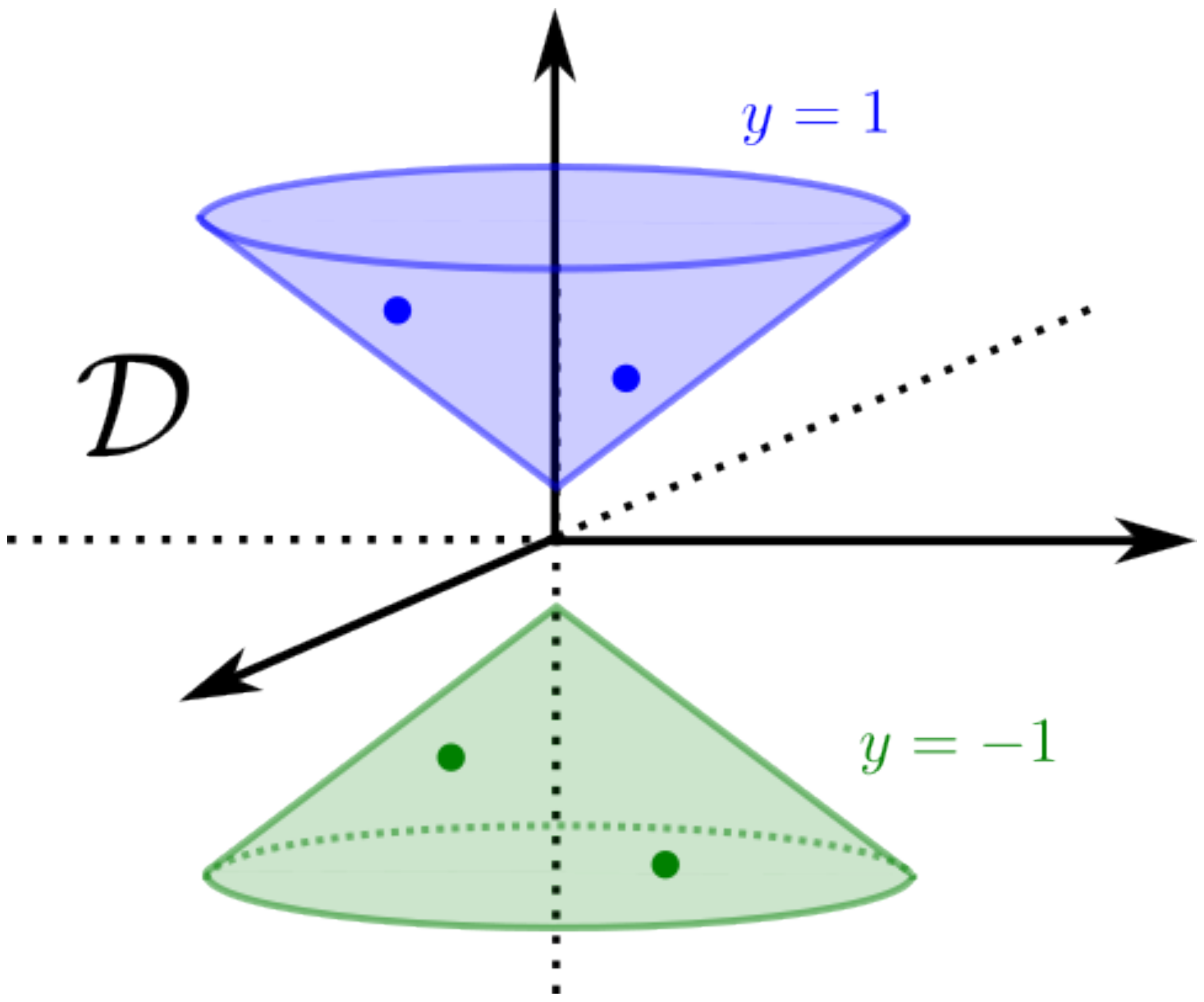


Group-invariant binary classification



Group-invariant binary classification

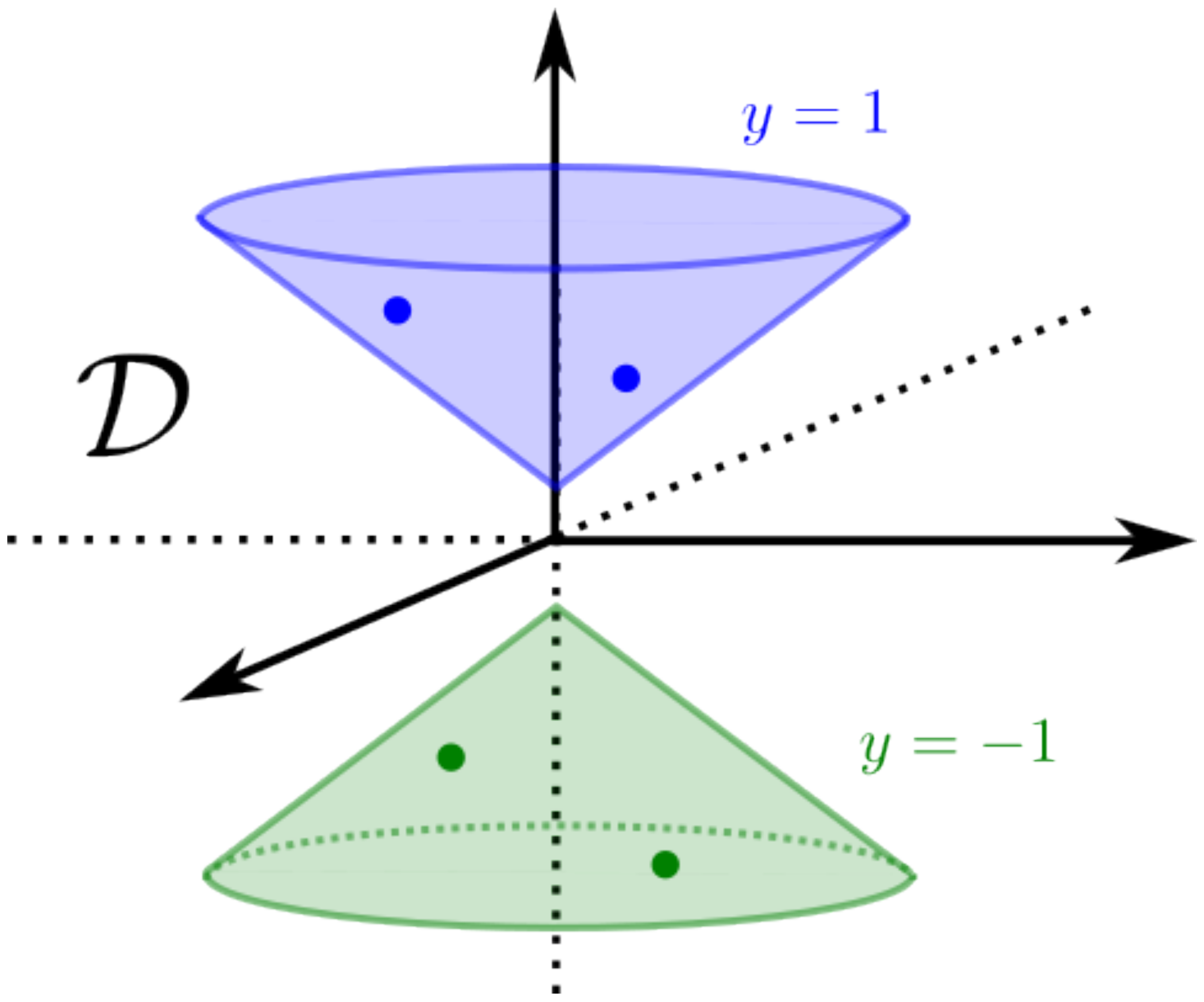
- Assume $S \sim \mathscr{D}$, and $\mathscr{D}$ is **invariant** to a linear G-action.



Group-invariant binary classification

- Assume $S \sim \mathscr{D}$, and $\mathscr{D}$ is **invariant** to a linear G-action.
- Parameterize the **invariant** linear predictor β using a **G-CNN**,

$$f_{\text{inv}}(\mathbf{x}; \mathbf{W}) = \left\langle \mathbf{x}, \mathscr{P}_{\text{inv}}(\mathbf{W}) \right\rangle$$

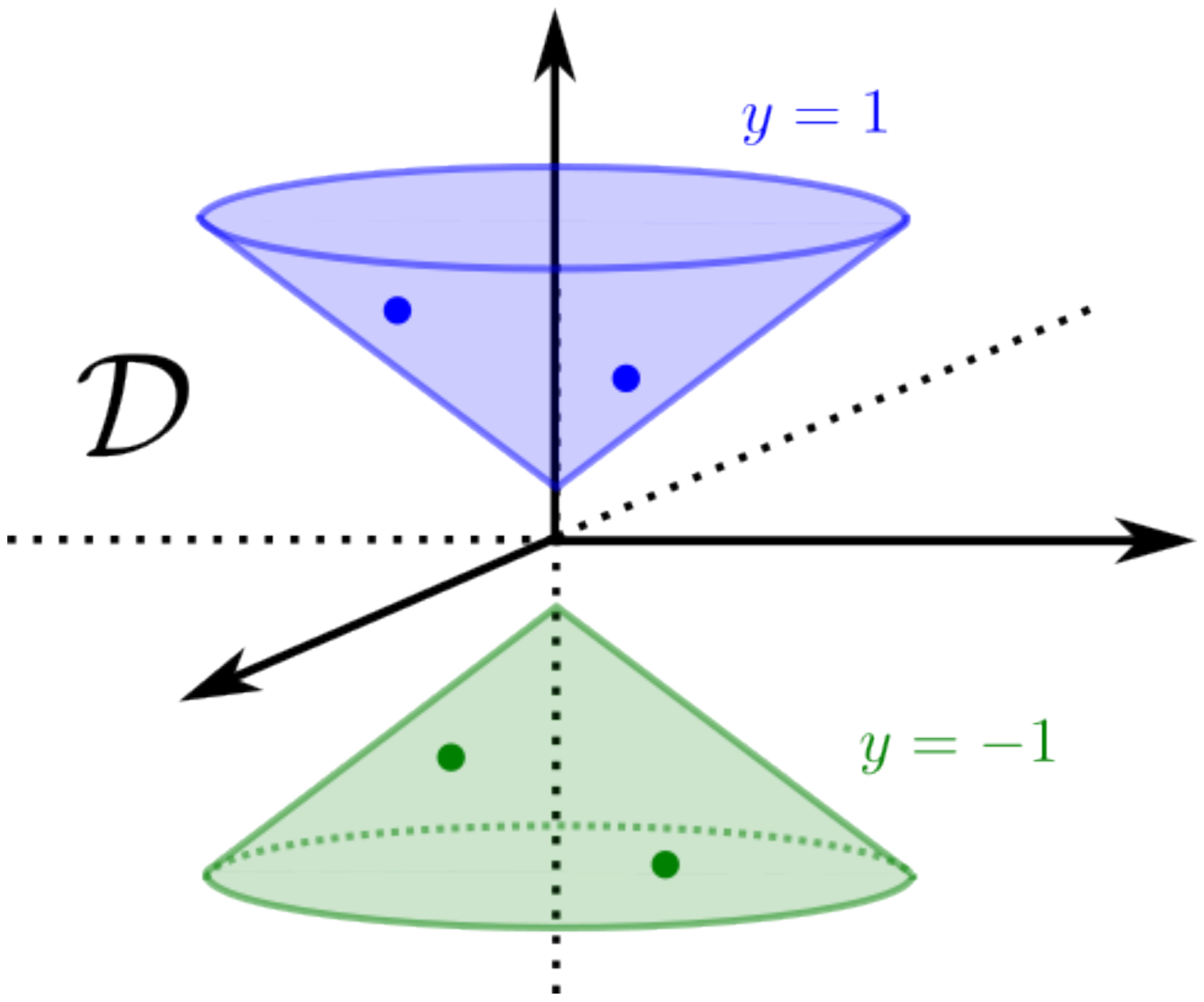


Group-invariant binary classification

- Assume $S \sim \mathscr{D}$, and $\mathscr{D}$ is **invariant** to a linear G-action.
- Parameterize the **invariant** linear predictor β using a **G-CNN**,

$$f_{\mathrm{inv}}(\mathbf{x}; \mathbf{W}) = \left\langle \mathbf{x}, \mathscr{P}_{\mathrm{inv}}(\mathbf{W}) \right\rangle$$

- Regression: $\min_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\mathrm{inv}}}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell_{\exp}\left(\left\langle \mathbf{x}_i, \mathscr{P}_{\mathrm{inv}}(\mathbf{W}) \right\rangle, y_i\right)$
- Gradient flow: $\dfrac{d\mathbf{W}}{dt} = -\nabla_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\mathrm{inv}}}(\mathbf{W}; S)$



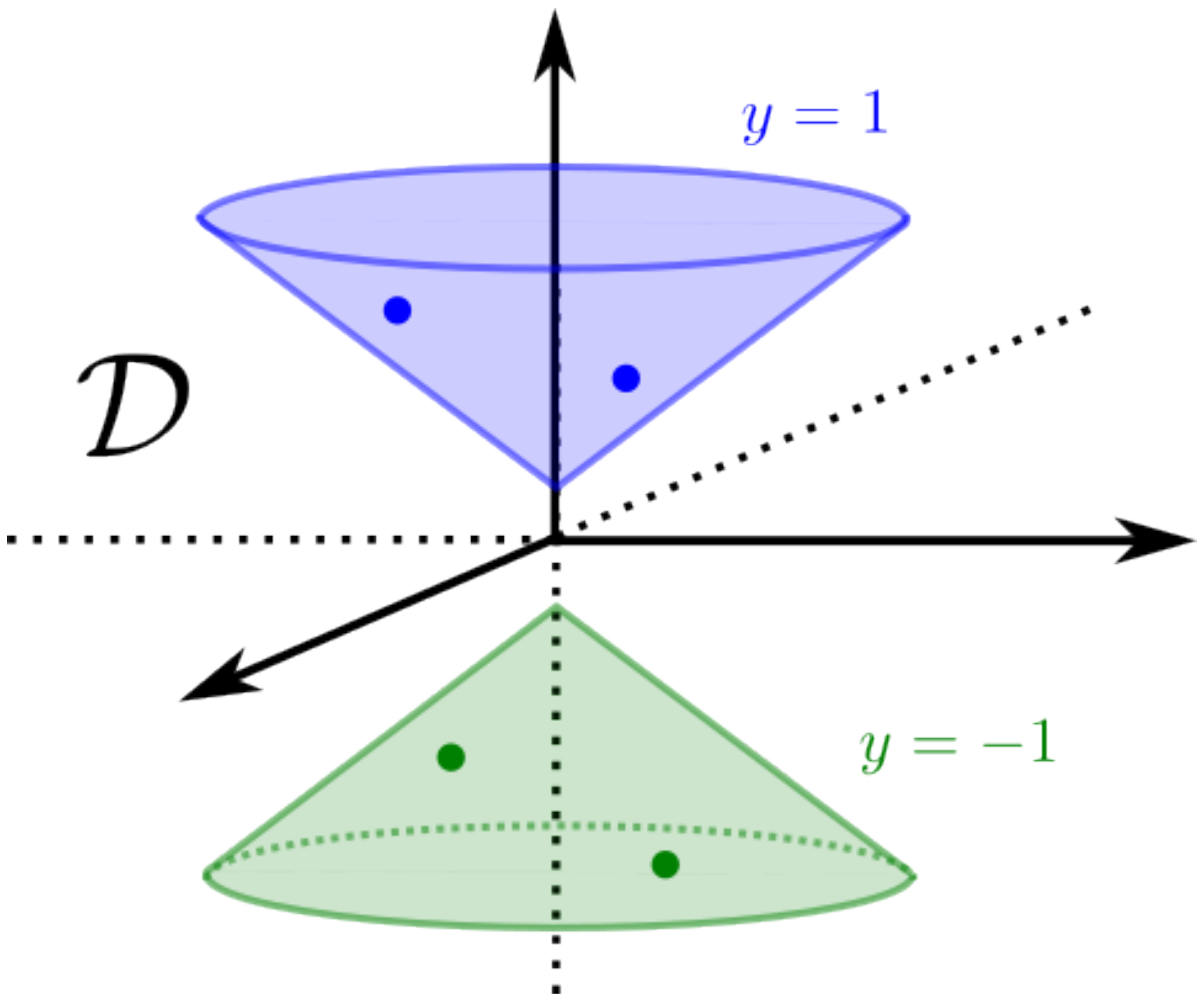
Group-invariant binary classification

- Assume $S \sim \mathscr{D}$, and $\mathscr{D}$ is **invariant** to a linear G-action.
- Parameterize the **invariant** linear predictor β using a **G-CNN**,

$$f_{\text{inv}}(\mathbf{x}; \mathbf{W}) = \left\langle \mathbf{x}, \mathscr{P}_{\text{inv}}(\mathbf{W}) \right\rangle$$

- Regression: $\min_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\text{inv}}}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell_{\exp}\left(\left\langle \mathbf{x}_i, \mathscr{P}_{\text{inv}}(\mathbf{W}) \right\rangle, y_i\right)$
- Gradient flow: $\dfrac{d\mathbf{W}}{dt} = -\nabla_{\mathbf{W}} \mathscr{L}_{\mathscr{P}_{\text{inv}}}(\mathbf{W}; S)$

Question: to what does $\beta_{\text{inv}}(t) = \mathscr{P}_{\text{inv}}(\mathbf{W}(t))$ converge?



Group-invariant binary classification

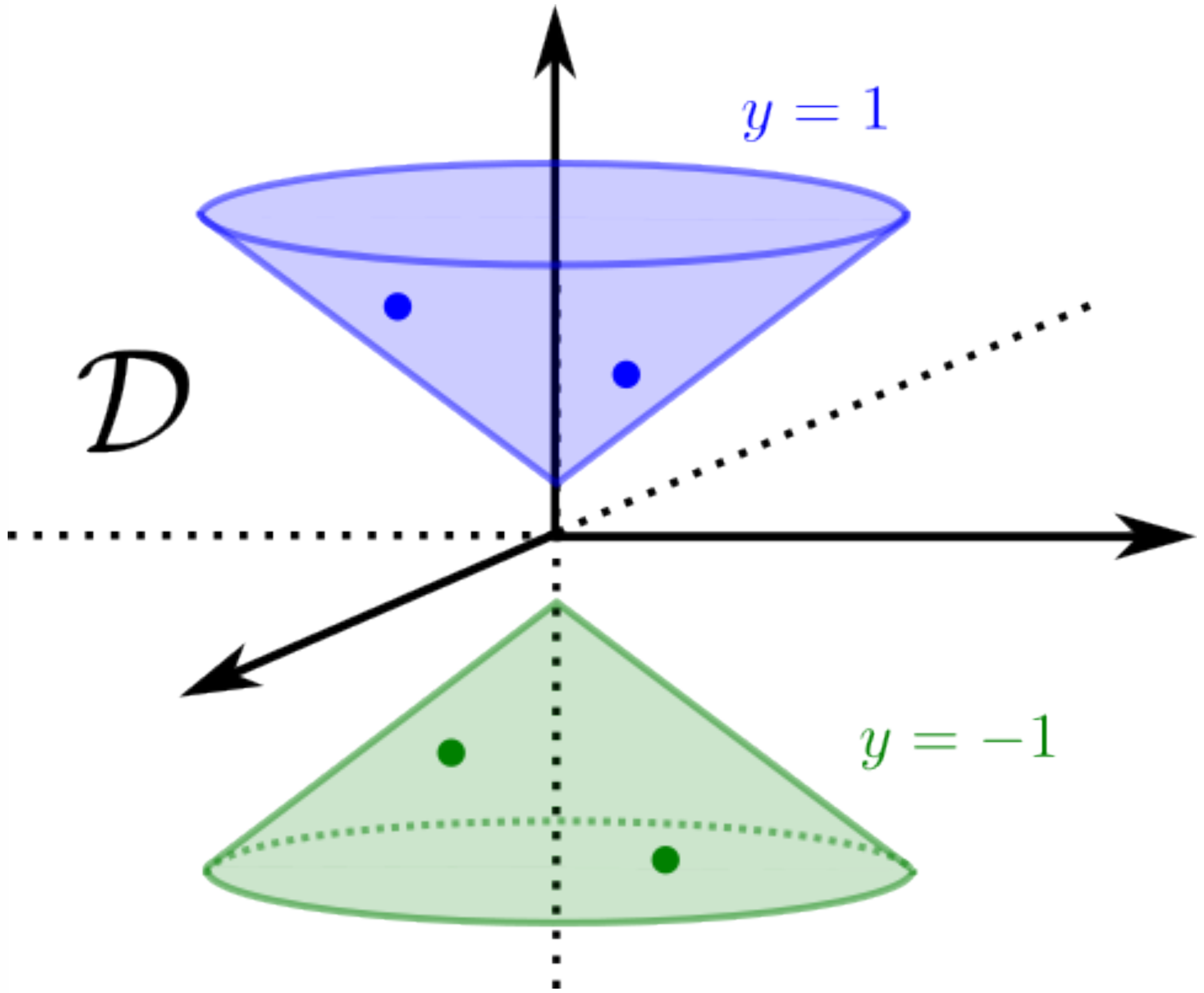
Group-invariant binary classification

Question: to what does $\beta_{\text{inv}}(t) = \mathscr{P}_{\text{inv}}(\mathbf{W}(t))$ converge?

Theorem [Chen and **Z.**, *NeurIPS* 2023]

If the input linear G-action is unitary, then

- $\beta_{\text{inv}}^{\infty} = \lim_{t\to\infty} \beta_{\text{inv}}(t)/\|\beta_{\text{inv}}(t)\|$ exists.



Group-invariant binary classification

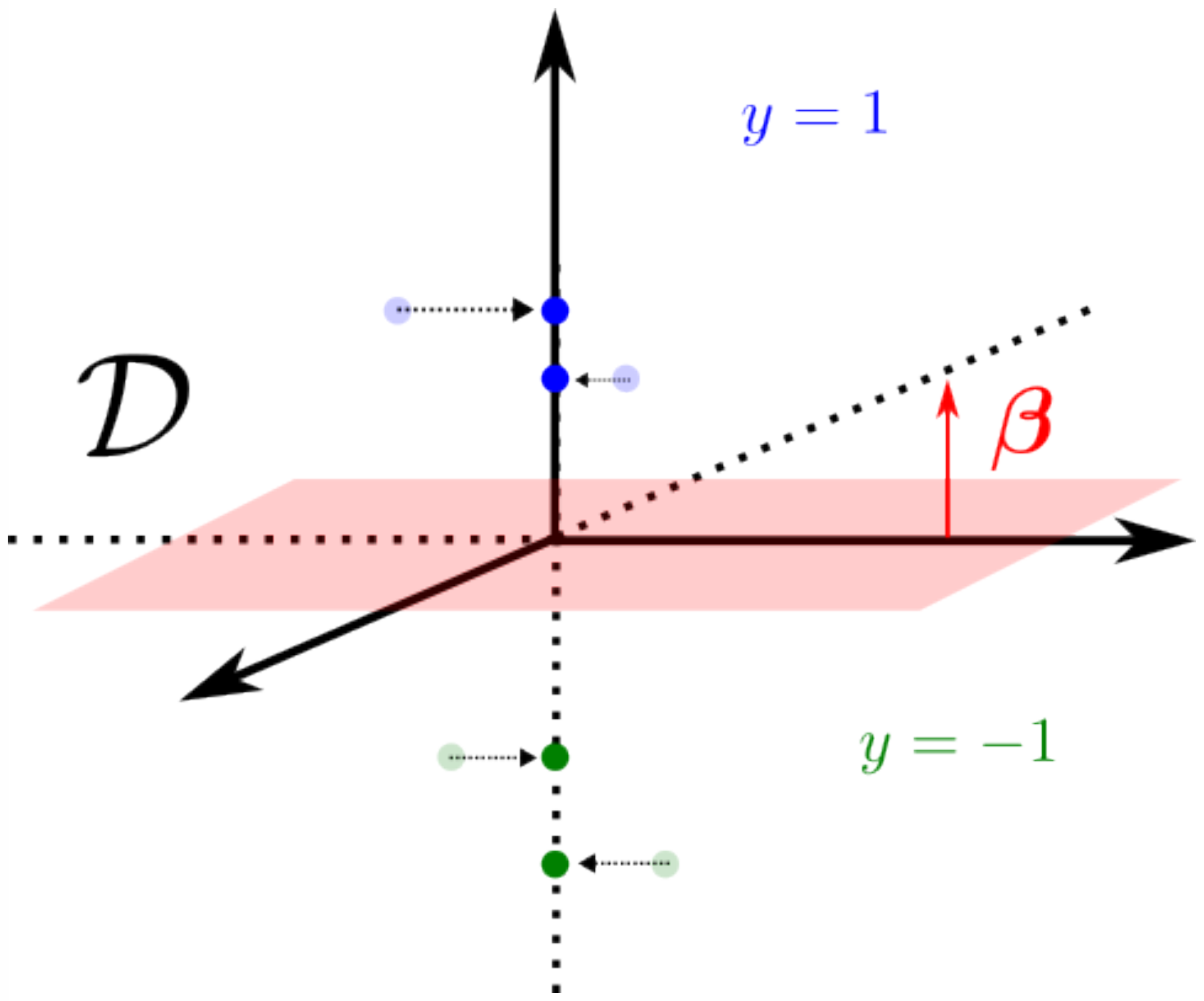
Question: to what does $\beta_{\text{inv}}(t) = \mathscr{P}_{\text{inv}}(\mathbf{W}(t))$ converge?

Theorem [Chen and **Z.**, *NeurIPS* 2023]

If the input linear G-action is unitary, then

- $\beta_{\text{inv}}^{\infty} = \lim_{t\to\infty} \beta_{\text{inv}}(t)/\|\beta_{\text{inv}}(t)\|$ exists.
- $\beta_{\text{inv}}^{\infty}$ is the the max-margin SVM on the **transformed dataset**

$$\overline{S} = \{(\overline{\mathbf{x}}_i, y_i) : i \in [n]\}, \text{ where } \overline{\mathbf{x}} = \frac{1}{|G|}\sum_{g\in G} g\mathbf{x}$$



Group-invariant binary classification

Question: to what does $\beta_{\text{inv}}(t) = \mathscr{P}_{\text{inv}}(\mathbf{W}(t))$ converge?

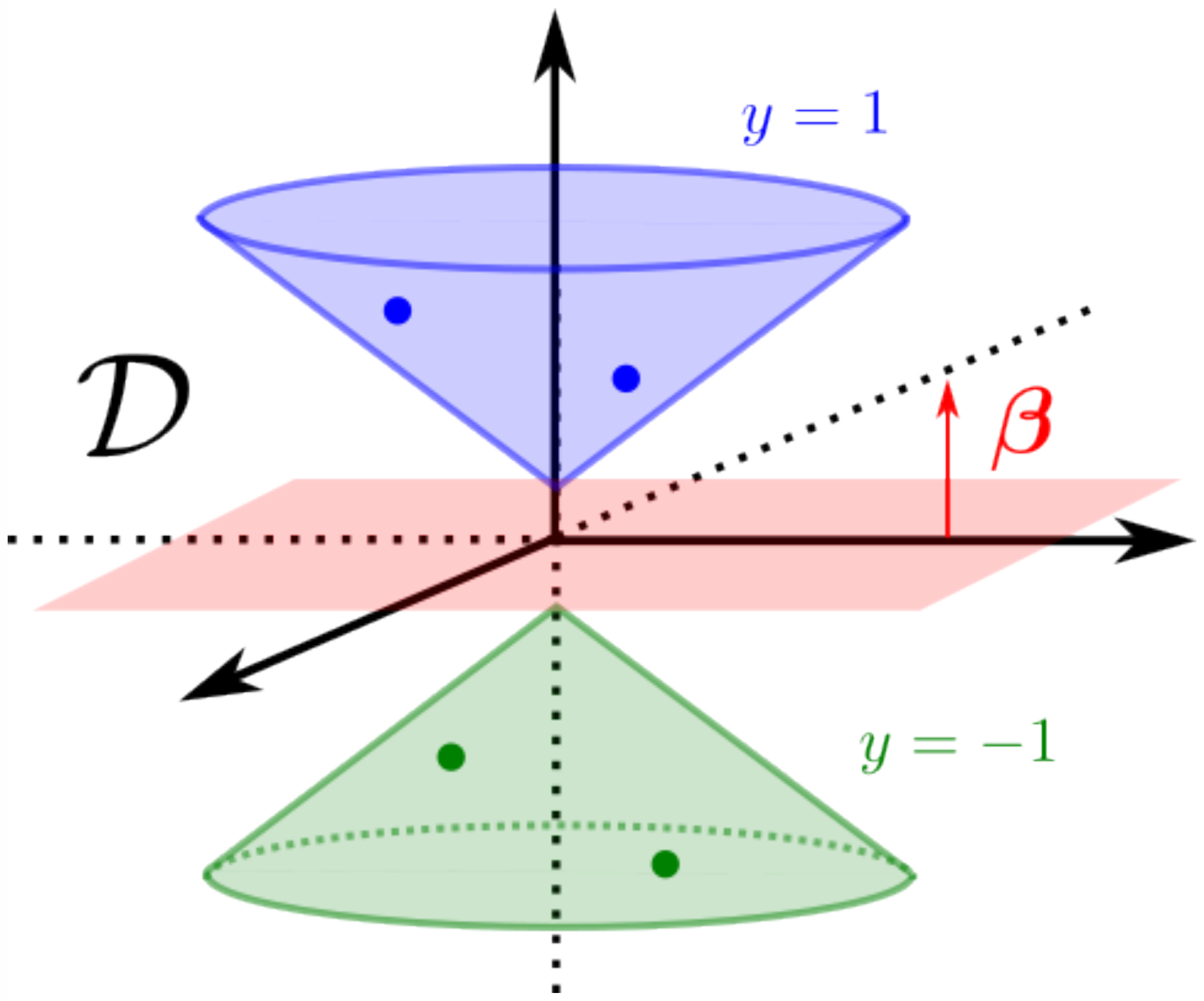
Theorem [Chen and **Z.**, *NeurIPS* 2023]

If the input linear G-action is unitary, then

- $\beta_{\text{inv}}^{\infty} = \lim_{t\to\infty} \beta_{\text{inv}}(t)/\|\beta_{\text{inv}}(t)\|$ exists.
- $\beta_{\text{inv}}^{\infty}$ is the the max-margin SVM on the **transformed dataset**

$$\bar{S} = \{(\bar{\mathbf{x}}_i, y_i) : i \in [n]\}, \text{ where } \bar{\mathbf{x}} = \frac{1}{|G|}\sum_{g\in G} g\mathbf{x}$$

- $\beta_{\text{inv}}^{\infty}$ is the unique max-margin invariant SVM on the **original dataset** $S = \{(\mathbf{x}_i, y_i) : i \in [n]\}$



Implications

Implications

Corollary (G-CNN vs data augmentation)

- $\beta^{\infty}_{\text{inv}}$: linear **G-CNN** trained on $S = \{(\mathbf{x}_i, y_i) : i \in [n]\}$.
- $\beta^{\infty}_{\text{fc}}$: linear fully-connected network trained on $S_{\text{aug}} = \{(g\mathbf{x}_i, y_i) : i \in [n], g \in G\}$.

$$\beta^{\infty}_{\text{steer}} = \beta^{\infty}_{\text{fc}}$$

Implications

Corollary (G-CNN vs data augmentation)

- $\beta_{\text{inv}}^{\infty}$: linear **G-CNN** trained on $S = \{(\mathbf{x}_i, y_i) : i \in [n]\}$.

- $\beta_{\text{fc}}^{\infty}$: linear fully-connected network trained on $S_{\text{aug}} = \{(g\mathbf{x}_i, y_i) : i \in [n], g \in G\}$.

$$\beta_{\text{steer}}^{\infty} = \beta_{\text{fc}}^{\infty}$$

- Equivariant neural networks is **equivalent** to data augmentation.

Implications

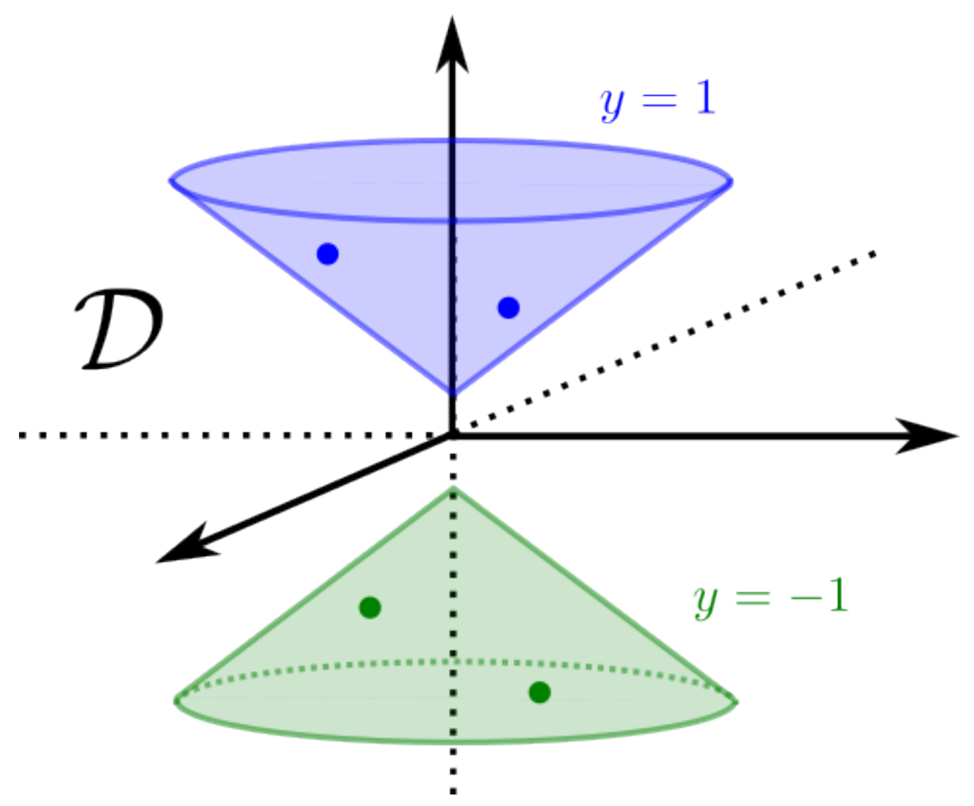
Corollary (G-CNN vs data augmentation)

- $\beta^{\infty}_{\text{inv}}$: linear **G-CNN** trained on $S = \{(\mathbf{x}_i, y_i) : i \in [n]\}$.
- $\beta^{\infty}_{\text{fc}}$: linear fully-connected network trained on $S_{\text{aug}} = \{(g\mathbf{x}_i, y_i) : i \in [n], g \in G\}$.

$$\beta^{\infty}_{\text{steer}} = \beta^{\infty}_{\text{fc}}$$

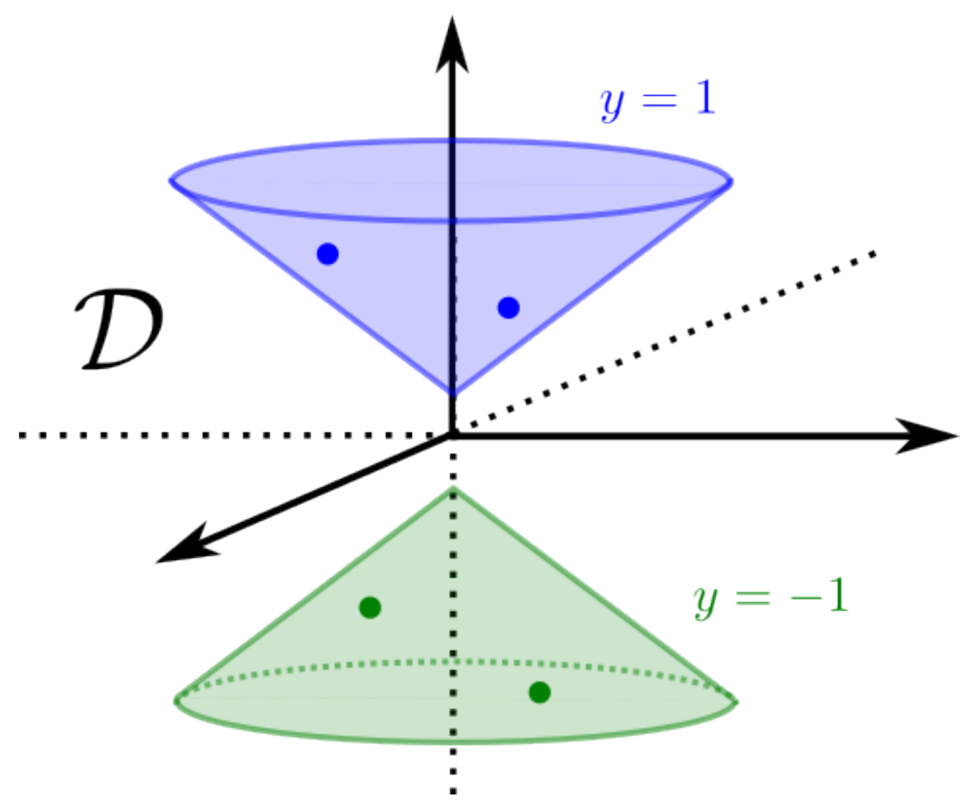
- Equivariant neural networks is **equivalent** to data augmentation.
- *Caveat*:
  - Full data augmentation on the **entire** group G.
  - **Unitary** input action.
  - Only linear models.

Improved generalization



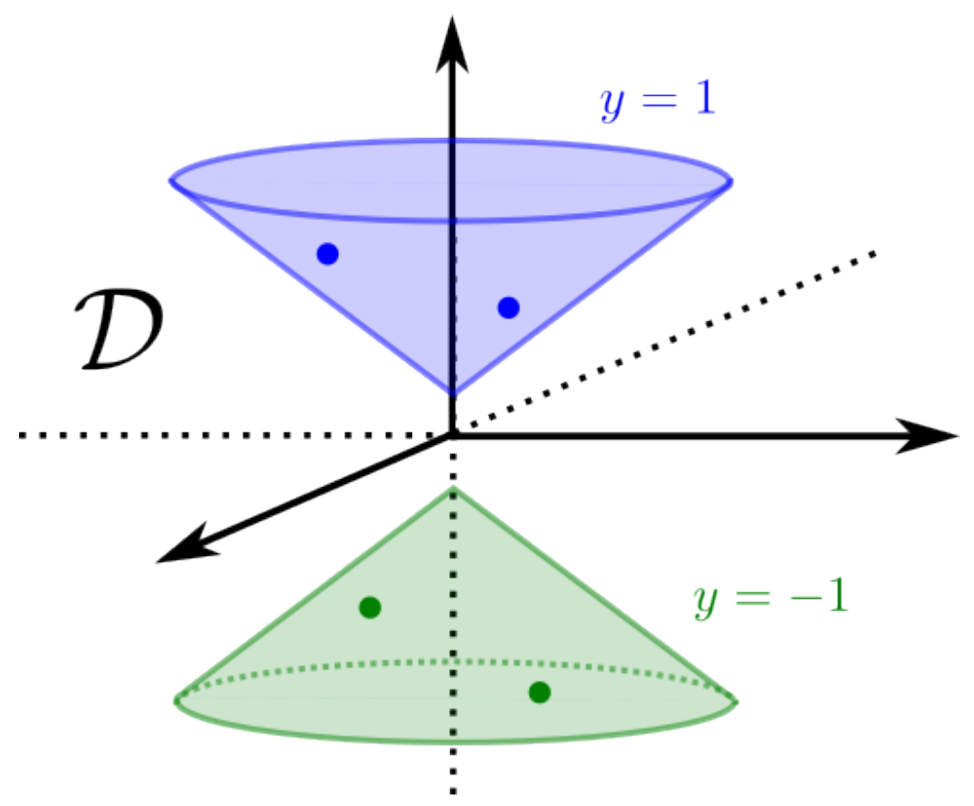
Improved generalization

- G-invariant distribution $\mathscr{D}$ over $\mathbb{R}^{d_0} \times \{\pm 1\}$.
- $\mathscr{D}$ can be separated by an invariant classifier β_0.



Improved generalization

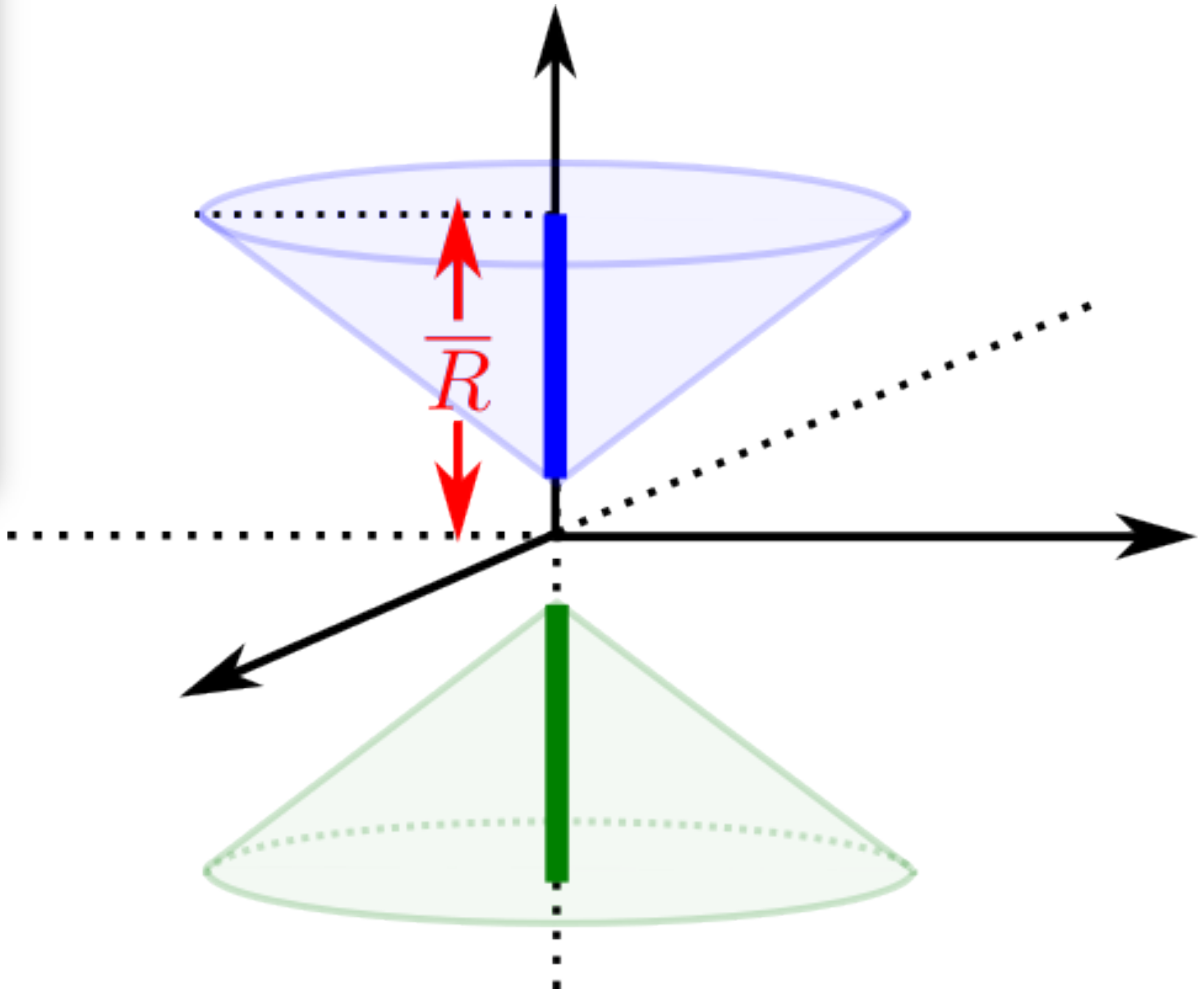
- G-invariant distribution $\mathscr{D}$ over $\mathbb{R}^{d_0} \times \{\pm 1\}$.
- $\mathscr{D}$ can be separated by an invariant classifier β_0.



Theorem [Chen and **Z.**, *NeurIPS* 2023]

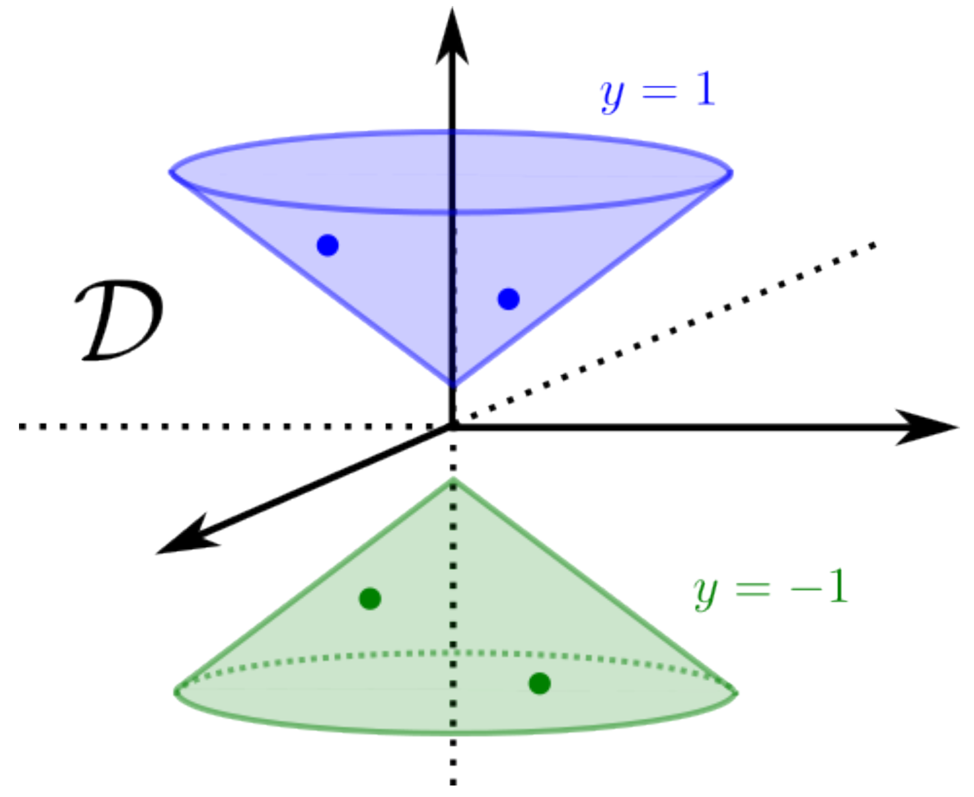
Let $\overline{R} = \inf \left\{ r > 0 : \|\overline{\mathbf{x}}\| \leq r \right\}$. For any $\delta > 0$, w.p. at least $1 - \delta$,

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathscr{D}} \left[y \neq \text{sign}\left(\left\langle \mathbf{x}, \beta_{\text{inv}}^{\infty} \right\rangle \right) \right] \leq \frac{2\overline{R}\|\beta_0\|}{\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}$$



Improved generalization

- G-invariant distribution $\mathscr{D}$ over $\mathbb{R}^{d_0} \times \{\pm 1\}$.
- $\mathscr{D}$ can be separated by an invariant classifier β_0.



Theorem [Chen and **Z.**, *NeurIPS* 2023]

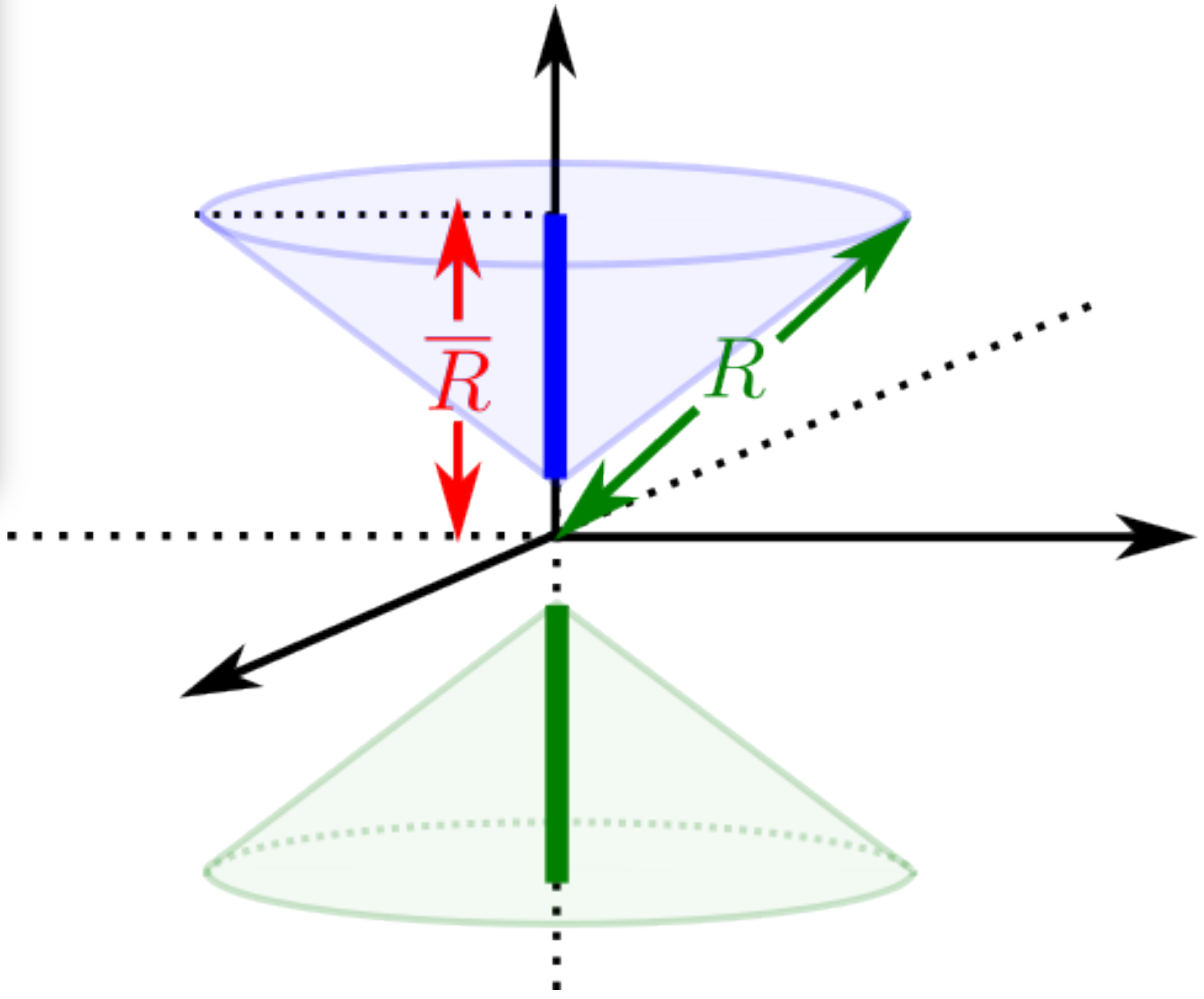
Let $\overline{R} = \inf\left\{r > 0 : \|\overline{\mathbf{x}}\| \leq r\right\}$. For any $\delta > 0$, w.p. at least $1-\delta$,

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathscr{D}}\left[y \neq \operatorname{sign}\left(\left\langle \mathbf{x}, \beta^{\infty}_{\mathrm{inv}}\right\rangle\right)\right] \leq \frac{2\overline{R}\|\beta_0\|}{\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}$$

Remark: In comparison, for fully-connected networks, we have

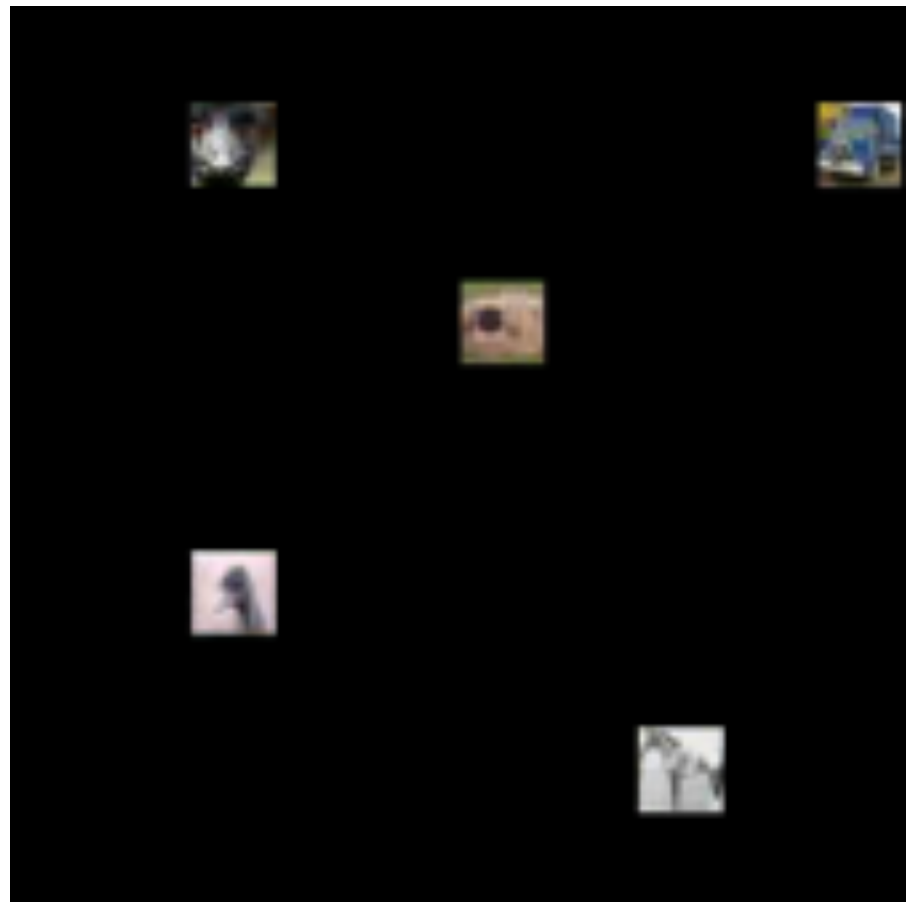
$$\mathbb{P}_{(\mathbf{x},y)\sim\mathscr{D}}\left[y \neq \operatorname{sign}\left(\left\langle \mathbf{x}, \beta^{\infty}_{\mathrm{fc}}\right\rangle\right)\right] \leq \frac{2R\|\beta_0\|}{\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{2n}},$$

where $R = \inf\left\{r > 0 : \|\mathbf{x}\| \leq r \text{ with probability 1}\right\} \geq \overline{R}$



Conclusion

- **Exact quantification of the improvement**
  - **Sample complexity** and **error bound**.



- **Does it converge? To what solution?**
  - **Training dynamics** of equivariant models

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathscr{L}(\mathbf{W}; S) = \sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \mathbf{W}), y_i)$$

Related papers

- J. Birrell, M.A. Katsoulakis, L. Rey-Bellet, **W. Zhu**. “Structure-preserving GANs”. *ICML* (2022)

- Z. Chen, M.A. Katsoulakis, L. Rey-Bellet, **W. Zhu**. “Sample complexity of probability divergences under group symmetry”. *ICML* (2023)

- Z. Chen and **W. Zhu**. “On the implicit bias of linear equivariant steerable networks: margin, generalization, and their equivalence to data augmentation”. *NeurIPS* (2023)

Acknowledgement

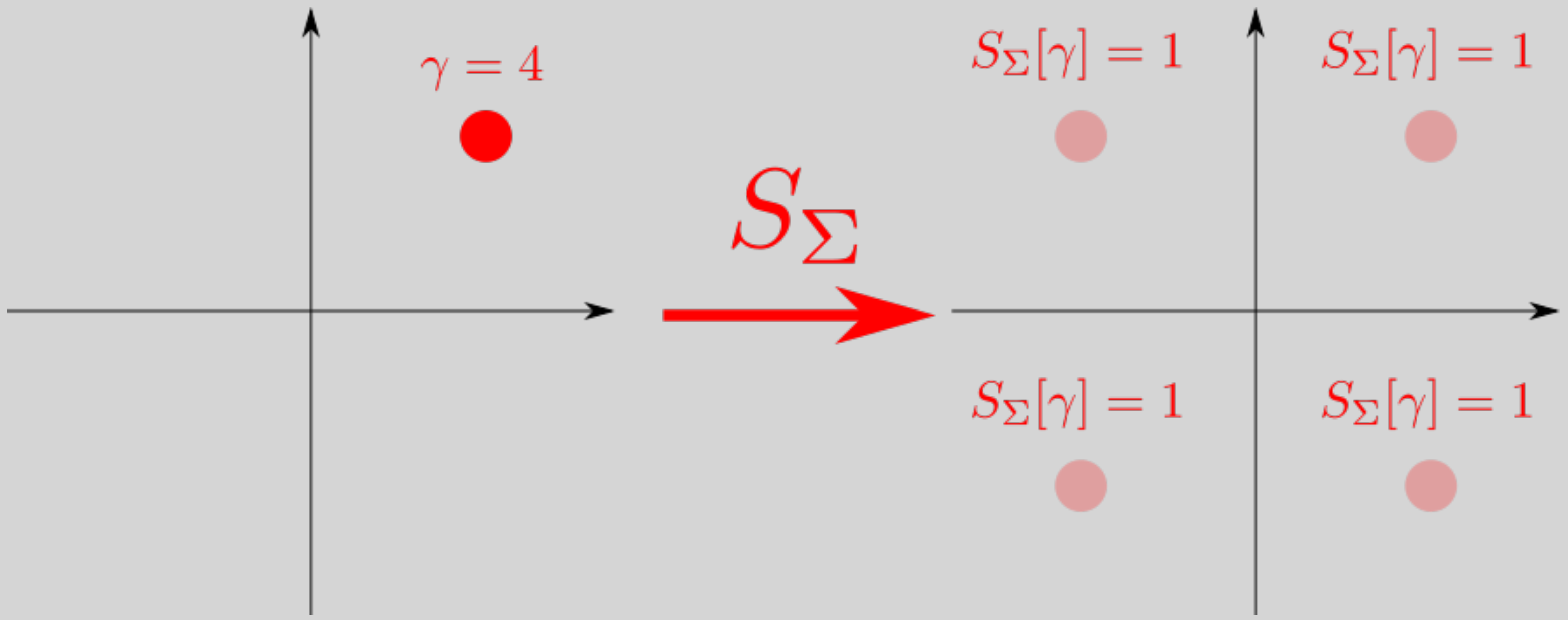
NSF DMS-2052525, DMS-2140982, and DMS-2244976.

Symmetrization operators S_Σ and S^Σ

Symmetrization operators S_Σ and S^Σ

- **Symmetrization of functions:** $S_\Sigma : \mathscr{M}_b(X) \to \mathscr{M}_b(X)$,

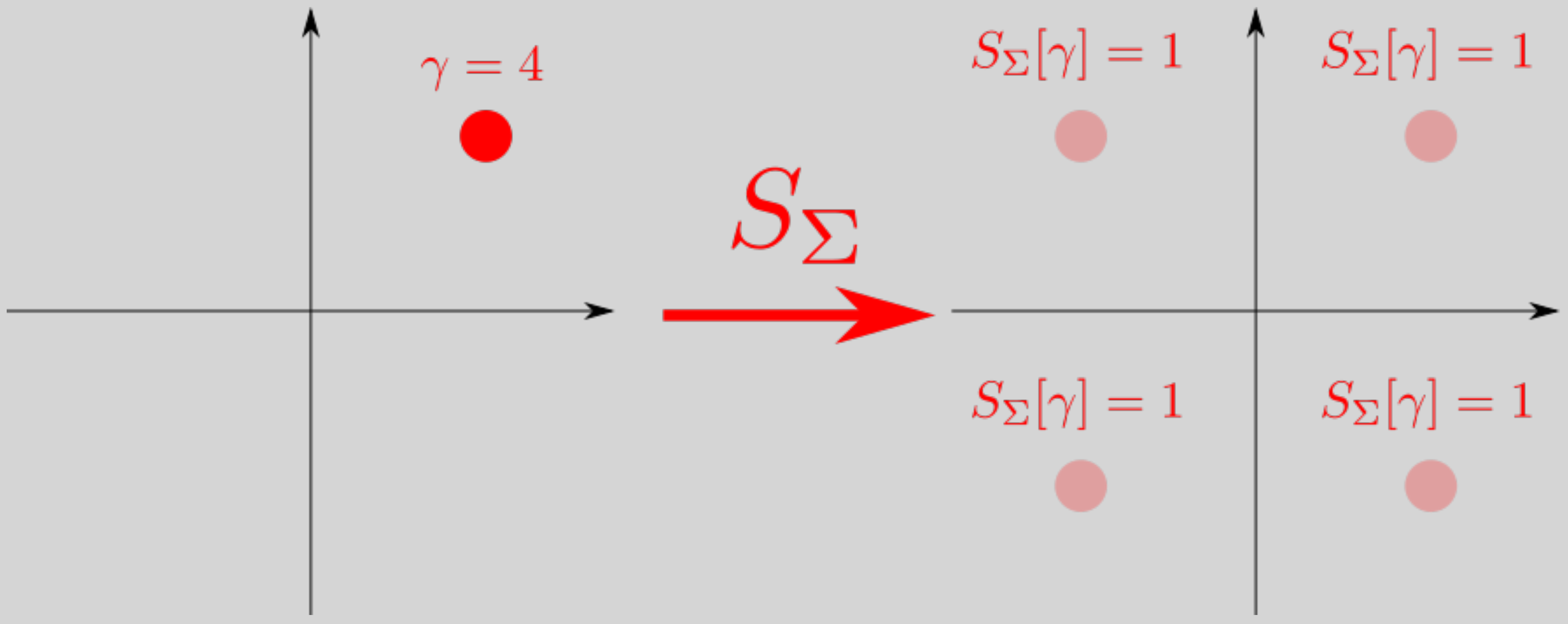
$$S_\Sigma[\gamma](x) = \int_\Sigma \gamma(T_{\sigma'}(x))\mu_\Sigma(d\sigma') = E_{\mu_\Sigma}[\gamma \circ T_{\sigma'}(x)].$$



Symmetrization operators S_Σ and S^Σ

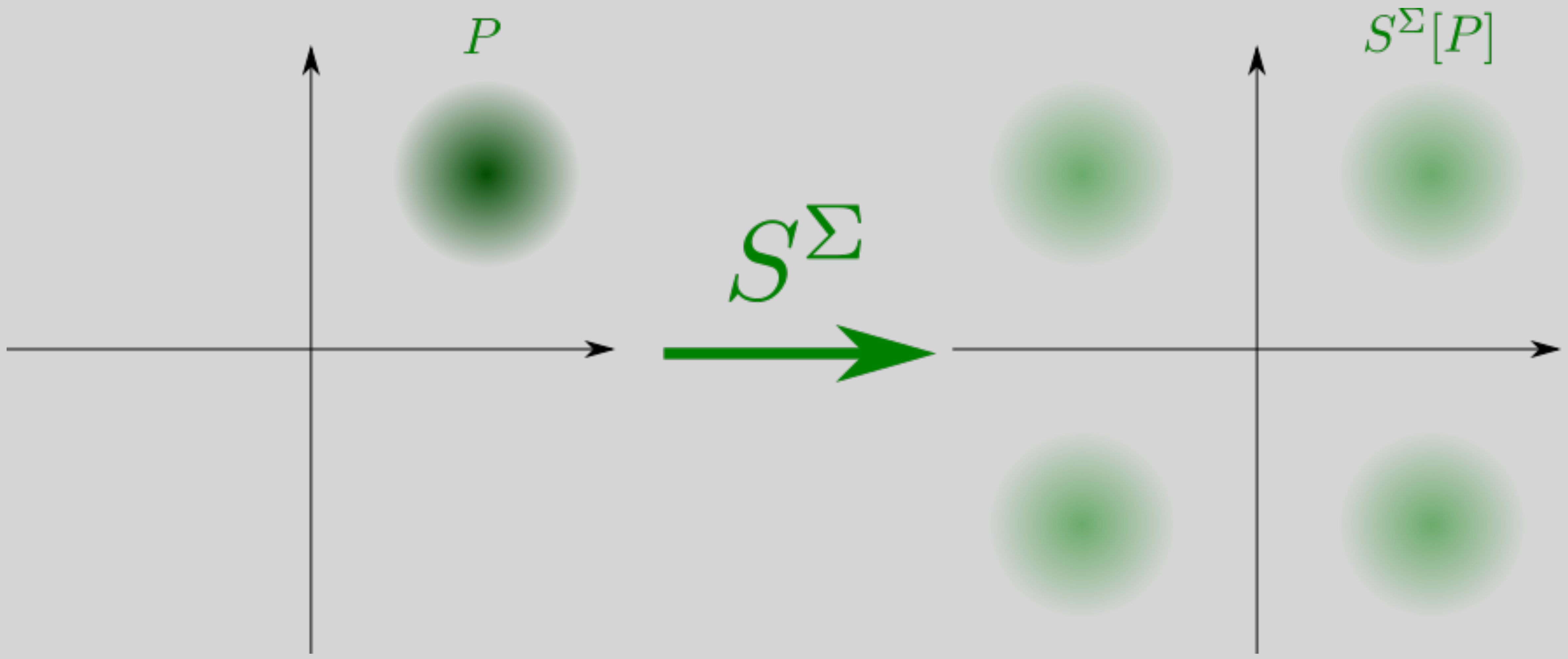
- **Symmetrization of functions:** $S_\Sigma : \mathscr{M}_b(X) \to \mathscr{M}_b(X)$,

$$S_\Sigma[\gamma](x) = \int_\Sigma \gamma(T_{\sigma'}(x))\mu_\Sigma(d\sigma') = E_{\mu_\Sigma}[\gamma \circ T_{\sigma'}(x)].$$



- **Symmetrization of measures:** $S^\Sigma : \mathscr{P}(X) \to \mathscr{P}(X)$,

$$E_{S^\Sigma[P]}\gamma = \int_X S_\Sigma[\gamma](x)dP(x) = E_P S_\Sigma[\gamma],\ \forall \gamma \in \mathscr{M}_b(X).$$

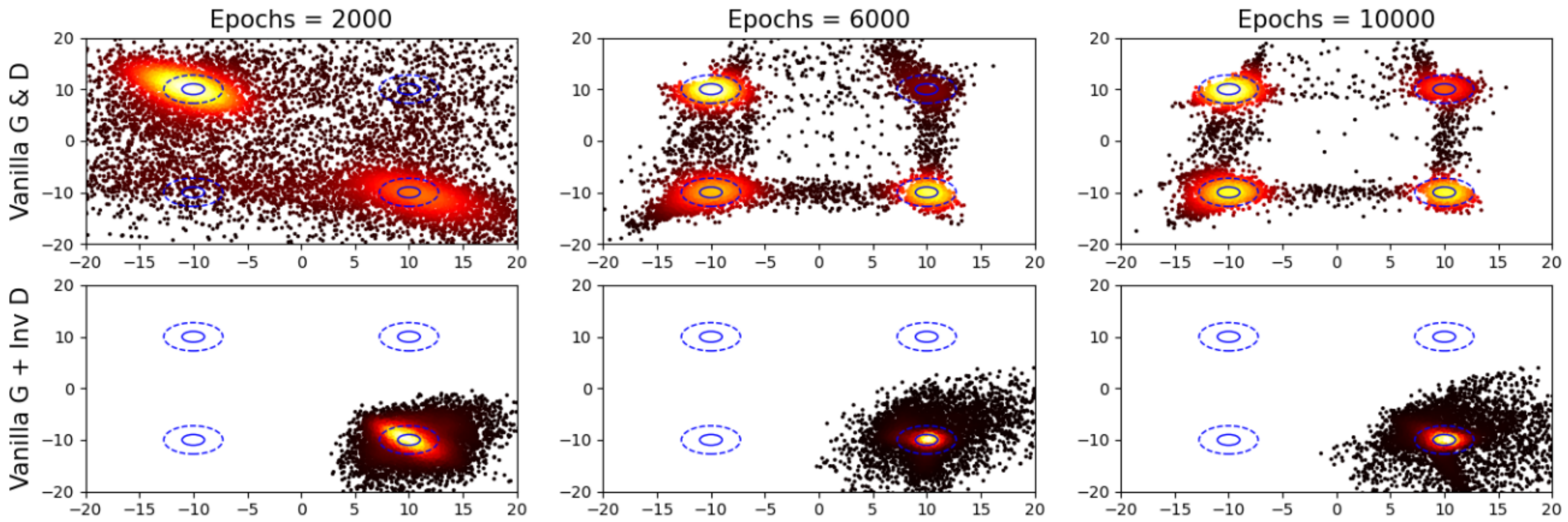


Mode collapse — a warning

Theorem [Birrell, Katsoulakis, Rey-Bellet, **Z.**, ICML 2022]

If $S_\Sigma[\Gamma] \subset \Gamma$ and $P, Q \in \mathscr{P}(X)$, i.e., **not necessarily Σ-invariant**, then

$$D^{\Gamma_\Sigma^{\text{inv}}}(Q\|P) = D^{\Gamma}(S^\Sigma[Q]\|S^\Sigma[P]).$$



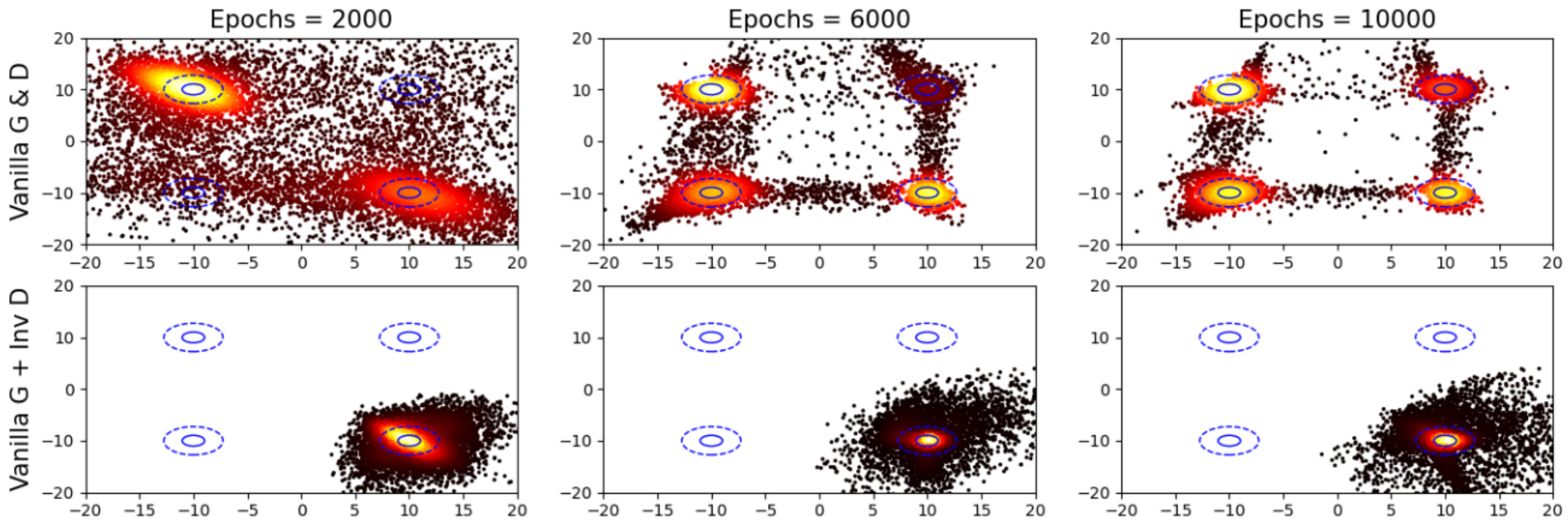
Mode collapse — a warning

Theorem [Birrell, Katsoulakis, Rey-Bellet, **Z.**, ICML 2022]

If $S_\Sigma[\Gamma] \subset \Gamma$ and $P, Q \in \mathscr{P}(X)$, i.e., **not necessarily Σ-invariant**, then

$$D^{\Gamma_\Sigma^{\text{inv}}}(Q\|P) = D^{\Gamma}(S^\Sigma[Q]\|S^\Sigma[P]).$$

- Reducing Γ to $\Gamma_\Sigma^{\text{inv}}$ might result in "**mode collapse**" if P_g is **NOT** Σ-invariant



Mode collapse — a warning

Theorem [Birrell, Katsoulakis, Rey-Bellet, **Z.**, ICML 2022]

If $S_\Sigma[\Gamma] \subset \Gamma$ and $P, Q \in \mathscr{P}(X)$, i.e., **not necessarily Σ-invariant**, then

$$D^{\Gamma_\Sigma^{\text{inv}}}(Q\|P) = D^{\Gamma}(S^\Sigma[Q]\|S^\Sigma[P]).$$

- Reducing Γ to $\Gamma_\Sigma^{\text{inv}}$ might result in "**mode collapse**" if P_g is **NOT** Σ-invariant
- The reason is as P_g only needs to equal Q after Σ-symmetrization.