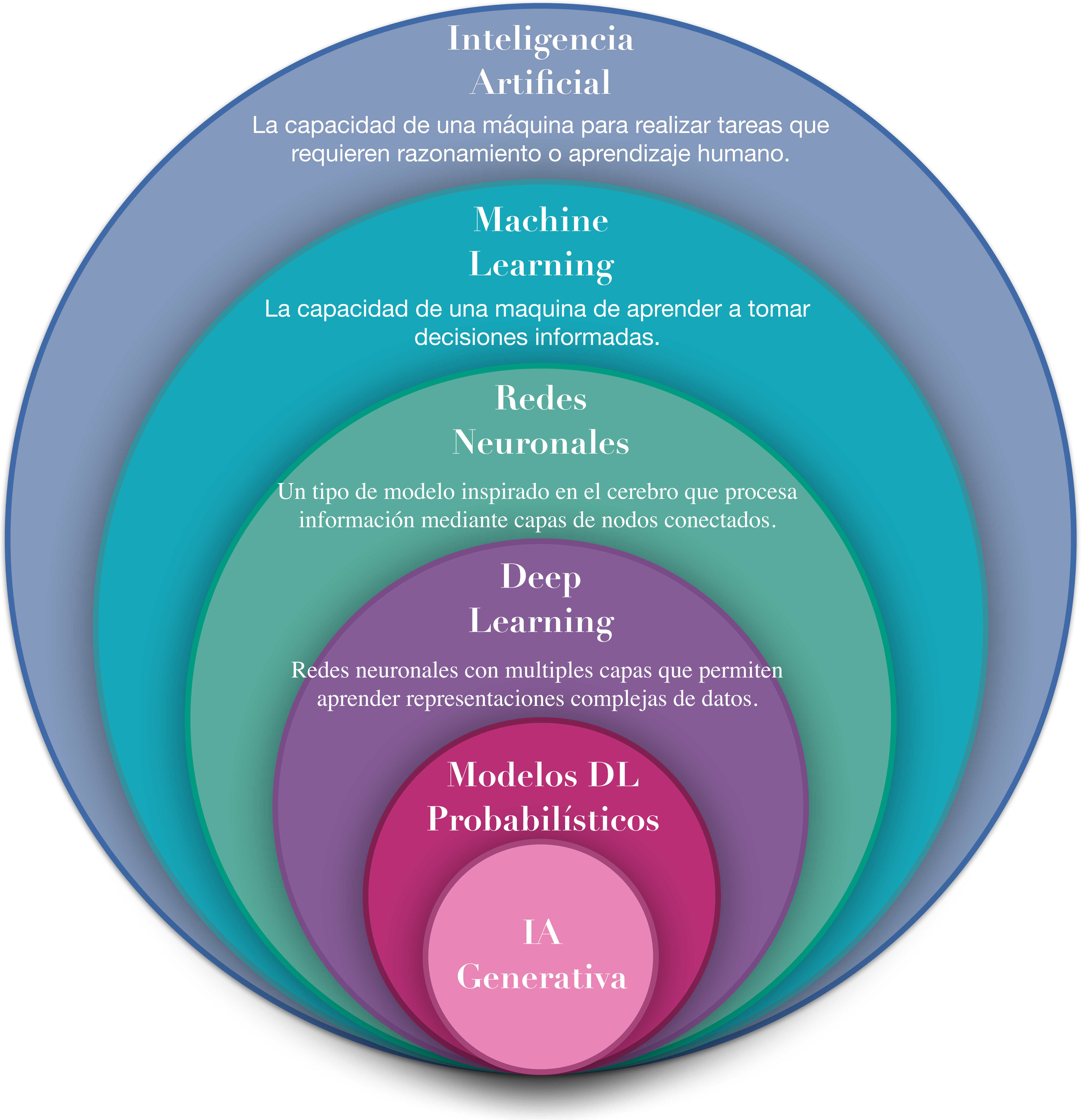
FCEN, UBA

Modelos Generativos

Cecilia Garraffo

Outline

- Modelos Generativos vs Discriminativos
- Redes Generativas Convolucionales
- Flujos Normalizantes
- Modelos de Difusión
- Modelos Generativos Físicos
- Aplicaciones en Cosmología


Inteligencia Artificial
La capacidad de una máquina para realizar tareas que requieren razonamiento o aprendizaje humano.
Machine Learning
La capacidad de una maquina de aprender a tomar decisiones informadas.
Redes Neuronales
Un tipo de modelo inspirado en el cerebro que procesa información mediante capas de nodos conectados.
Deep Learning
Redes neuronales con multiples capas que permiten aprender representaciones complejas de datos.
Modelos DL Probabilísticos
IA Generativa

IA

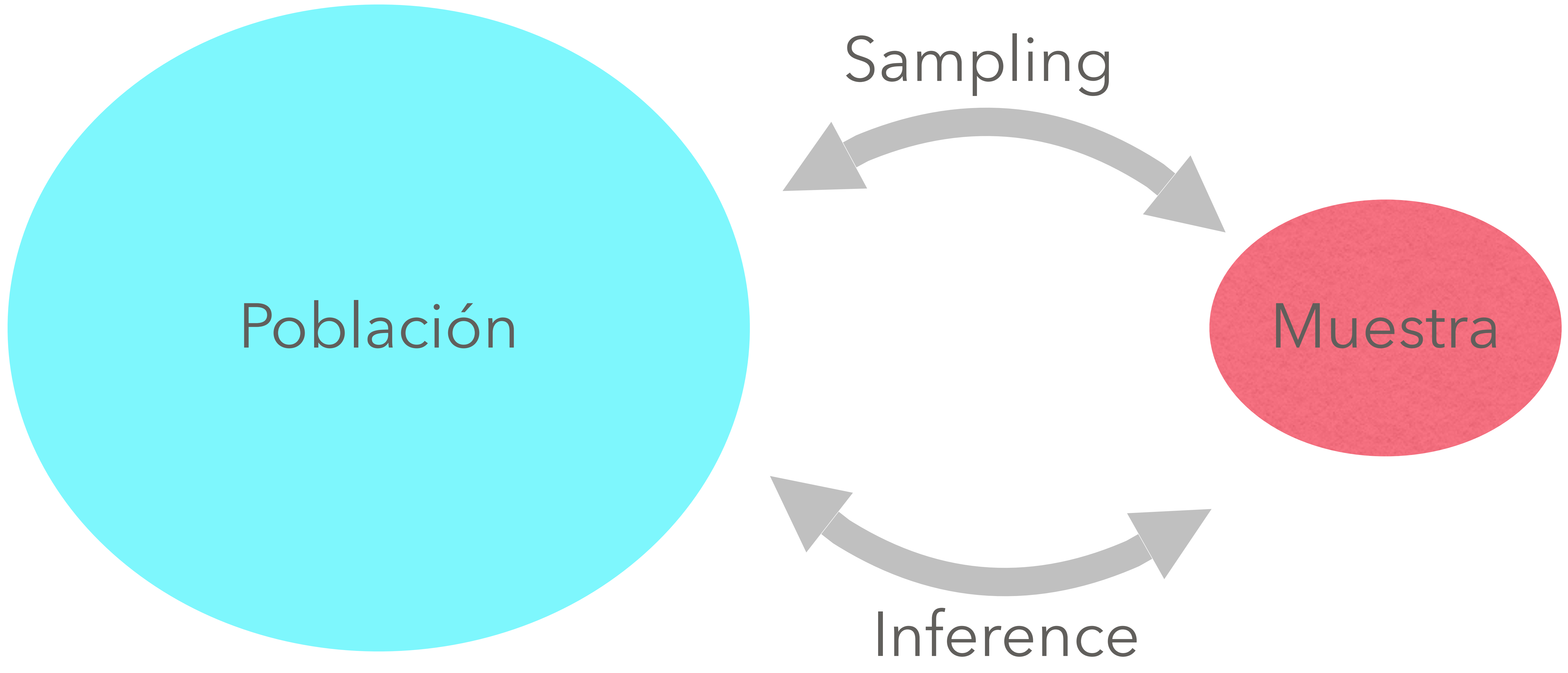
Generativa

Recap

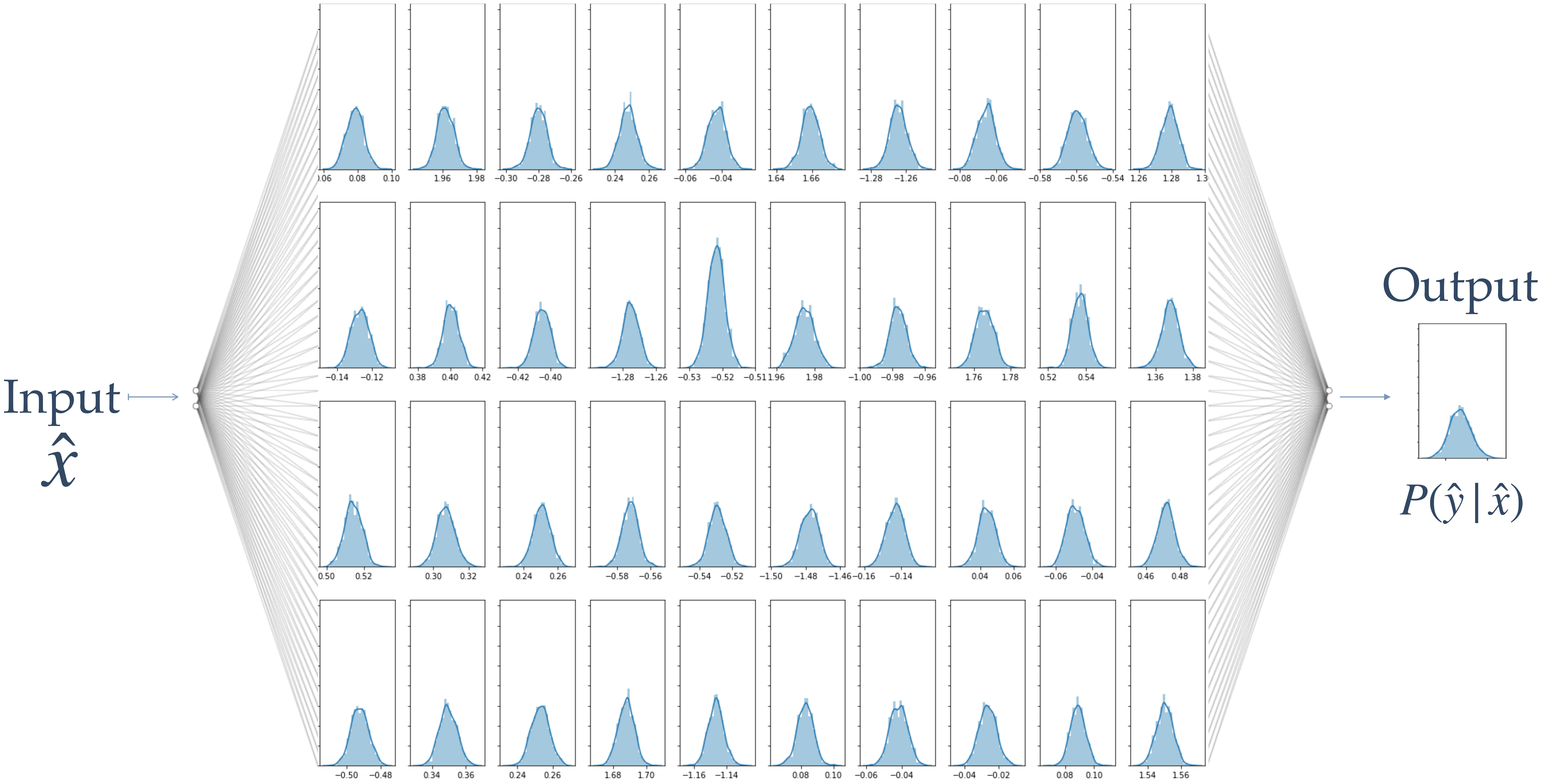
- Aplicación de Inferencia Bayesiana a Redes Neuronales
- Drop Out como Aproximación Bayesiana
- Inferencia via Bootstrap
- Medida de performance / cuantificación de incerteza

Inferencia Estadística

Sacar conclusions de una distribución de probabilidades a partir de una muestra



Inference in Machine Learning

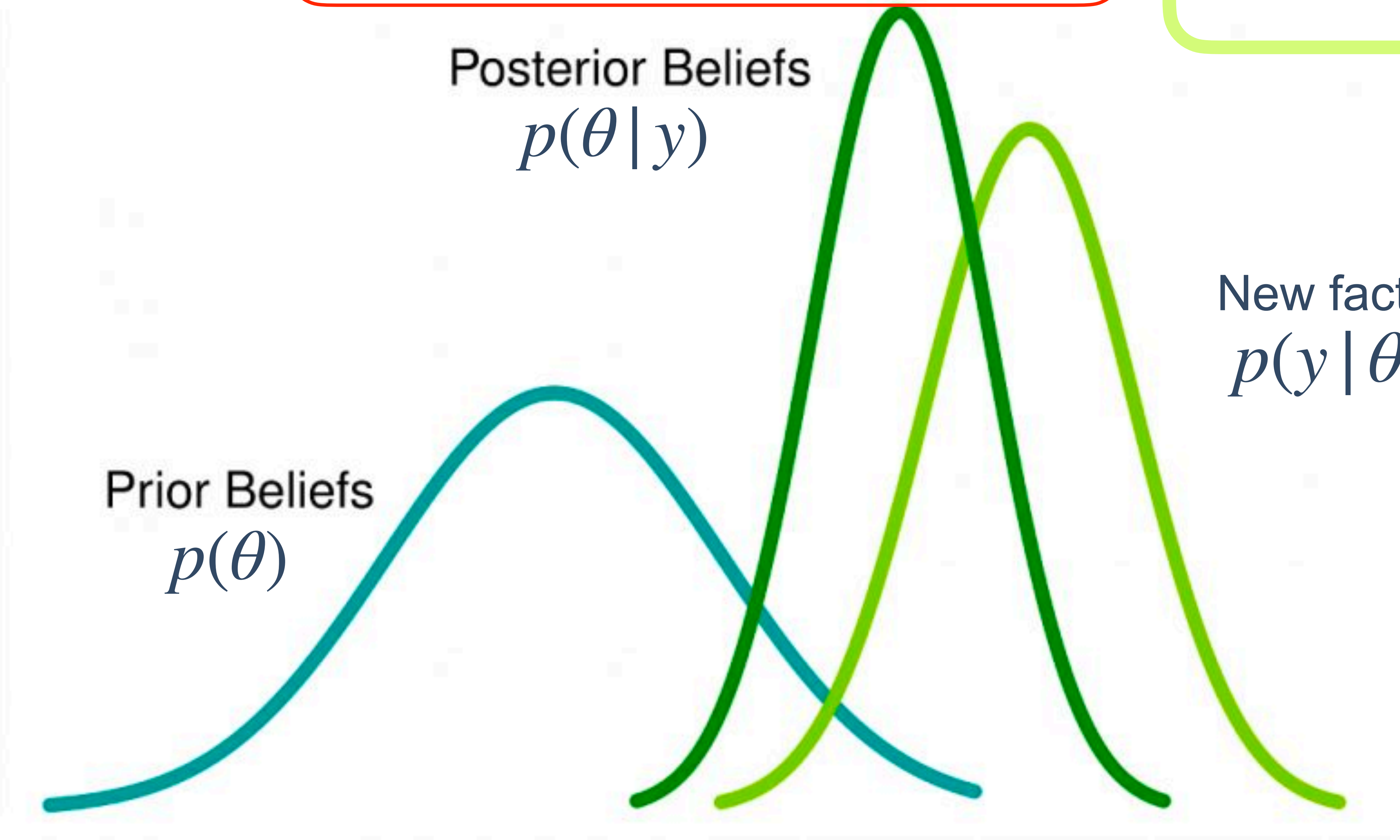


Bayesian Inference

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

Intractable

$$p(y) = \int_\theta p(y|\theta)p(\theta)d\theta$$



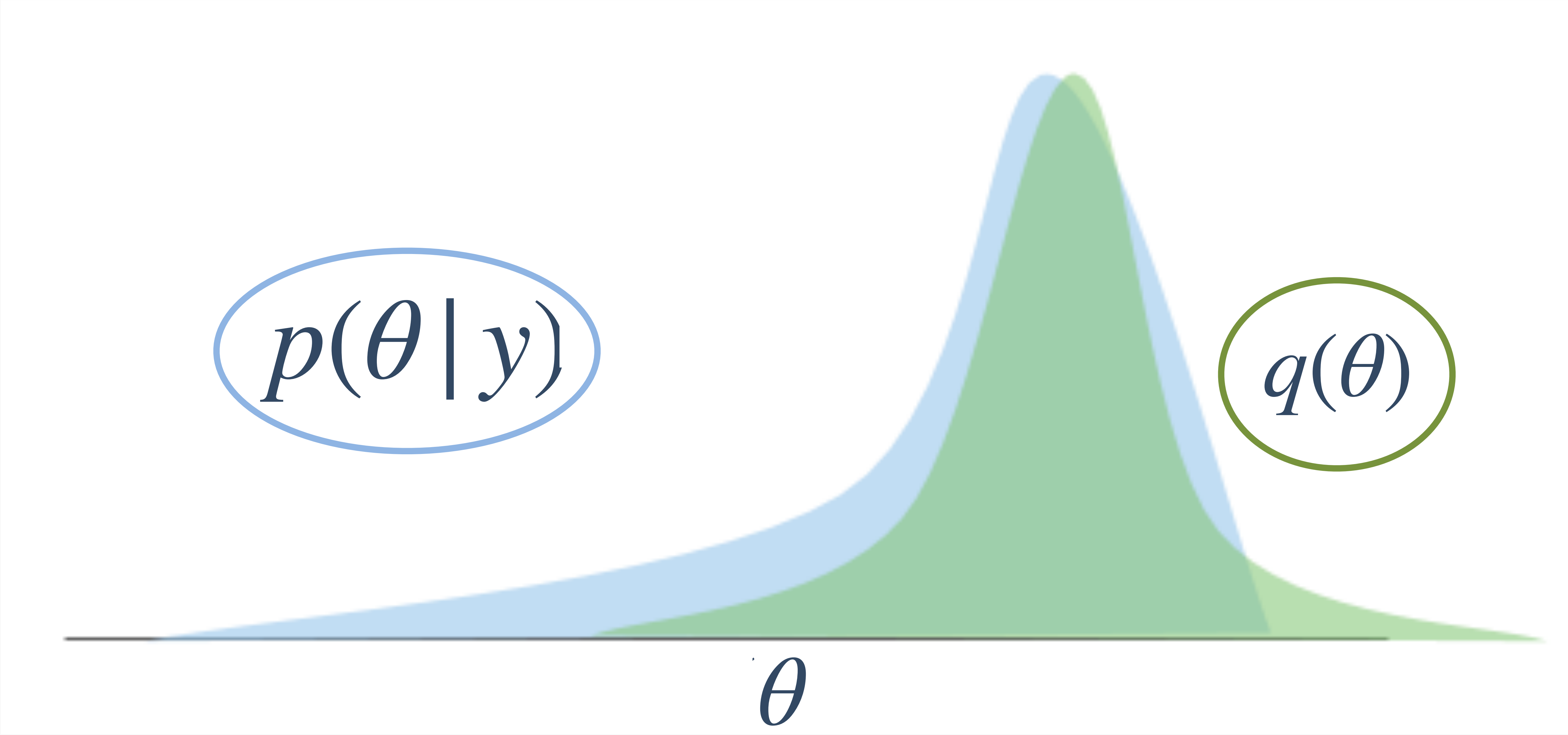
"When the facts change, I change my mind. What do you do, sir? "

John Maynard Keynes

Approximate Bayesian Inference: Variational Inference

Optimization approach -> Q a family of "nice" distributions

$$p(\theta | y) = \frac{p(y | \theta)\, p(\theta)}{\int p(y, \theta) d\theta}$$



Approximate Bayesian Inference: Variational Inference

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{\int p(y, \theta) d\theta} \quad \int p(y, \theta) d\theta = p(y) \longleftarrow \text{evidence}$$

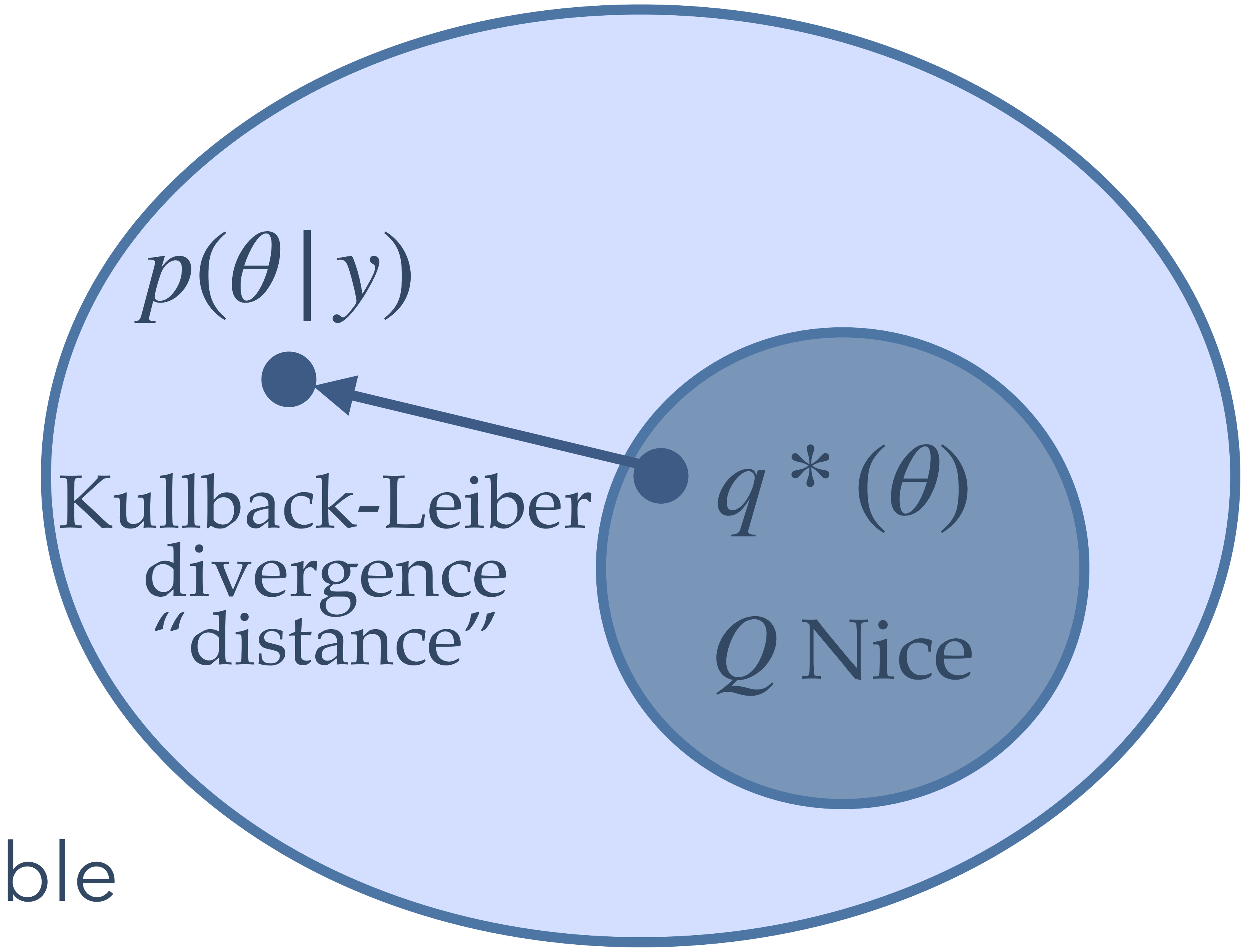
$$p(y) = \int\int_{\dots}\int p(y \mid \theta_1, \theta_2, \theta_3, \dots, \theta_m) d\theta_1 d\theta_2 d\theta_3 \dots d\theta_m$$

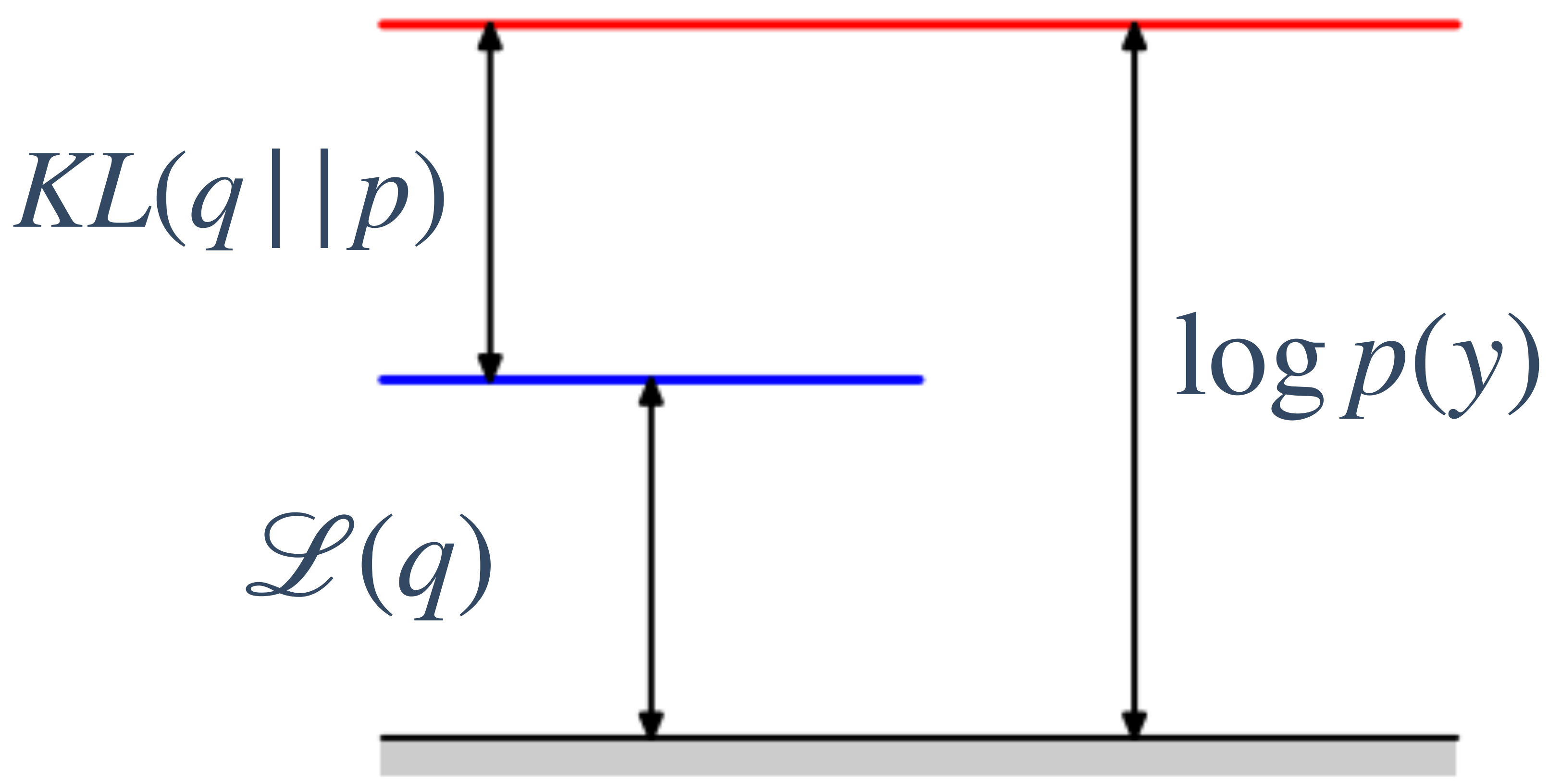
intractable

approximation

$$p(\theta \mid y) \approx q^*(\theta) \longleftarrow \text{tractable family of distributions}$$

$$q^*(\theta) = argmin_{q \in Q} KL(q(\cdot)) \mid\mid p(\cdot \mid y)$$



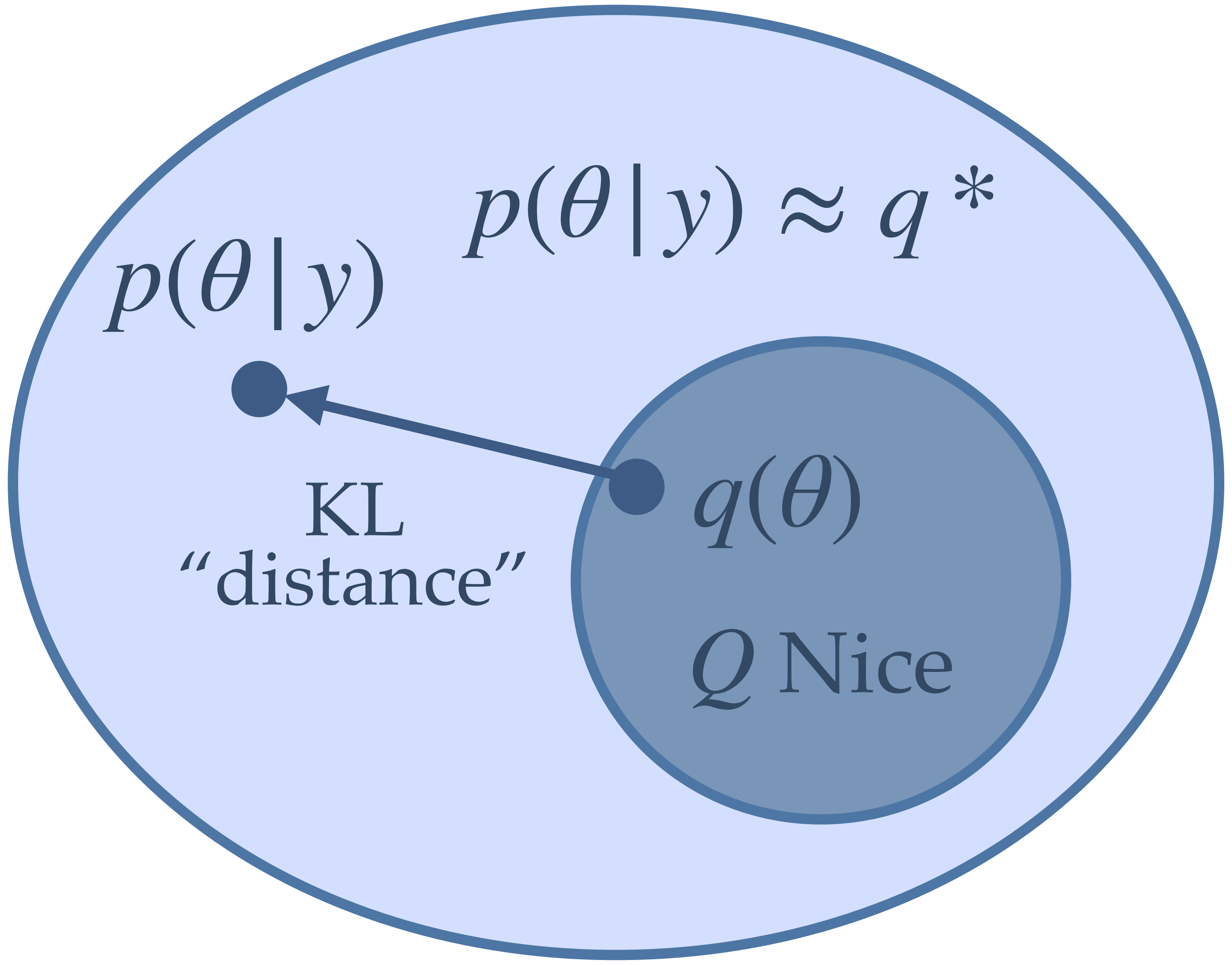


Bishop

Approximate Bayesian Inference: Variational Bayes

$$p(\theta|y) = \frac{p(y|\theta)\,p(\theta)}{\int p(y,\theta)d\theta}$$

$$q^*(\theta) = argmin_{q\in Q} KL(q(\cdot))||p(\cdot|y)$$

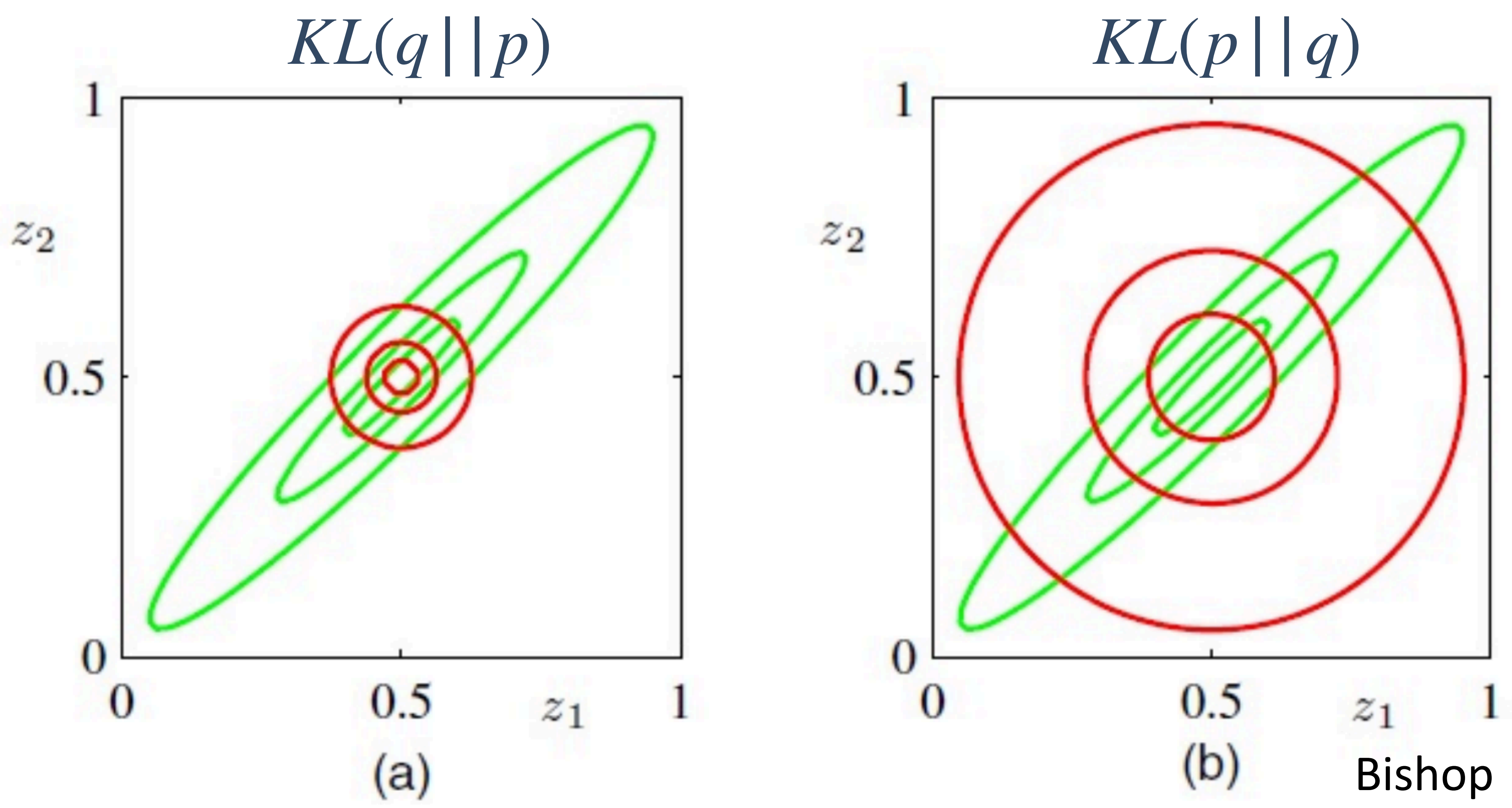


$$KL \geq 0 \implies \log p(y) \geq \text{ELBO}$$

$$q^* = argmax_{q\in Q}\, ELBO(q) = argmax_{q\in Q} \int q(\theta)\log\frac{p(\theta,y)}{q(\theta)}d\theta$$

Approximate Bayesian Inference: Mean Field Variational Bayes

Green:
1, 2, and 3 standard deviations for a correlated Gaussian distribution

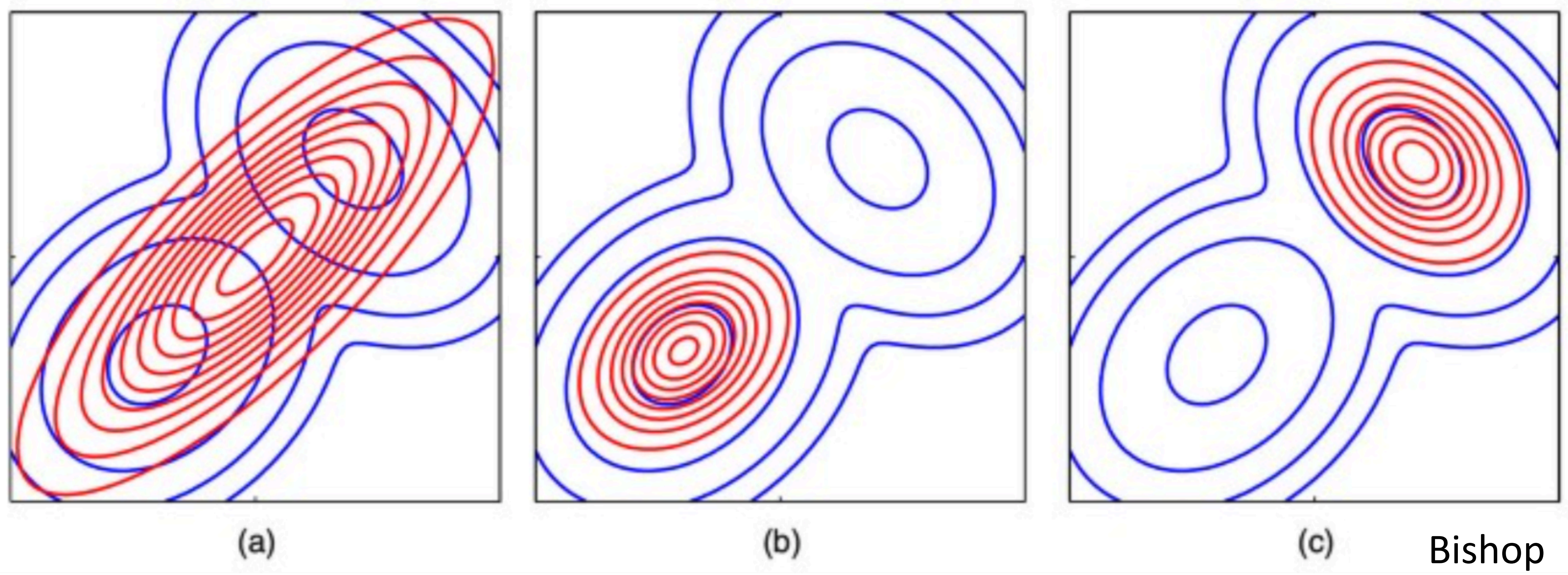


Bishop

Red:
Same levels of an approximate distribution given by the product of two independent univariate Gaussians obtained by:

a) minimizing $KL(q||p)$ divergence
b) minimizing $KL(p||q)$ divergence

Blue:
Bimodal distribution given by a mixture of two Gaussians

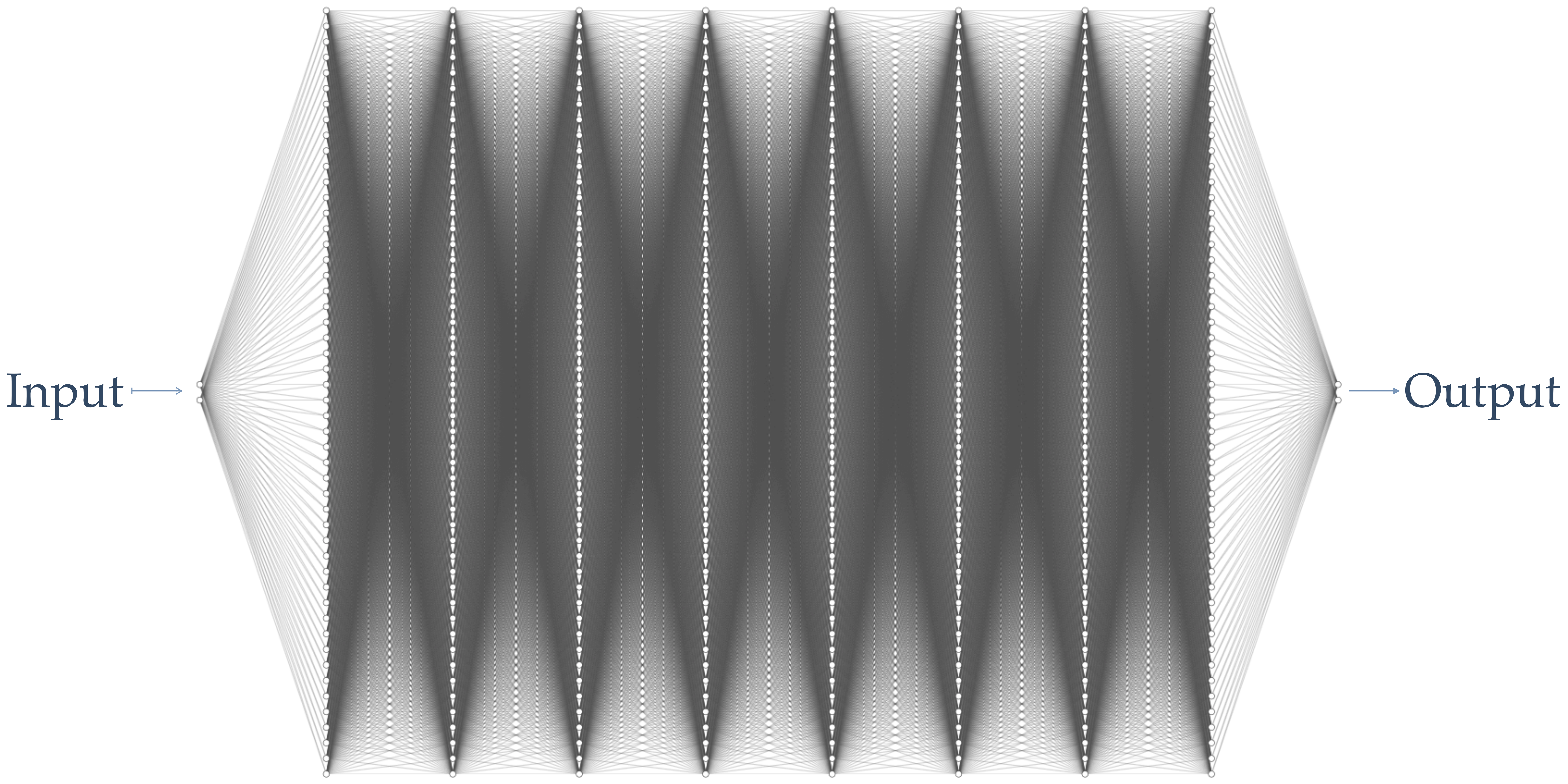


Bishop

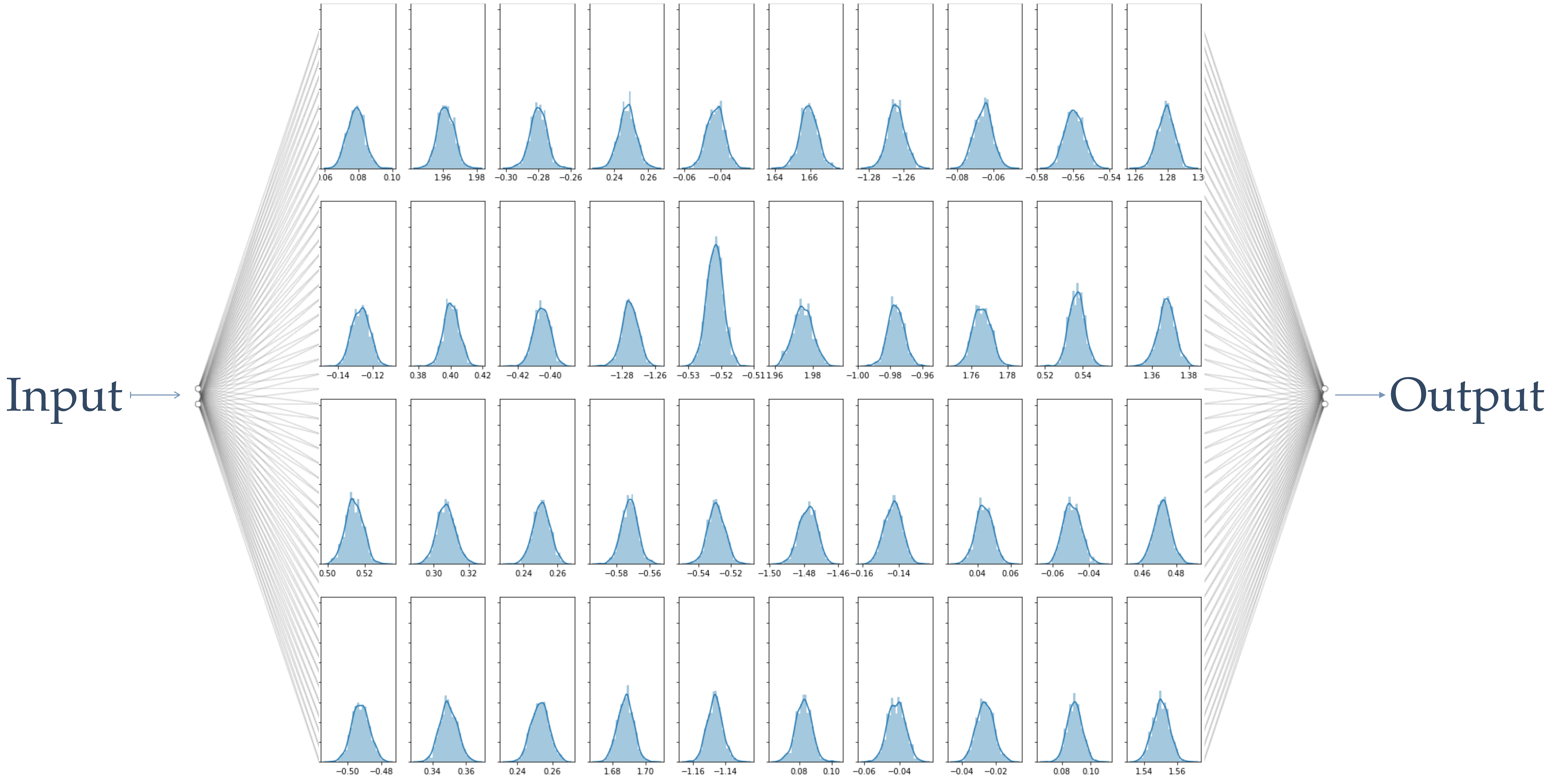
Red:

a) Best approximation by a single Gaussian by minimizing the KL divergence
b) same as (a) but numerically minimizing KL divergence
c) same as in (b) but another local minimum of the KL divergence

Bayesian Inference for Neural Networks



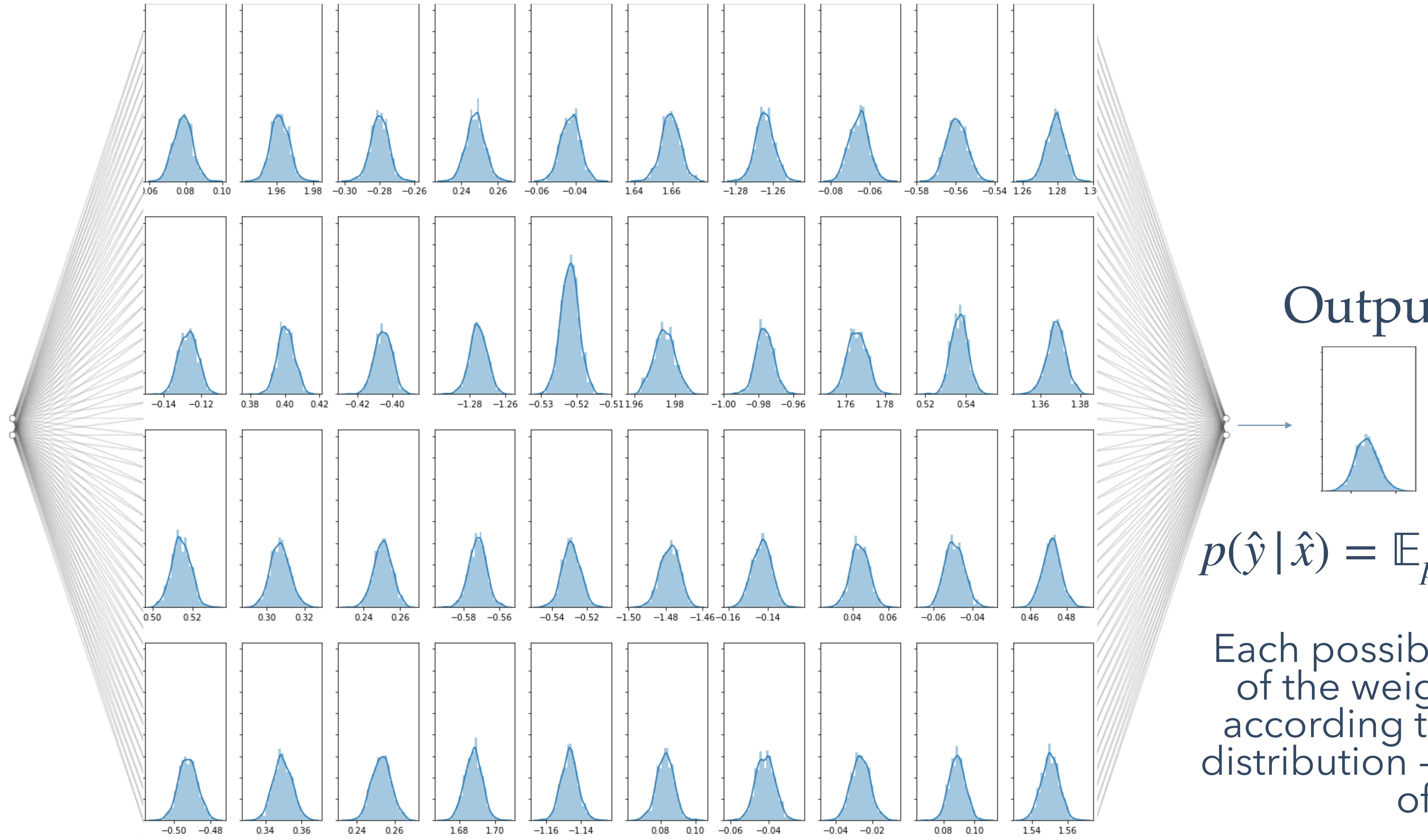
Bayesian Inference for Neural Networks



Bayesian Inference for Neural Networks

Input

$\hat{x}$



Output

$\hat{y}$

$$p(\hat{y}\,|\,\hat{x}) = \mathbb{E}_{p(\omega|y)}[p(\hat{y}\,|\,\hat{x}, \omega)]$$

Each possible configuration of the weights, weighted according to the posterior distribution - infinite number of NNs

$p(\omega\,|\,y)$

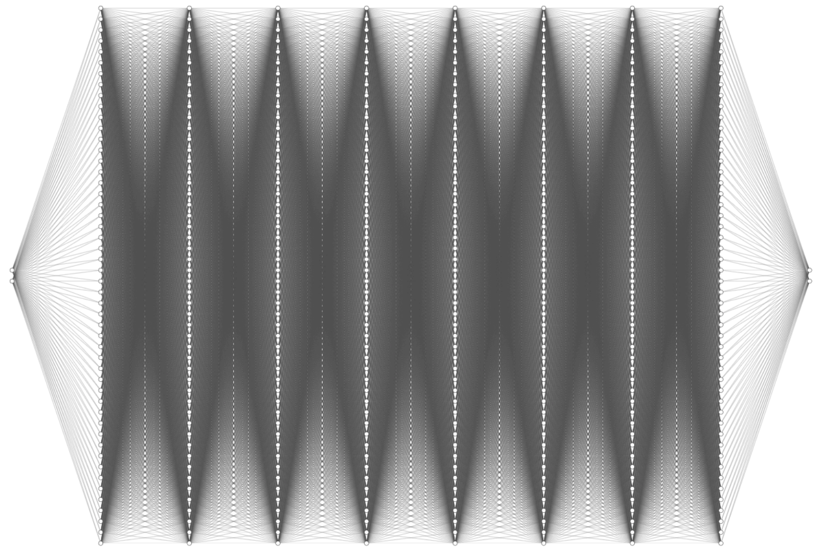
BI for NN calculates the posterior distribution of the weights given the training data

Bayesian Neural Networks

$$\text{Priors} \quad p(\theta) \longrightarrow \frac{p(\theta)\,p(y\,|\,\theta)}{p(y)_{\text{intractable}}} = p(\theta\,|\,y)$$

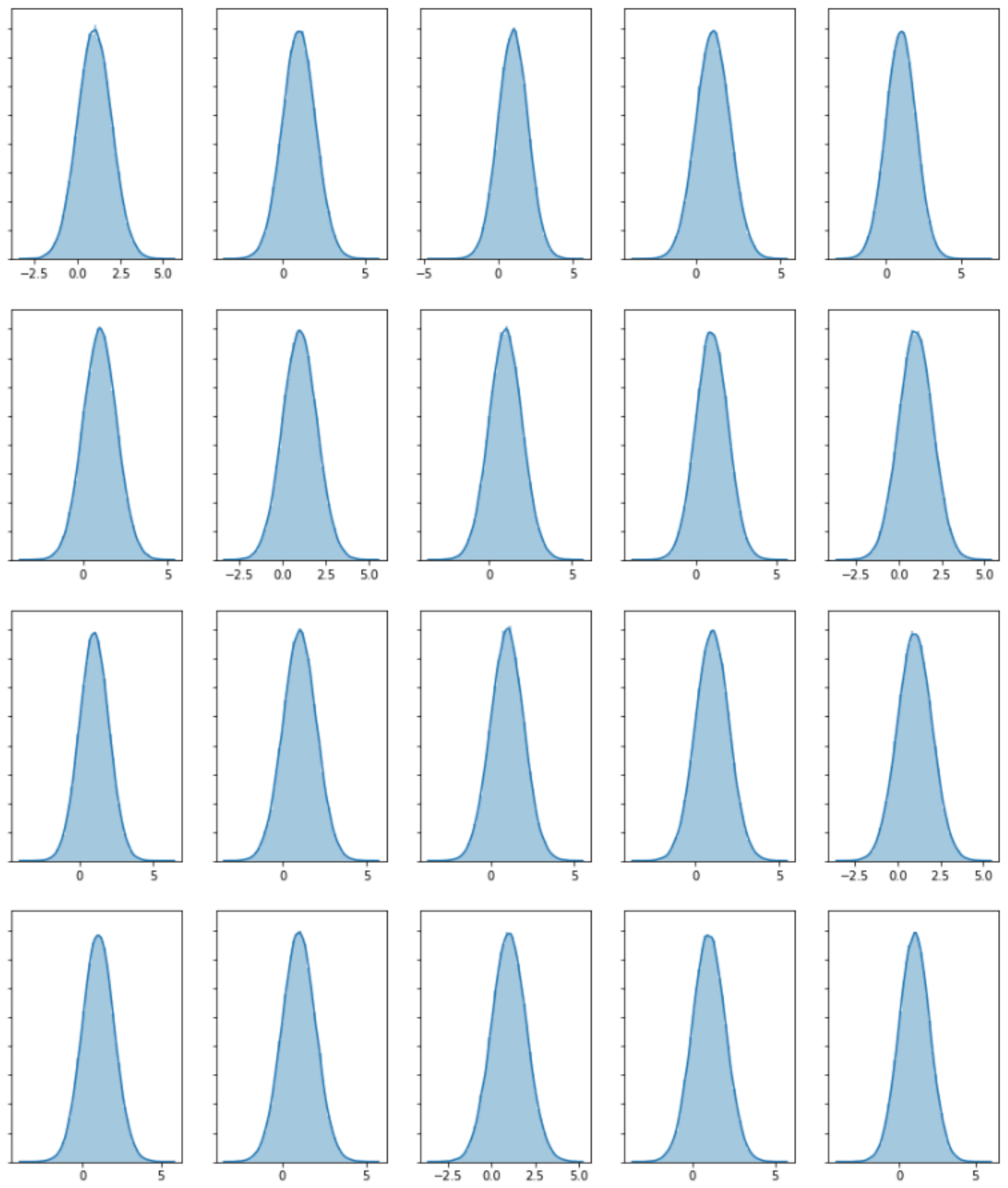
Means

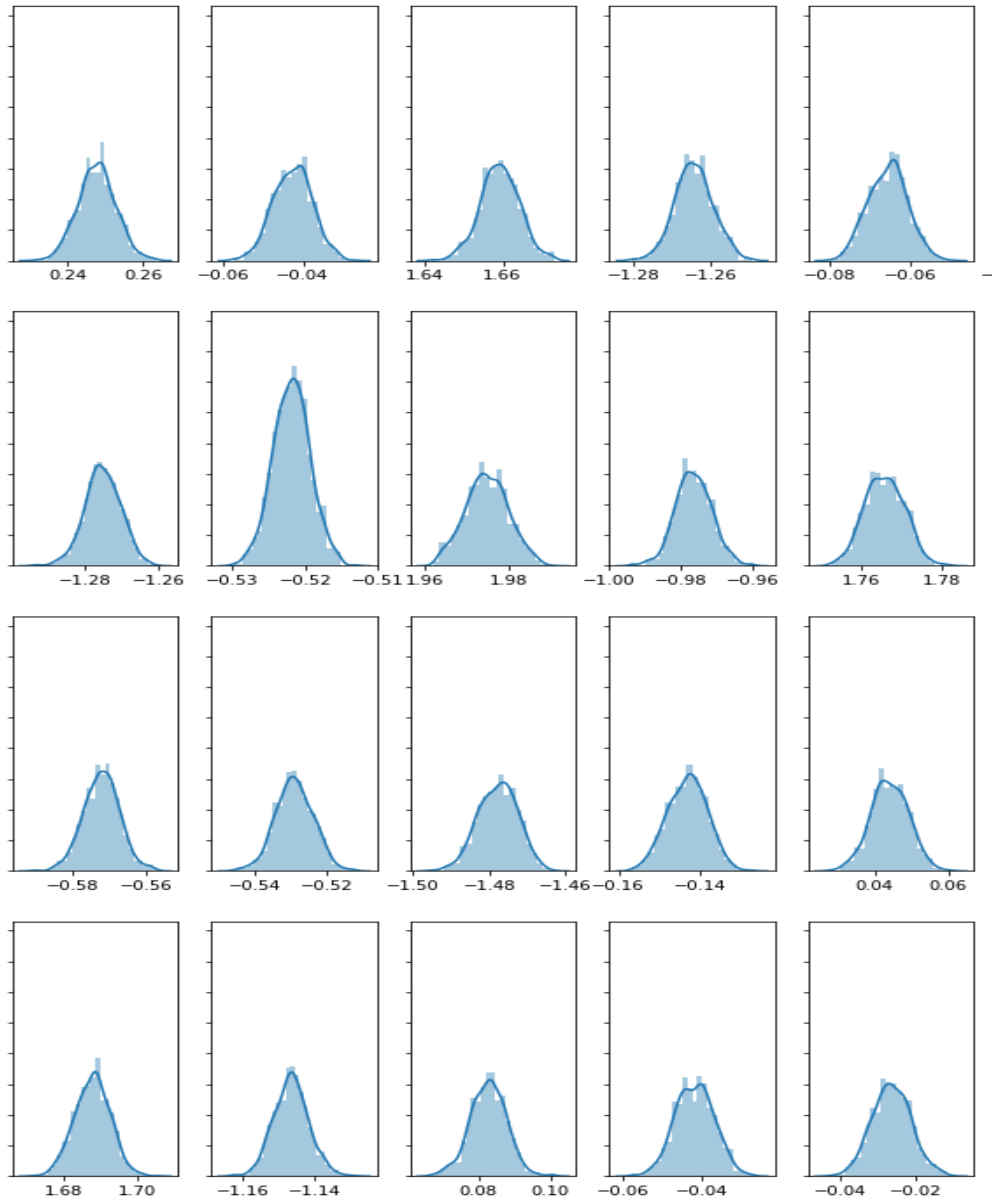
FCNN



&

Scale





Bayesian Neural Networks

MCMC: Eventually accurate $$p(\theta) \times p(y \mid \theta) \propto p(\theta \mid y)$$

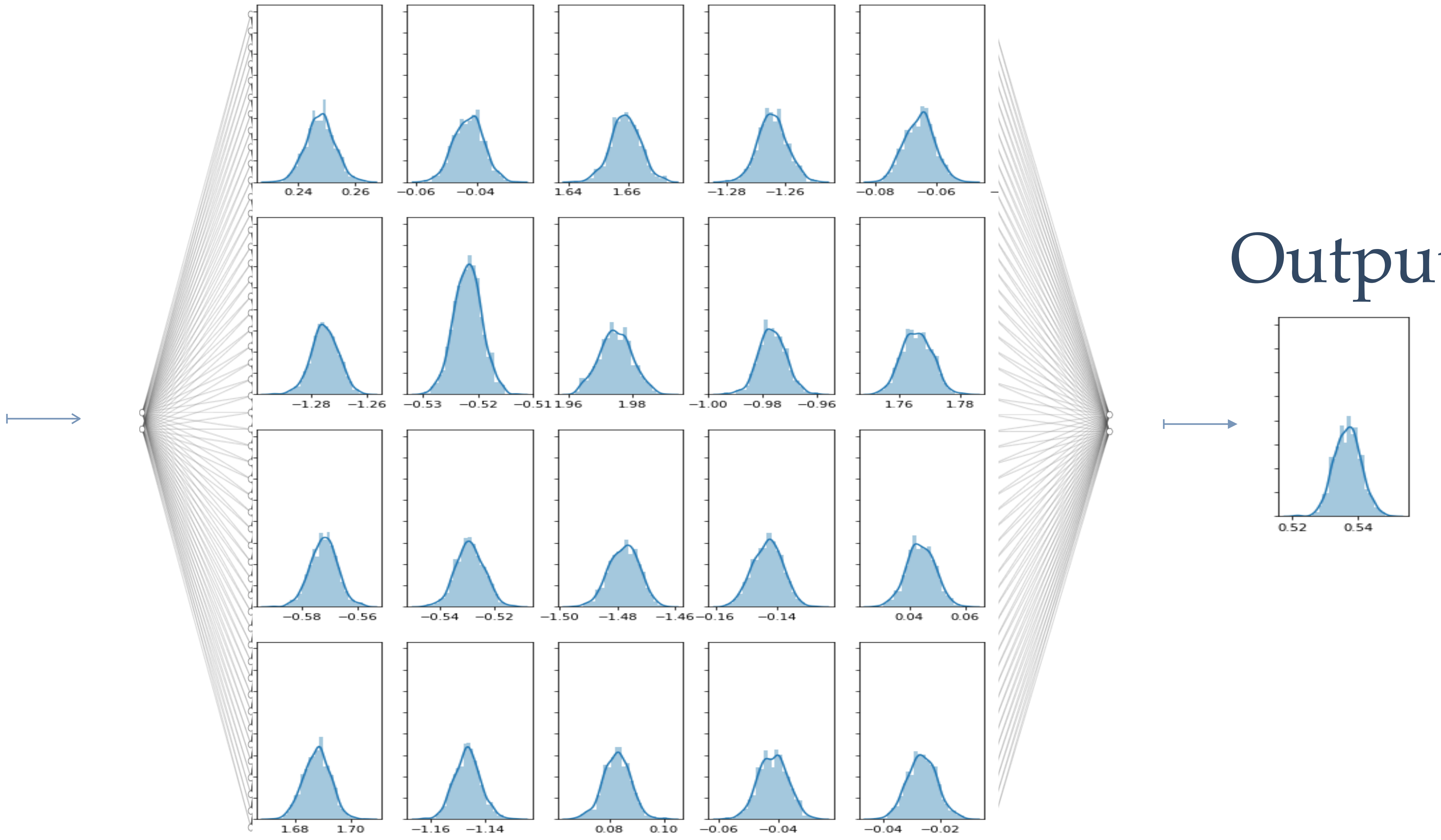
Only forward passes

$$\theta = (\mu, \sigma)$$

$$\mu^j \sim N(\mu^{(j-1)}, \sigma)$$

Step of random walk

Input

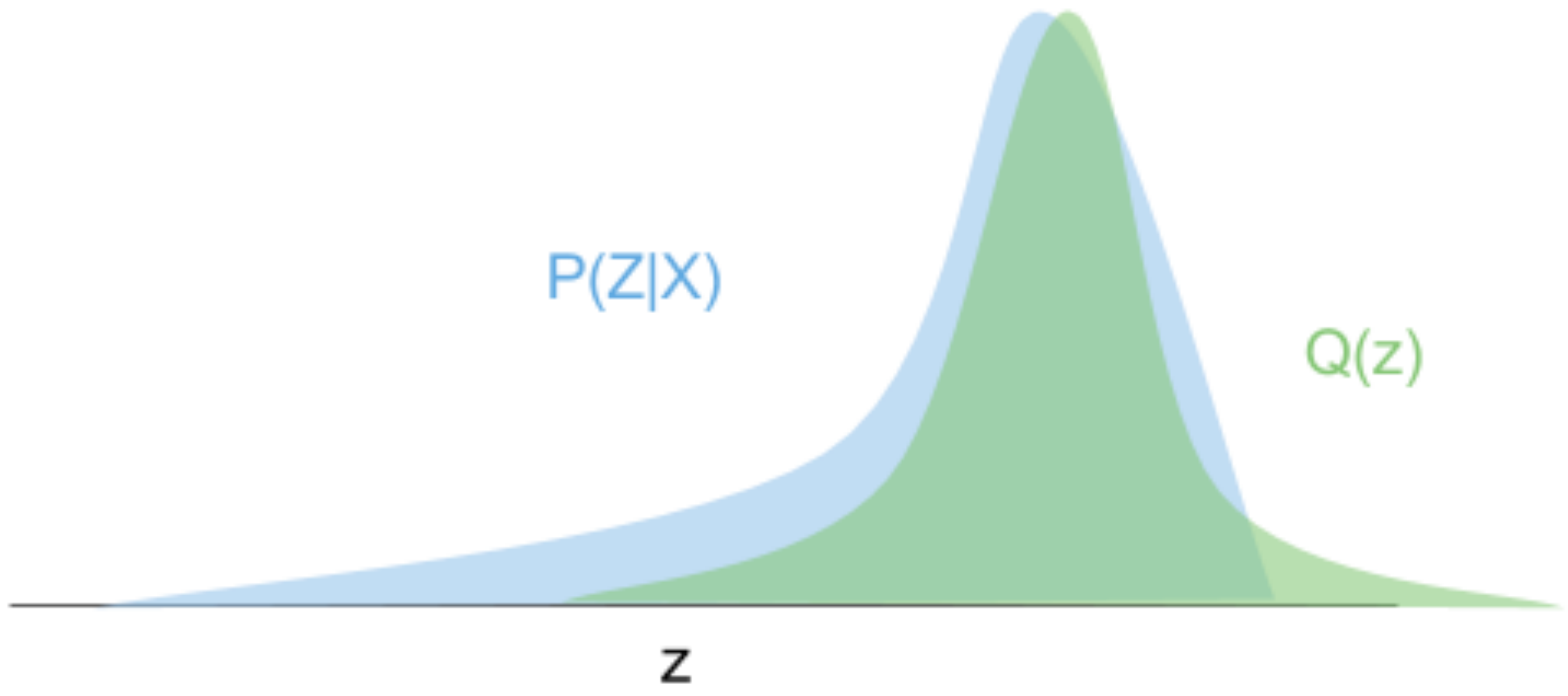


Output

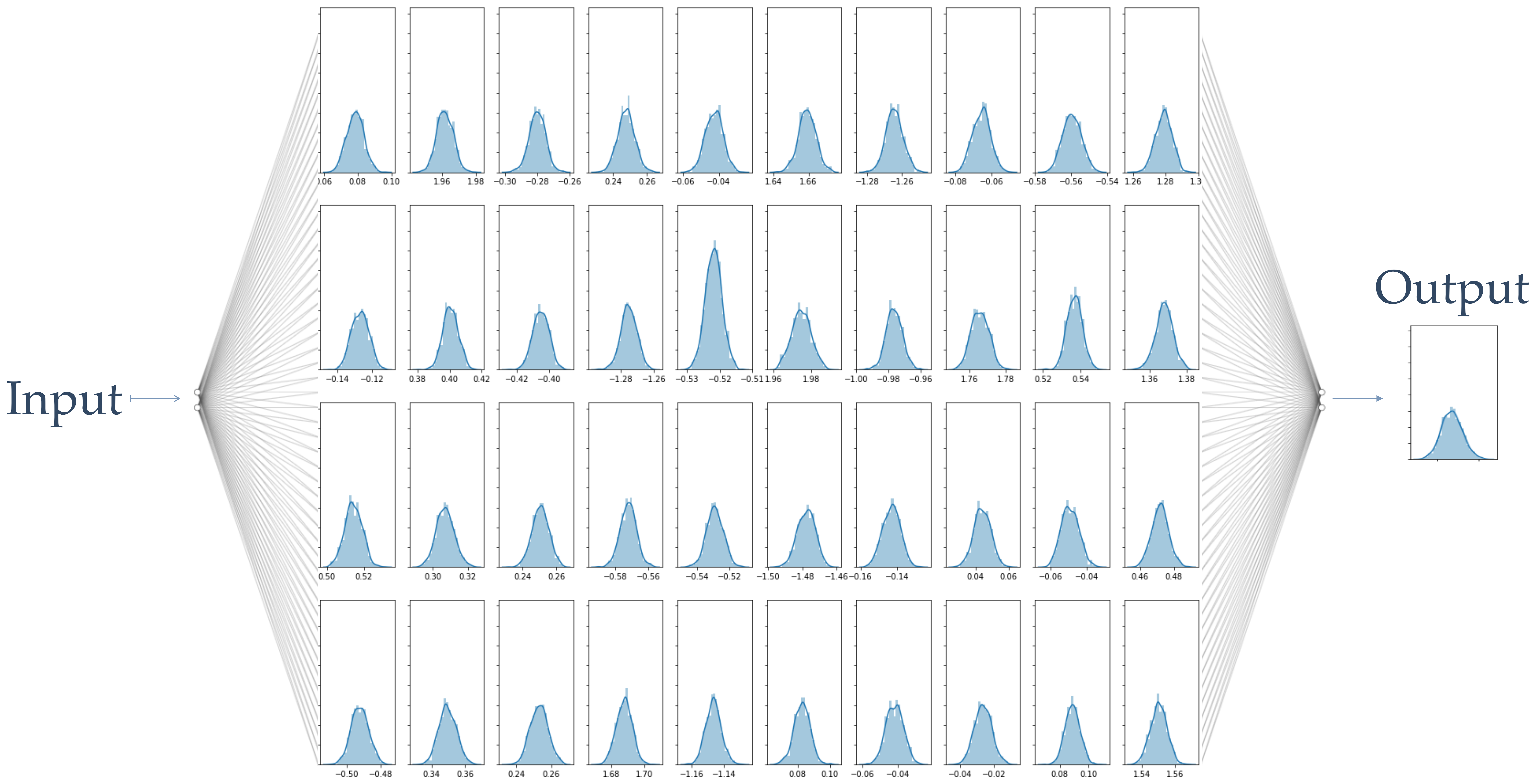
Approximate Bayesian Inference: Variational Bayes

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{\int p(y, \theta) d\theta} \approx q^*$$

Optimization approach ➡ Q a family of "nice" distributions



Bayesian Neural Network



Bayesian Neural Network: Bayes by Backprop

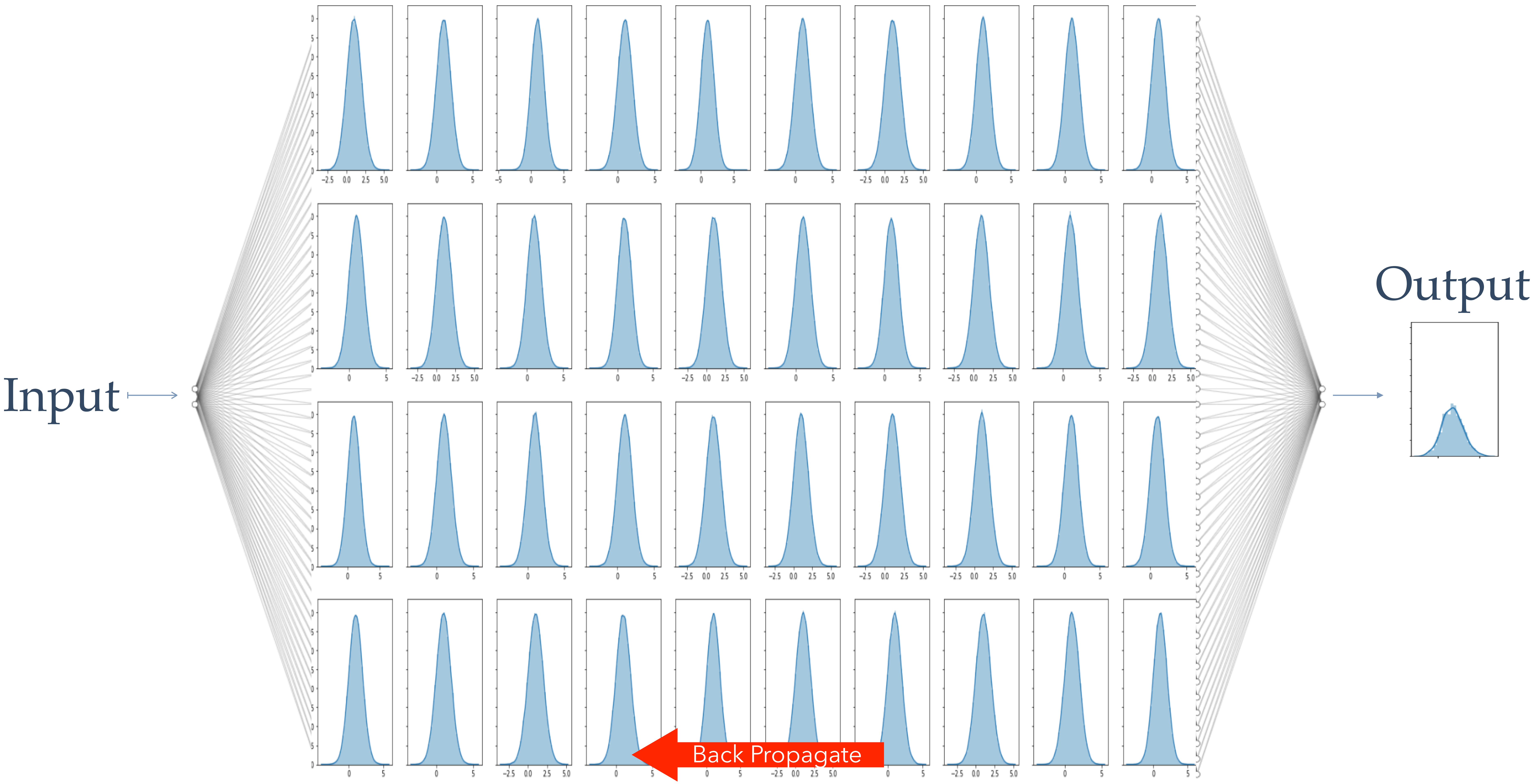
$$\mathscr{L}(y\,|\,\theta) = KL[q(\omega\,|\,\theta))\,||\,p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y\,|\,\omega)]$$

Forward Pass

$$q(\epsilon)$$

sample

$$\omega = \mu + \sigma \odot \epsilon$$



$$\mu = \quad \mu - \alpha \nabla_{\mu} \mathscr{L} \qquad \sigma = \quad \sigma - \alpha \nabla_{\sigma} \mathscr{L}$$

Bayesian Neural Network: Flipout

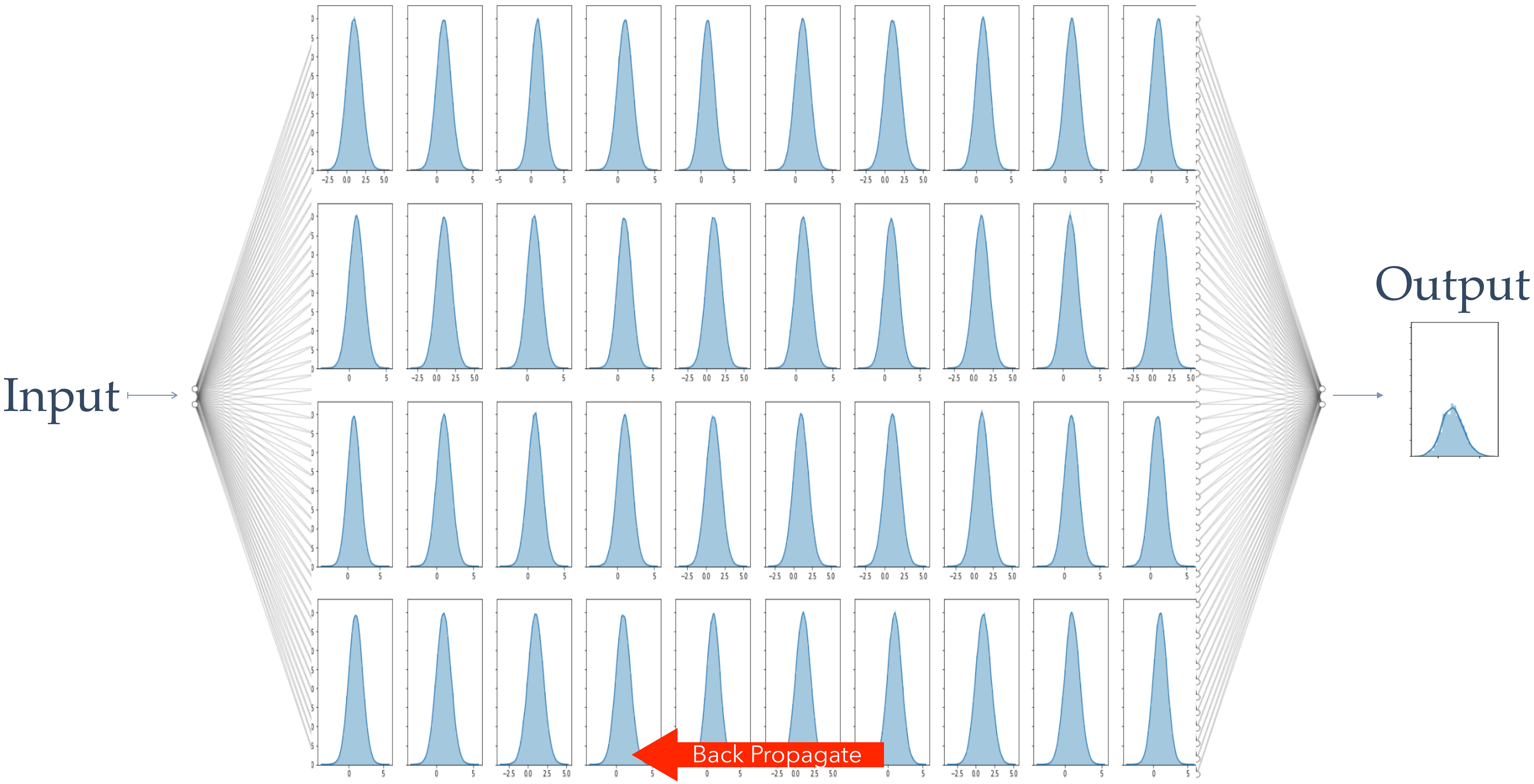
$$\mathcal{L}(y|\theta) = KL[q(\omega|\theta)||p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y|\omega)]$$

Forward Pass

$q(\epsilon)$

sample + flip de coin

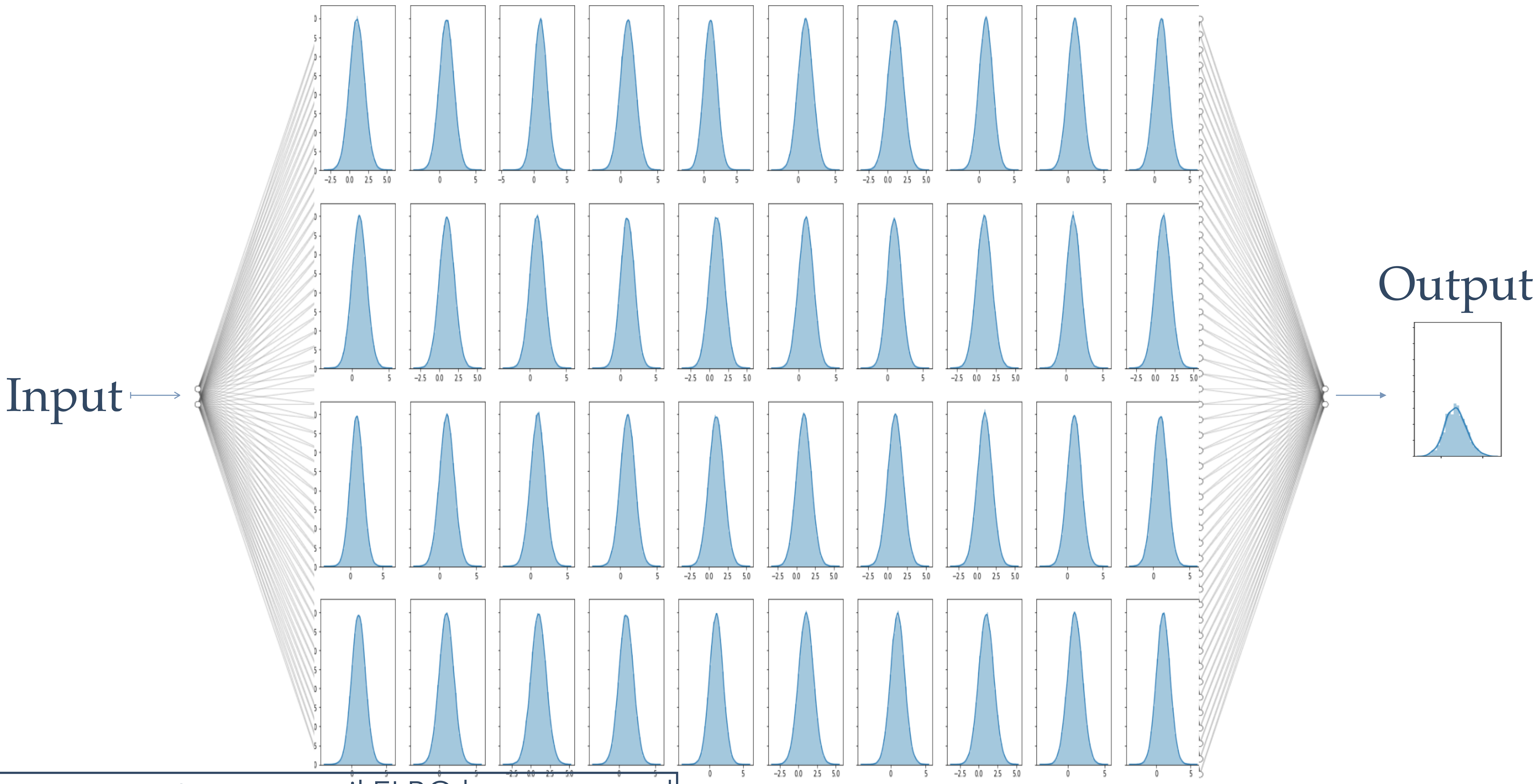
$$\omega = \mu + \sigma \odot (\pm\epsilon)$$



$$\mu = \quad \mu - \alpha \nabla_{\mu}\mathcal{L} \qquad \sigma = \quad \sigma - \alpha \nabla_{\sigma}\mathcal{L}$$

Bayesian Neural Network: MFVB

$$p(\theta | y) \approx q^* = argmax_{q \in Q} ELBO$$



for $j \in \{1,\ldots.,m\}$ do until ELBO has converged

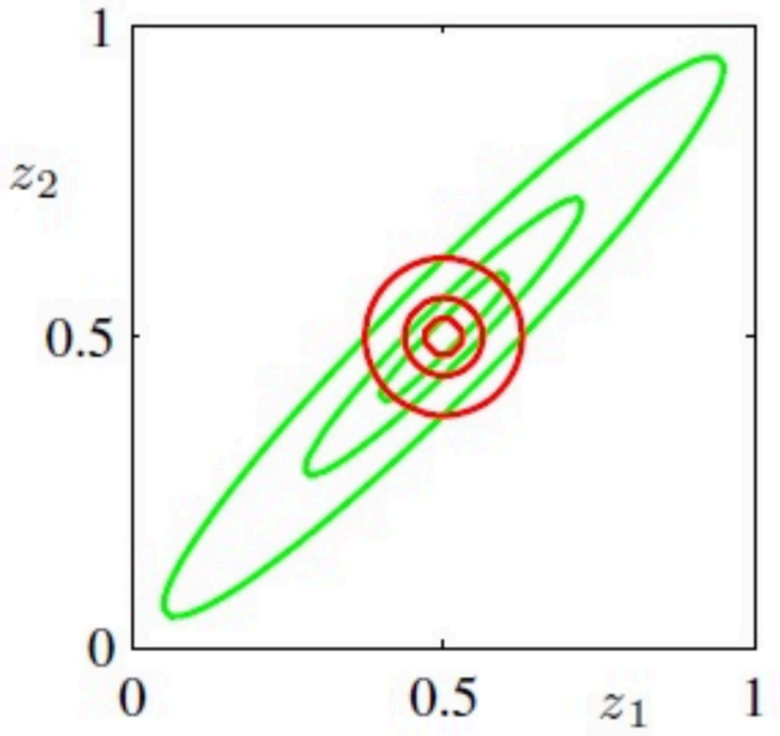
$$q_j(\theta_j) \propto \exp\{\mathbb{E}_{i \neq j}[\log(p(\theta_j | \theta_{i \neq j}, y)]\}$$

compute ELBO(q)

Variational Bayesian Inference: The problem

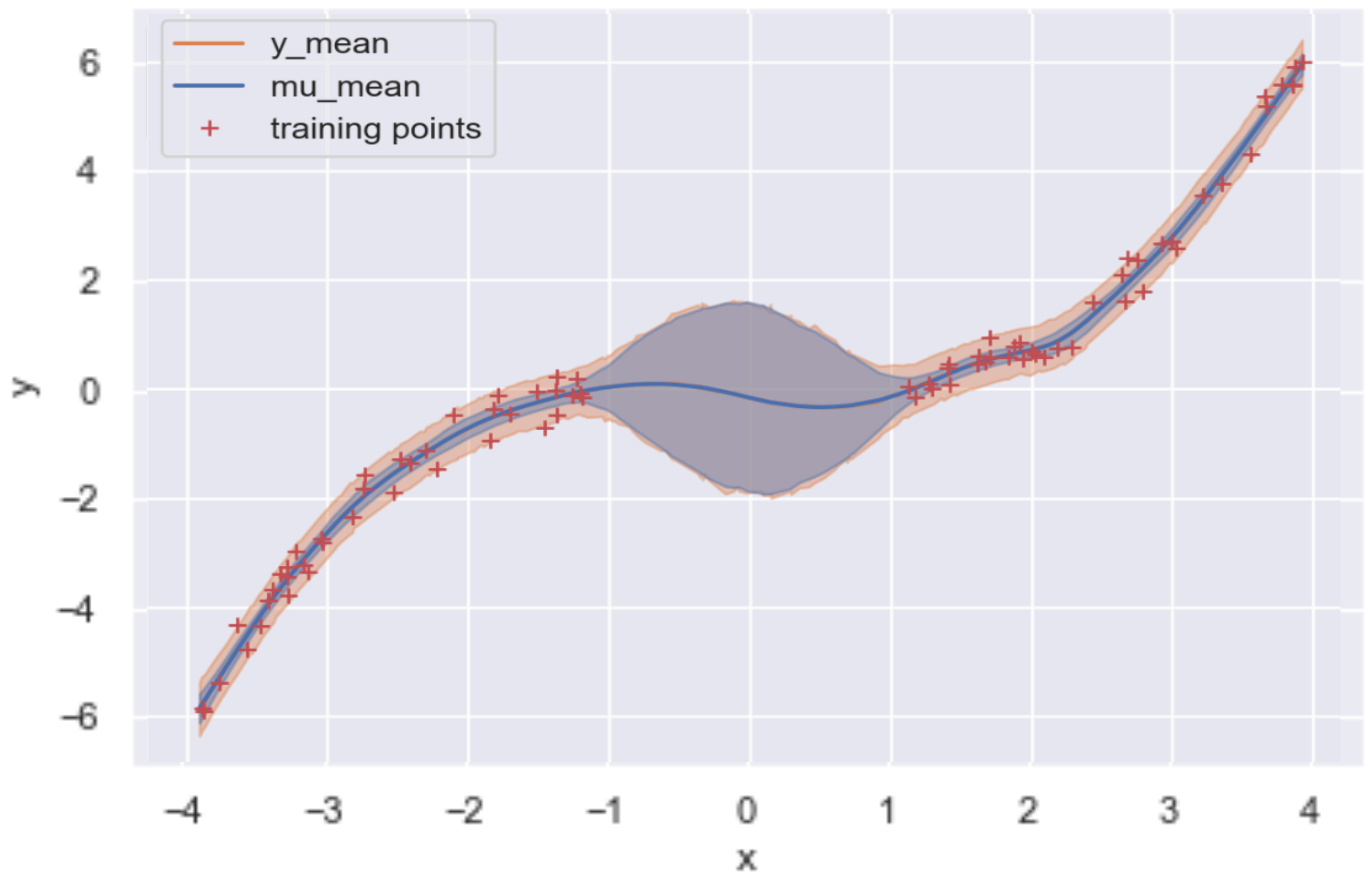
Variational Bayesian Inference: The right solution (MCMC)

Variational Bayesian Inference

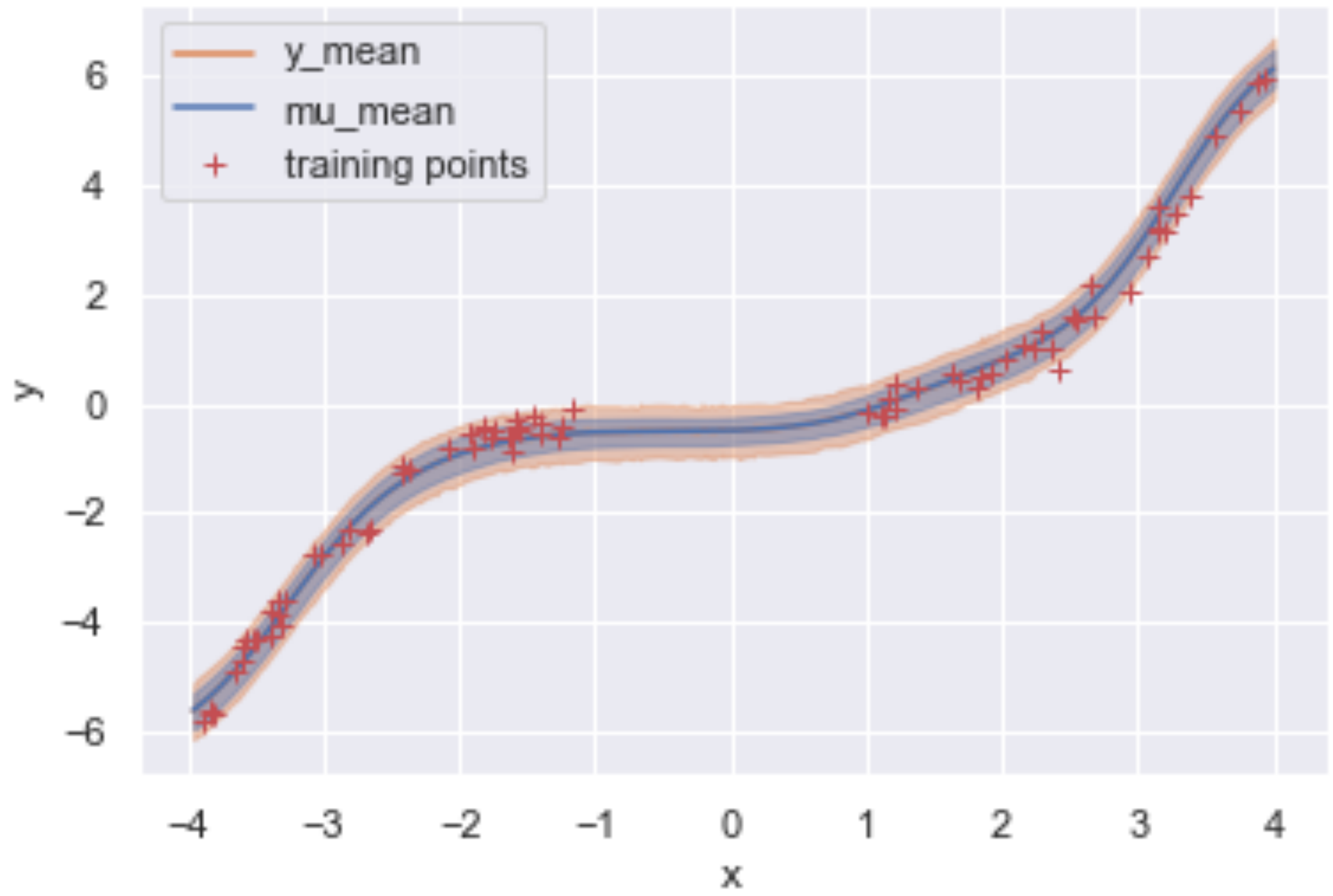


(a)

MCMC



MFVB

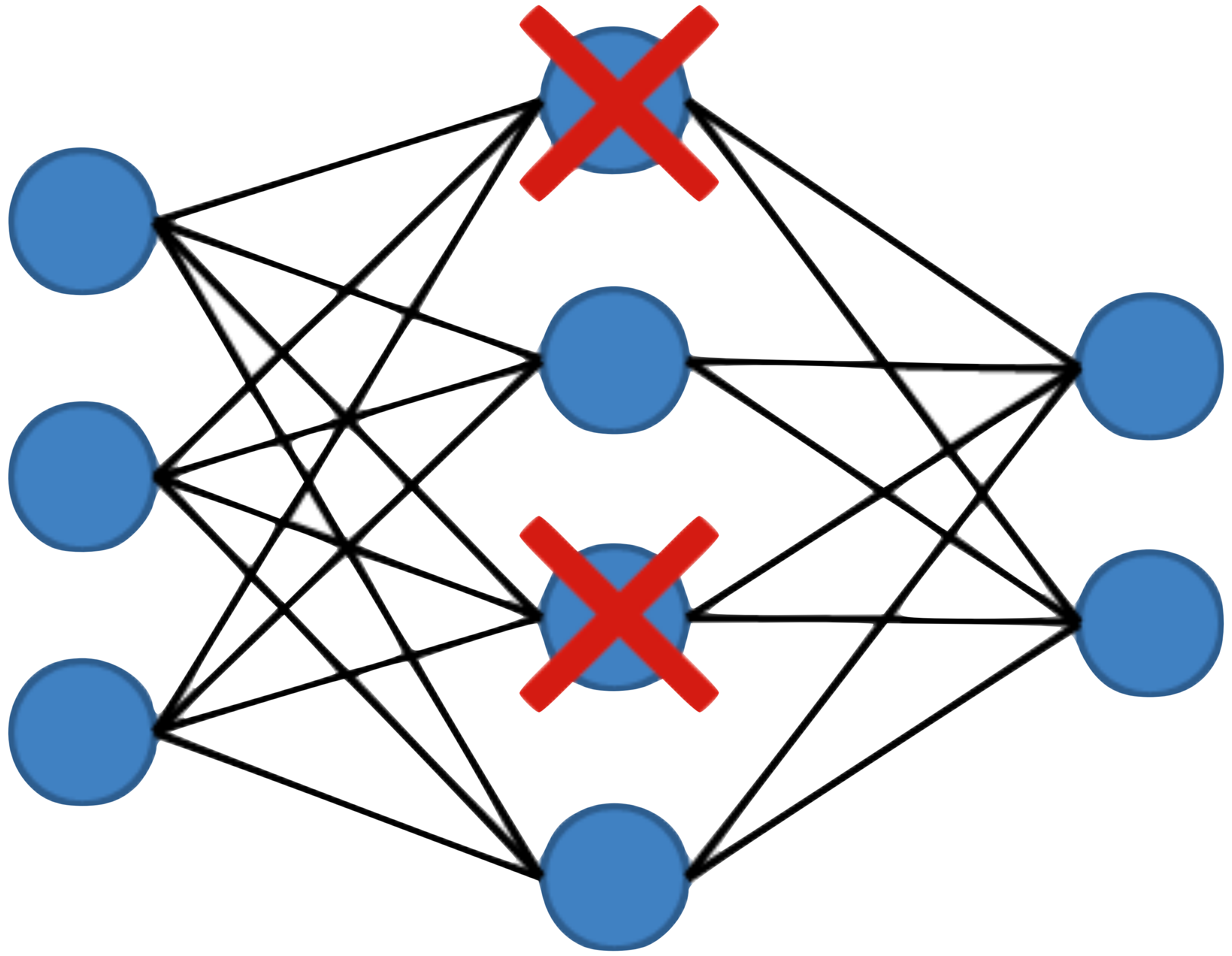


Dropout

arXiv:1506.02142

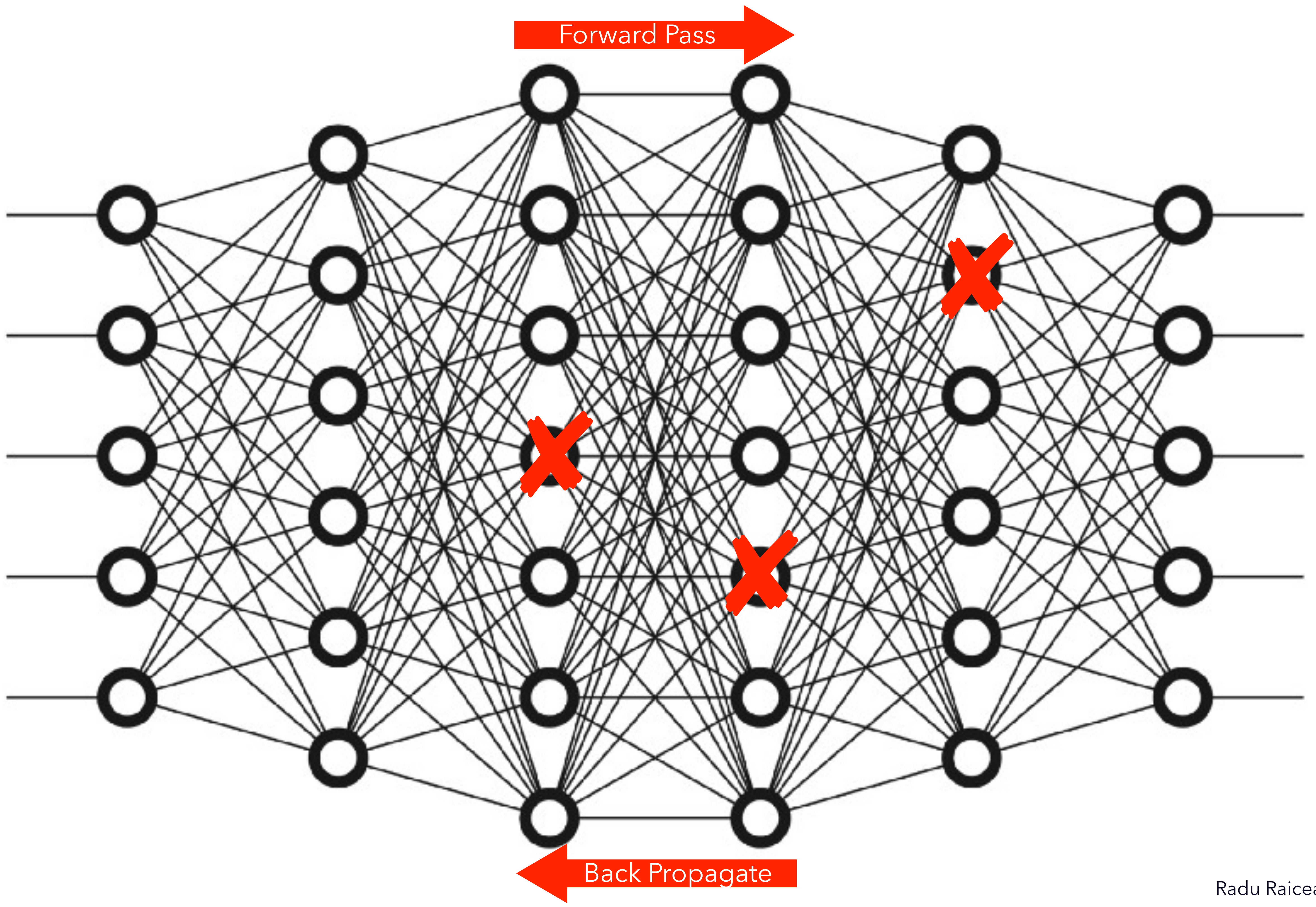
Dropout as a Bayesian Approximation:
Representing Model Uncertainty in Deep Learning

Yarin Gal YG279@CAM.AC.UK
Zoubin Ghahramani ZG201@CAM.AC.UK
University of Cambridge



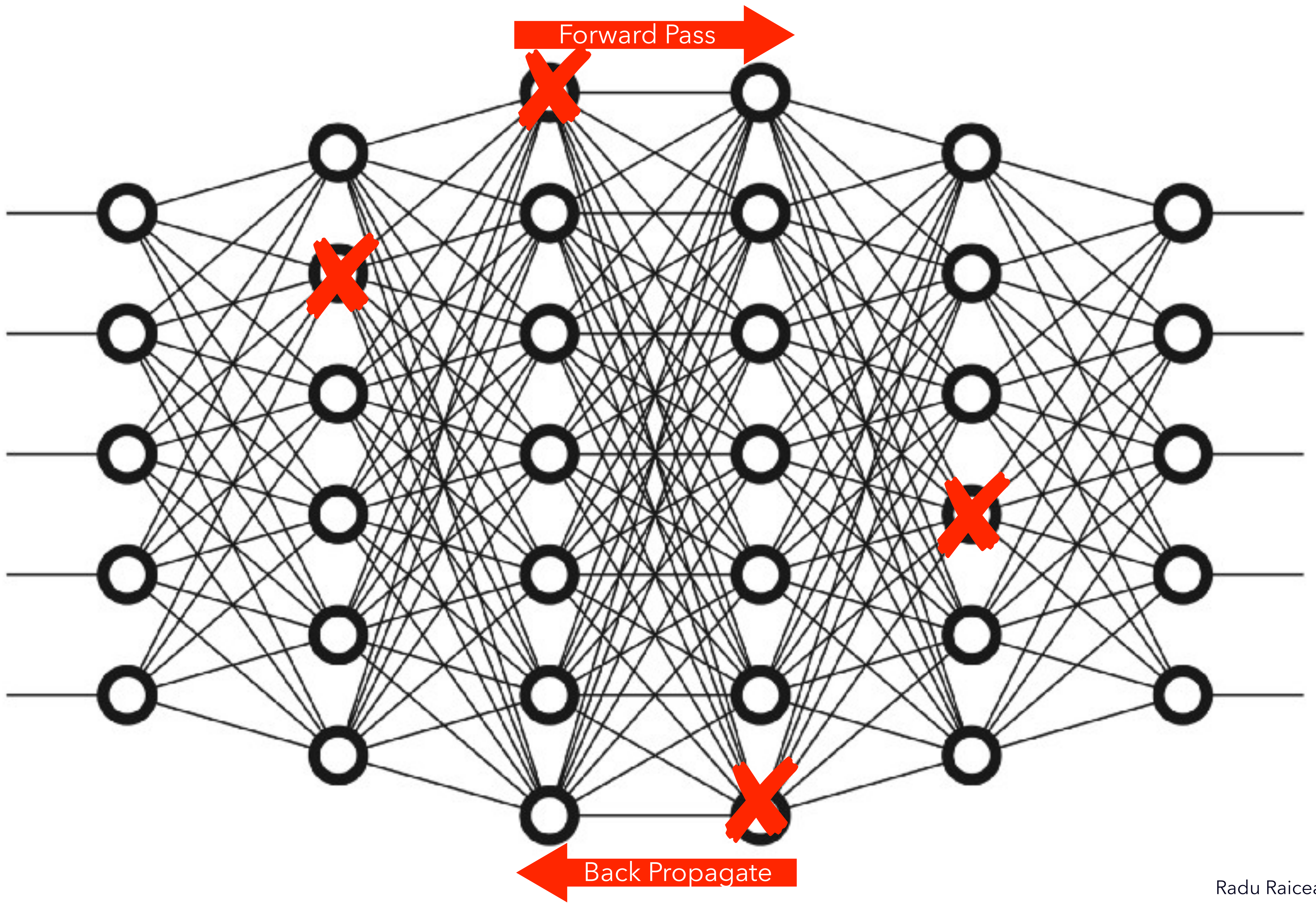
They show that a NN with arbitrary depth and non-linearities, with dropout applied before every weight layer, is mathematically equivalent to an approximation to the probabilistic deep Gaussian process.

Dropout: train



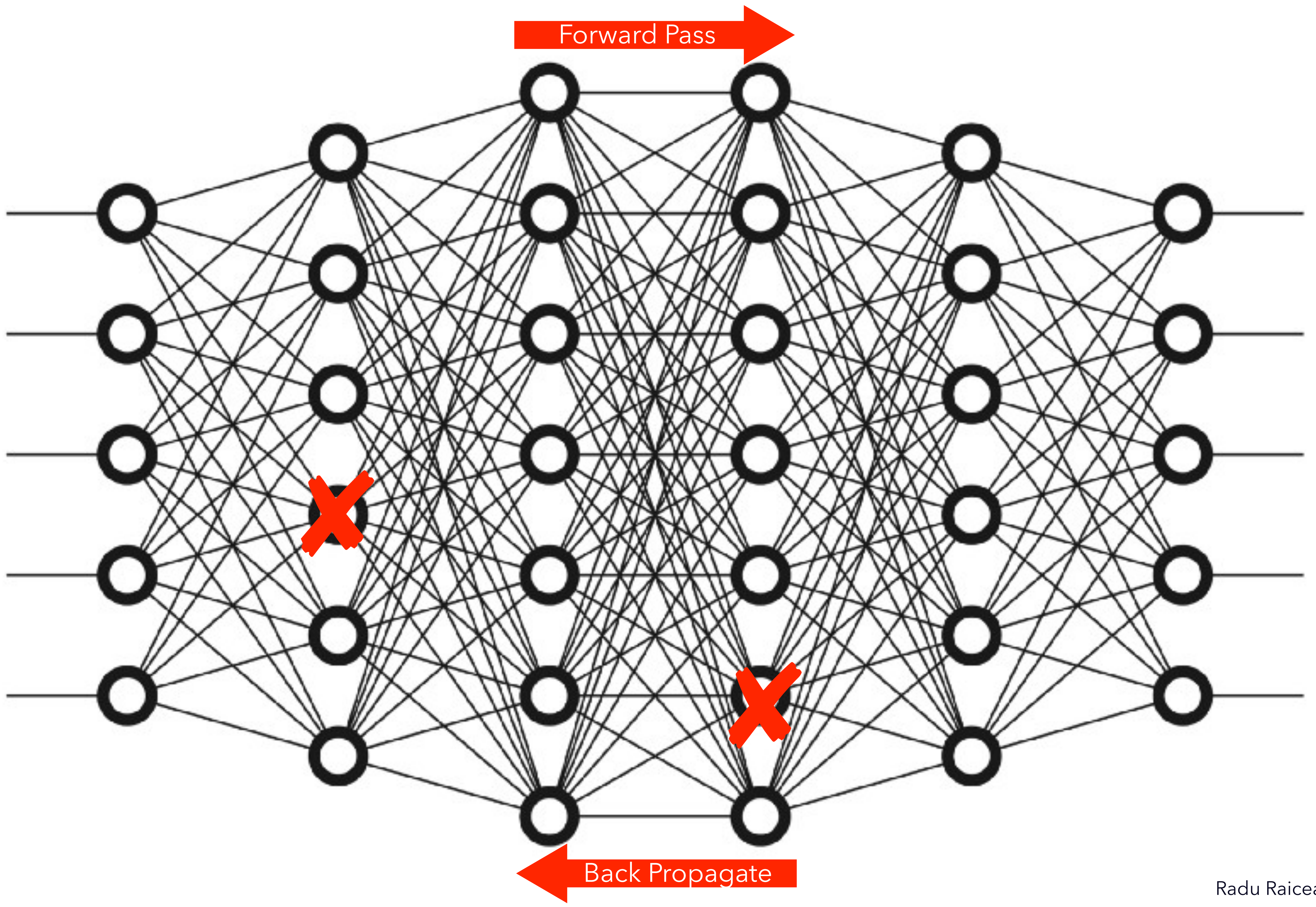
Radu Raicea

Dropout: train



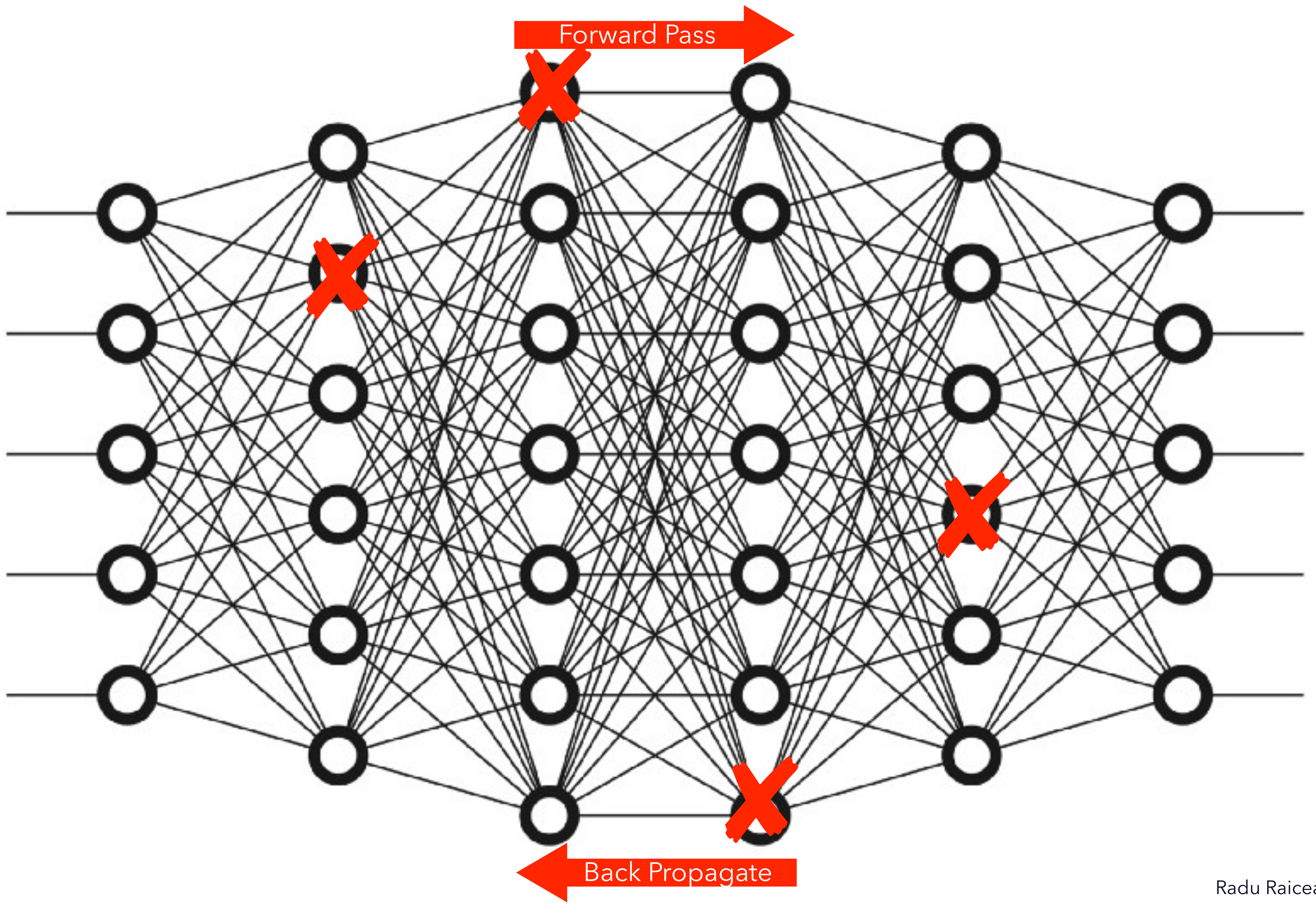
Radu Raicea

Dropout: train



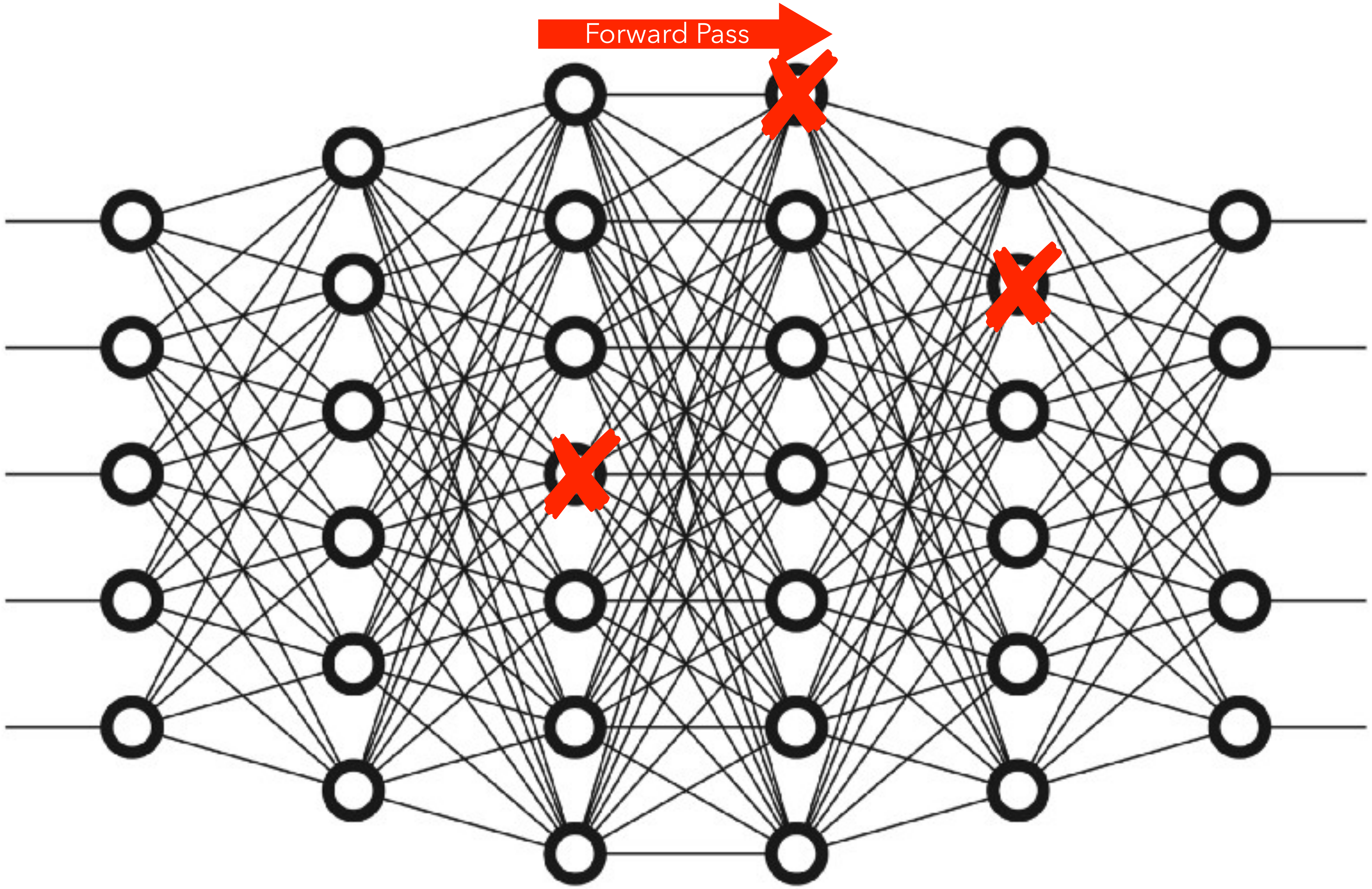
Radu Raicea

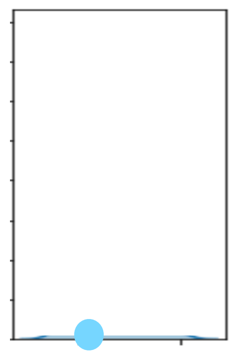
Dropout: train



Radu Raicea

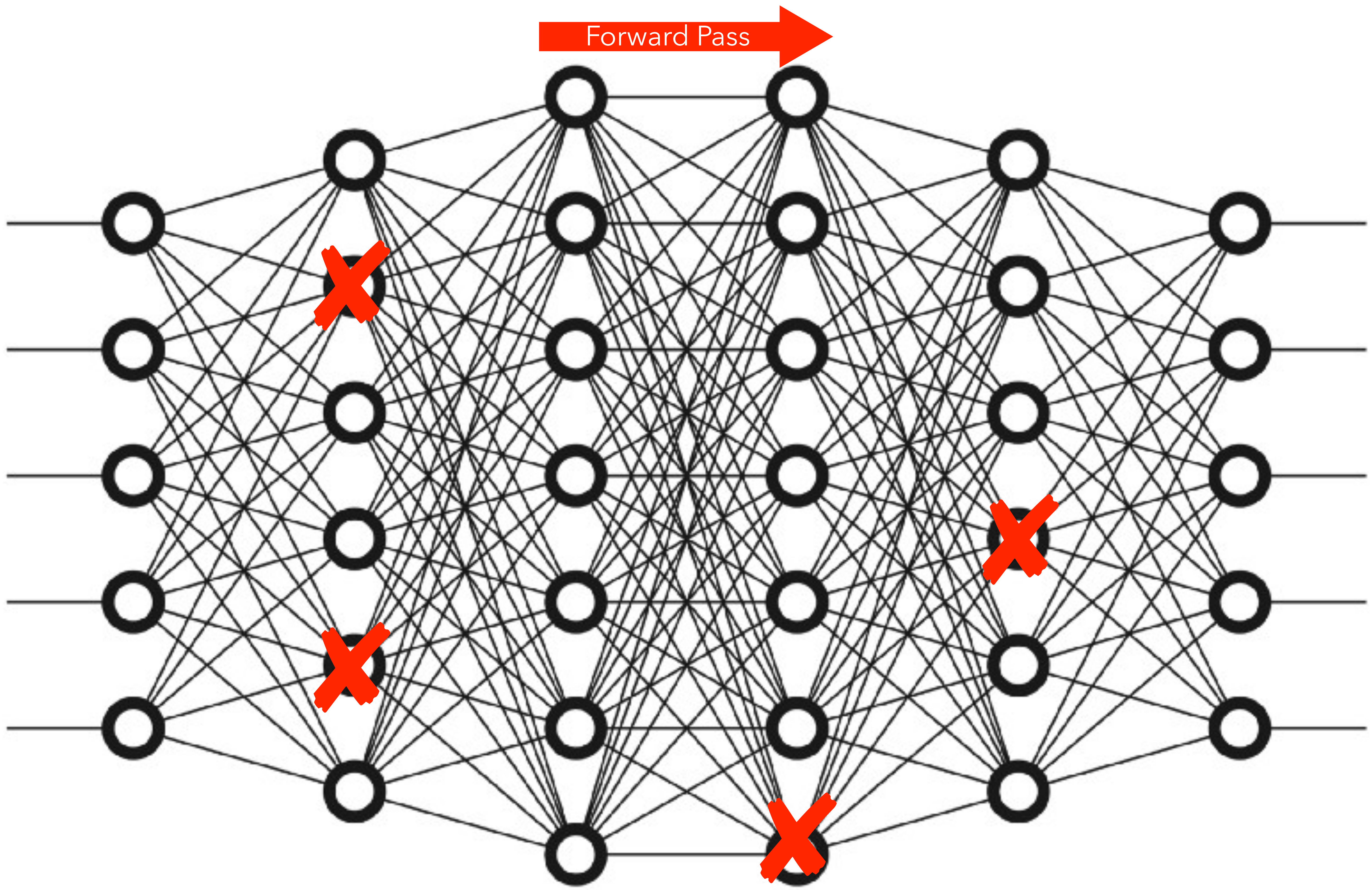
Dropout: evaluate

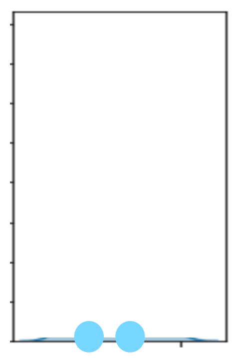




Radu Raicea

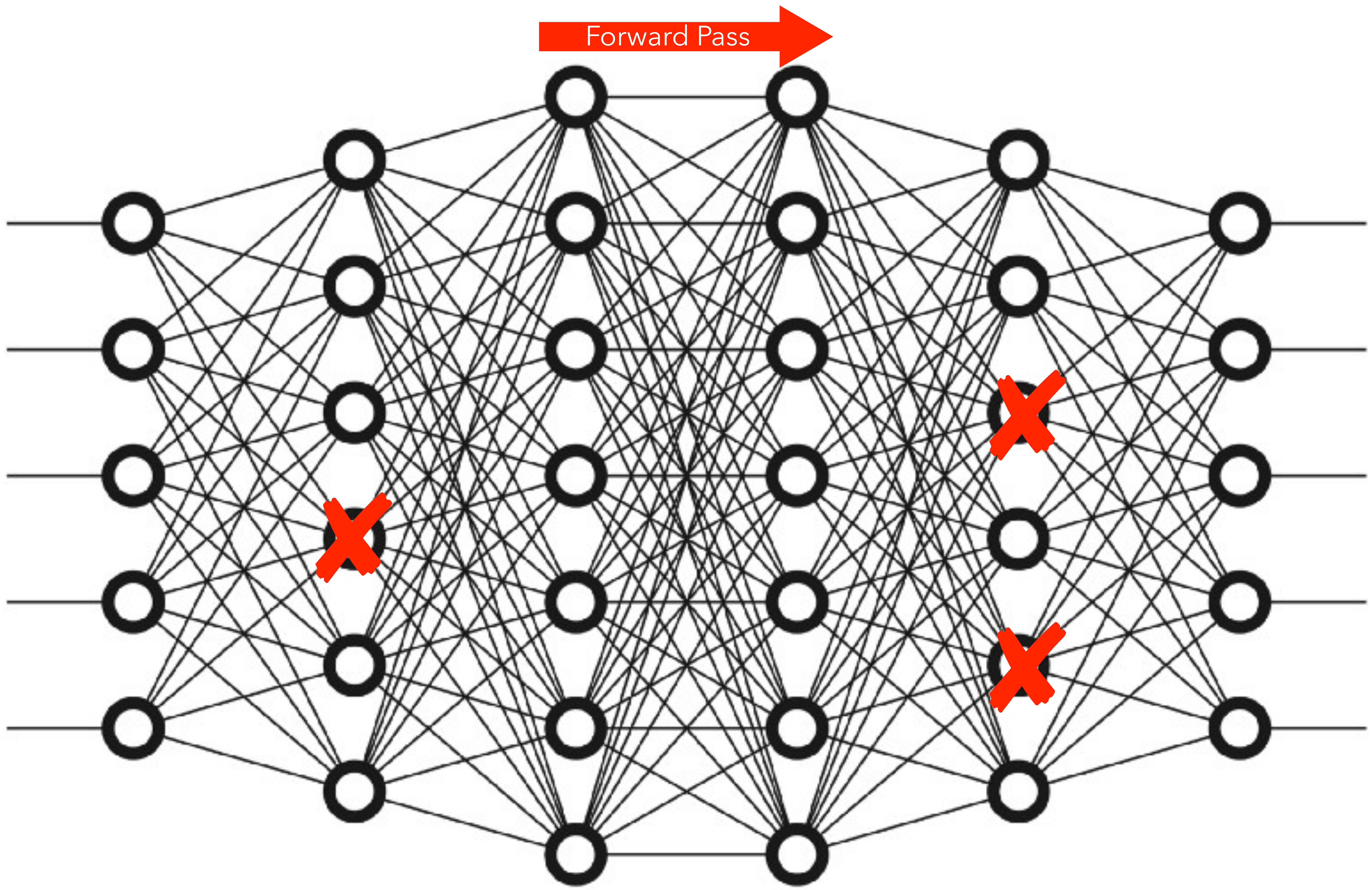
Dropout: evaluate

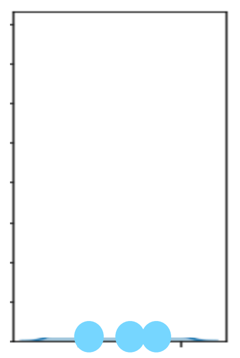




Radu Raicea

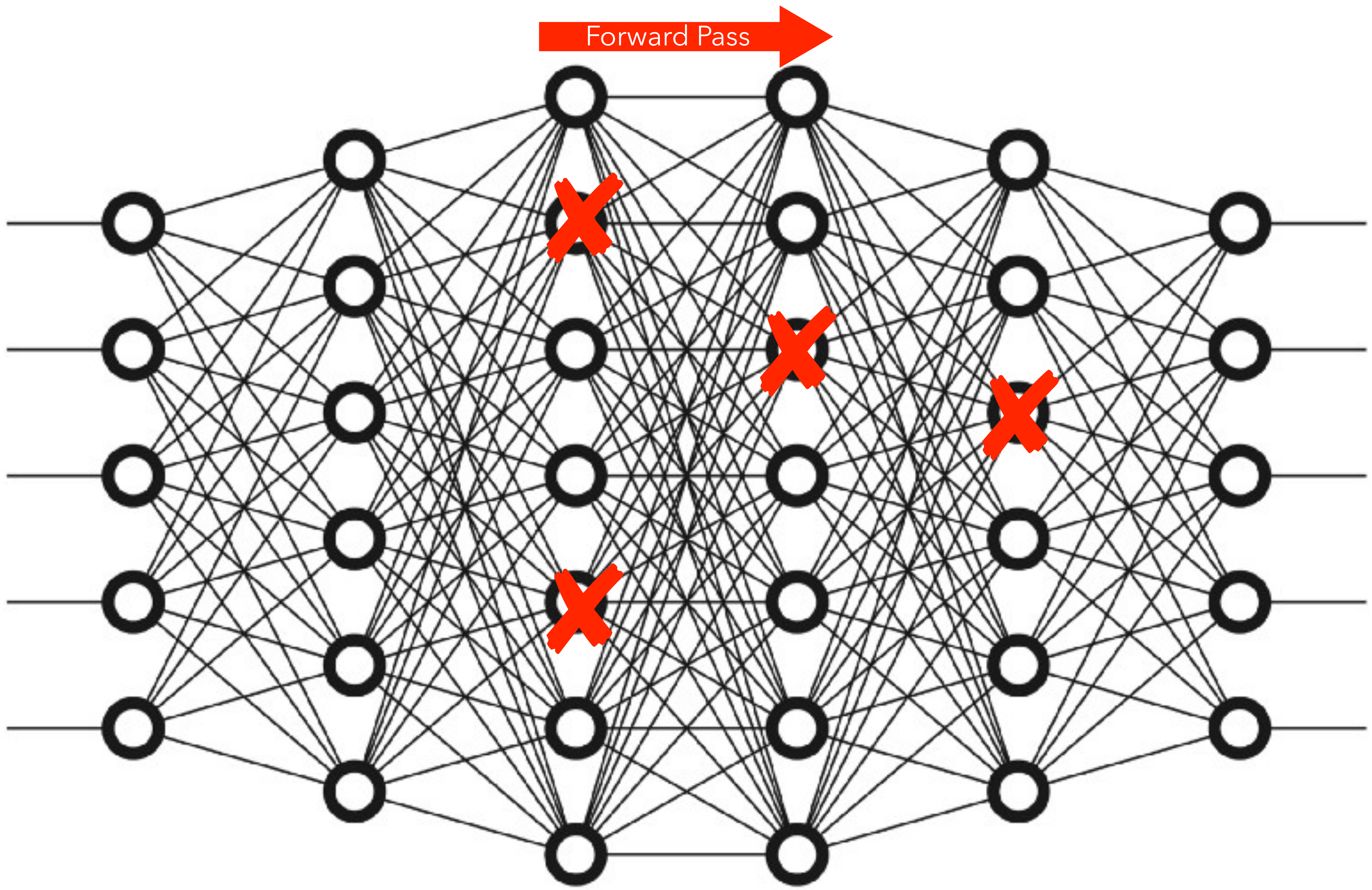
Dropout: evaluate

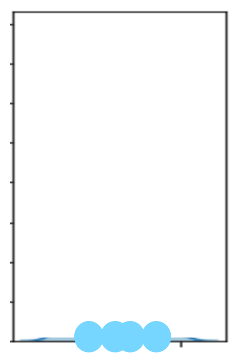




Radu Raicea

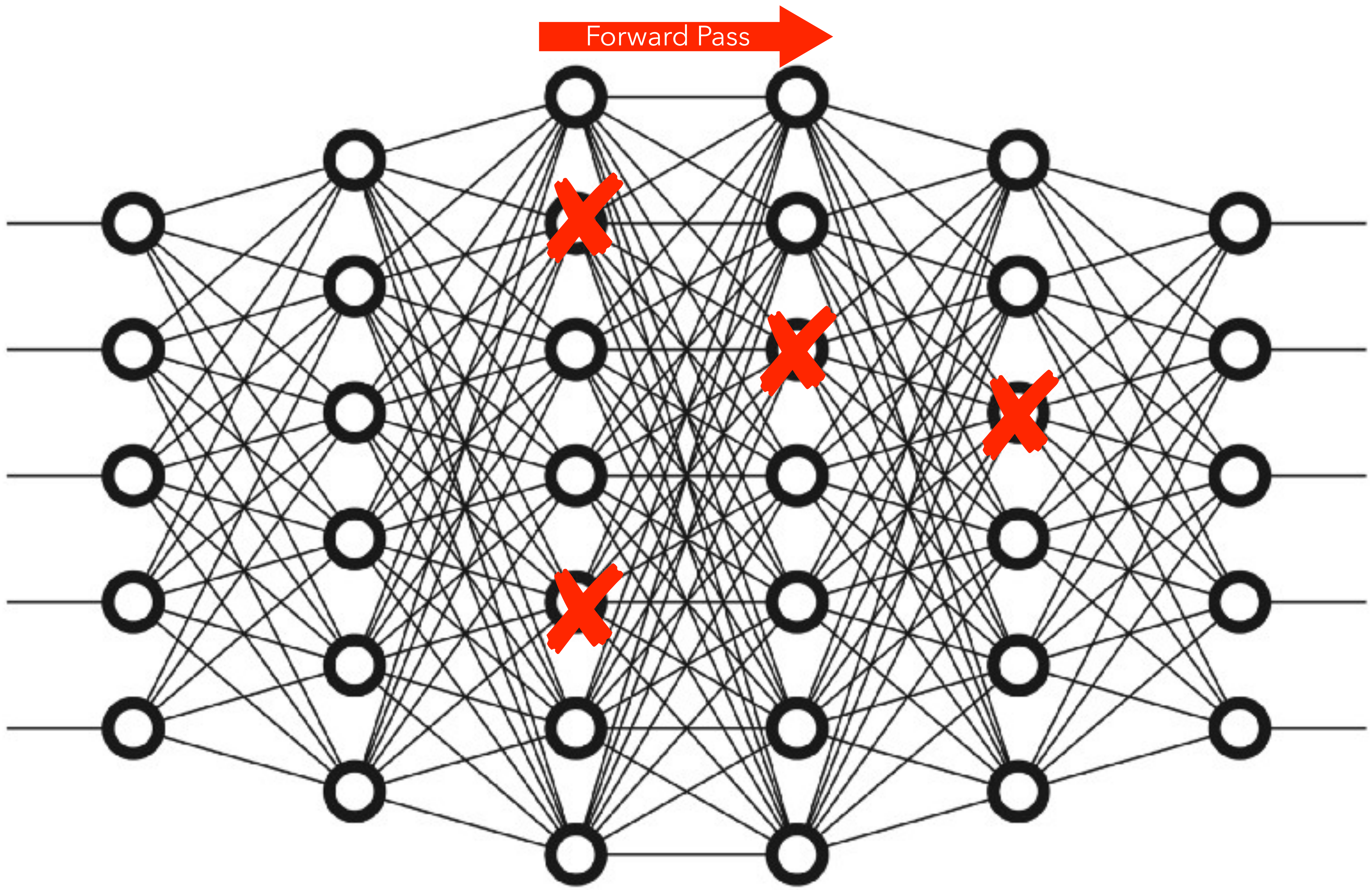
Dropout: evaluate

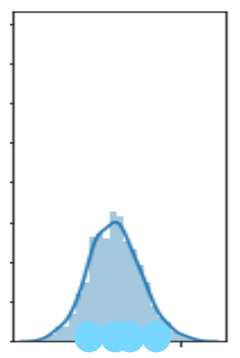




Radu Raicea

Dropout: evaluate



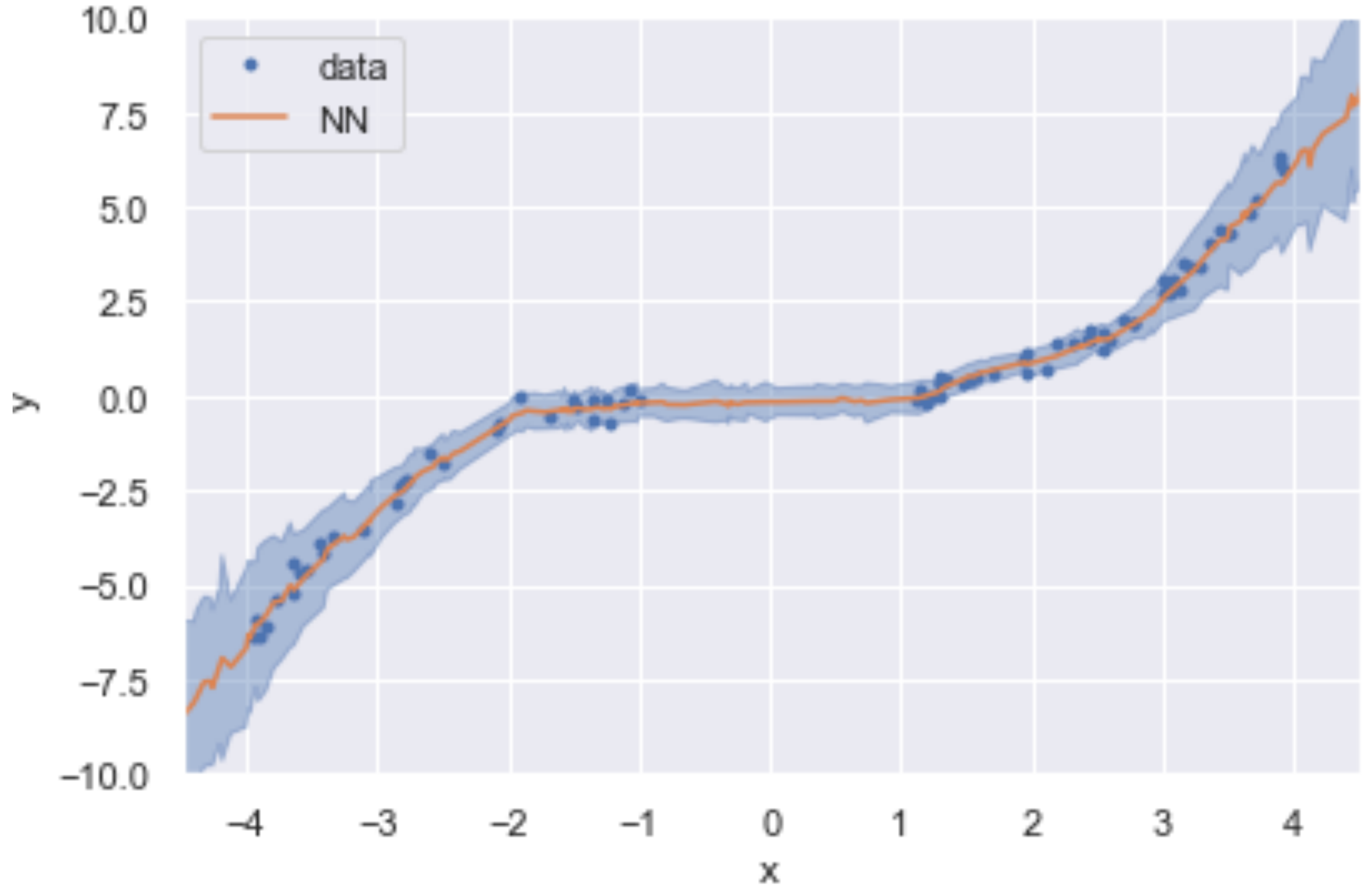


Radu Raicea

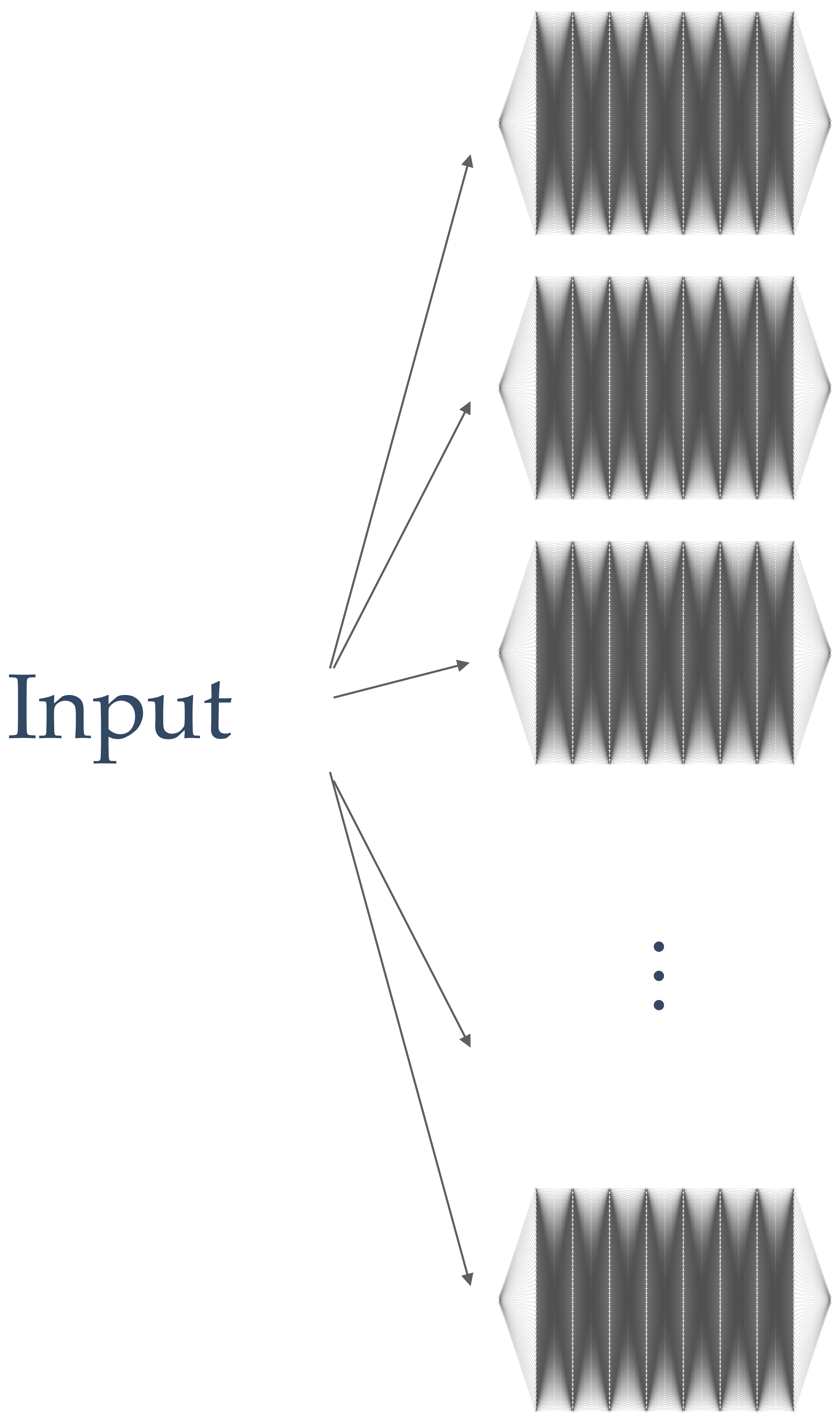
Dropout

MCMC

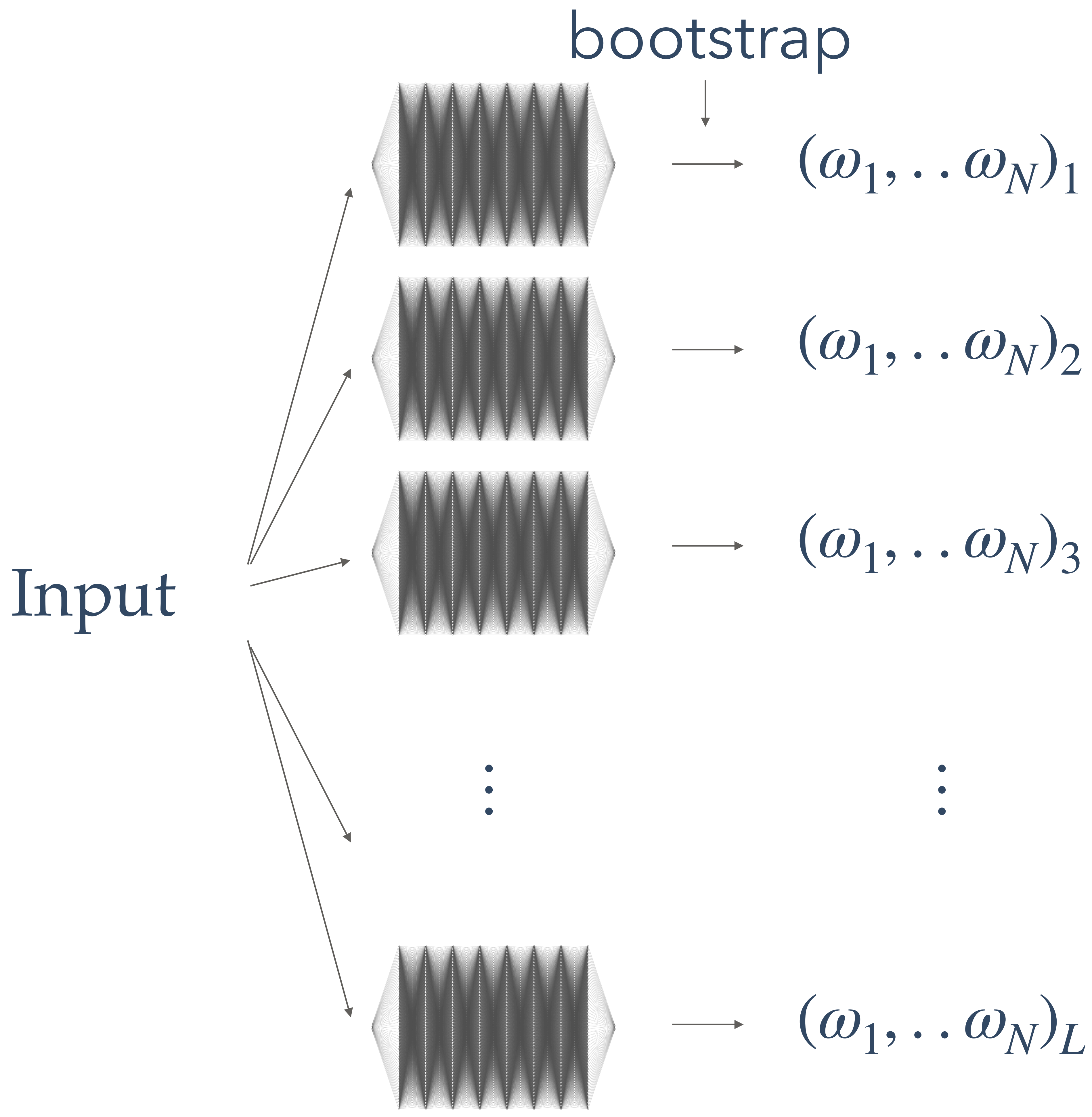
Dropout



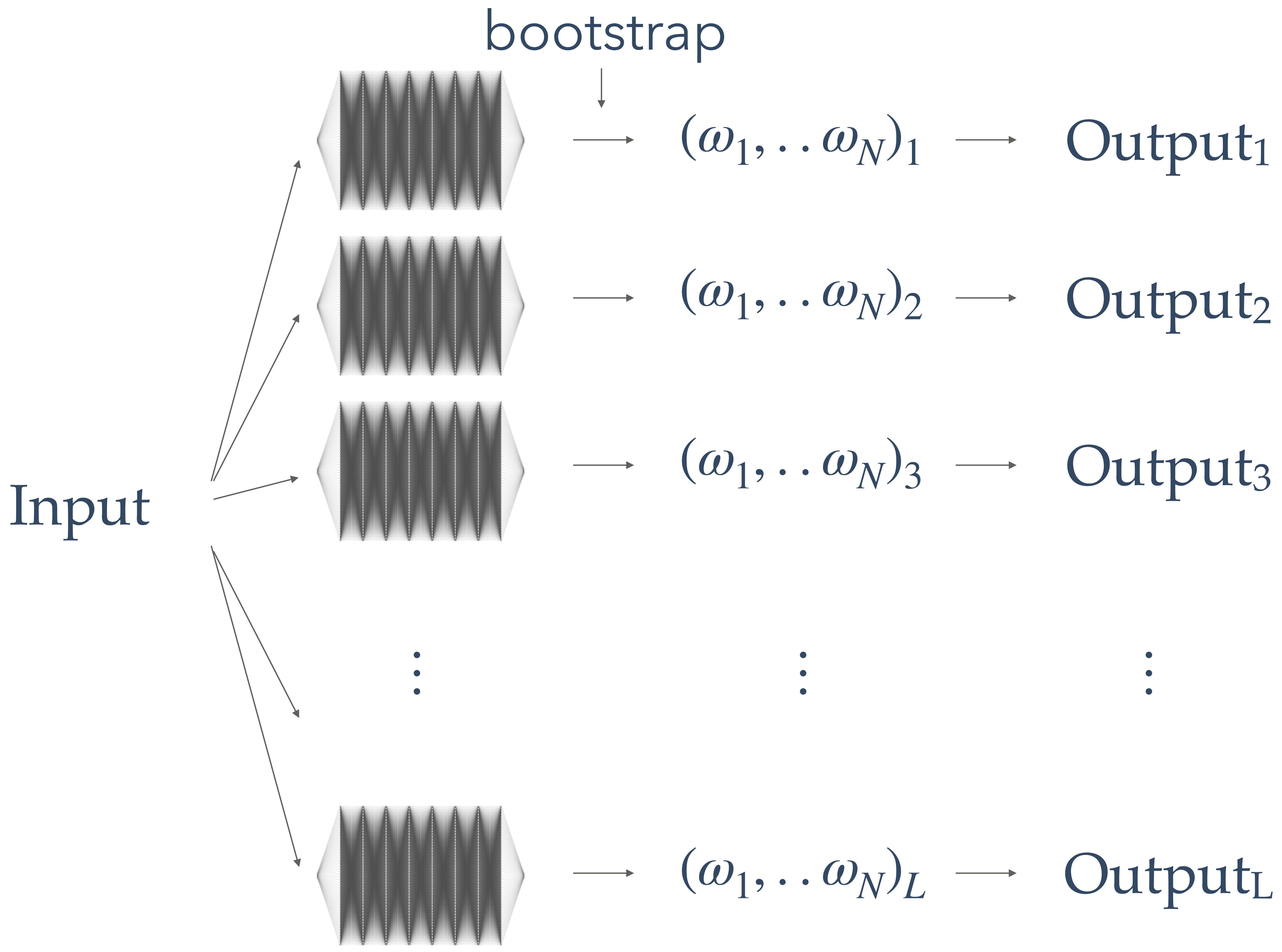
Bootstrap: L models



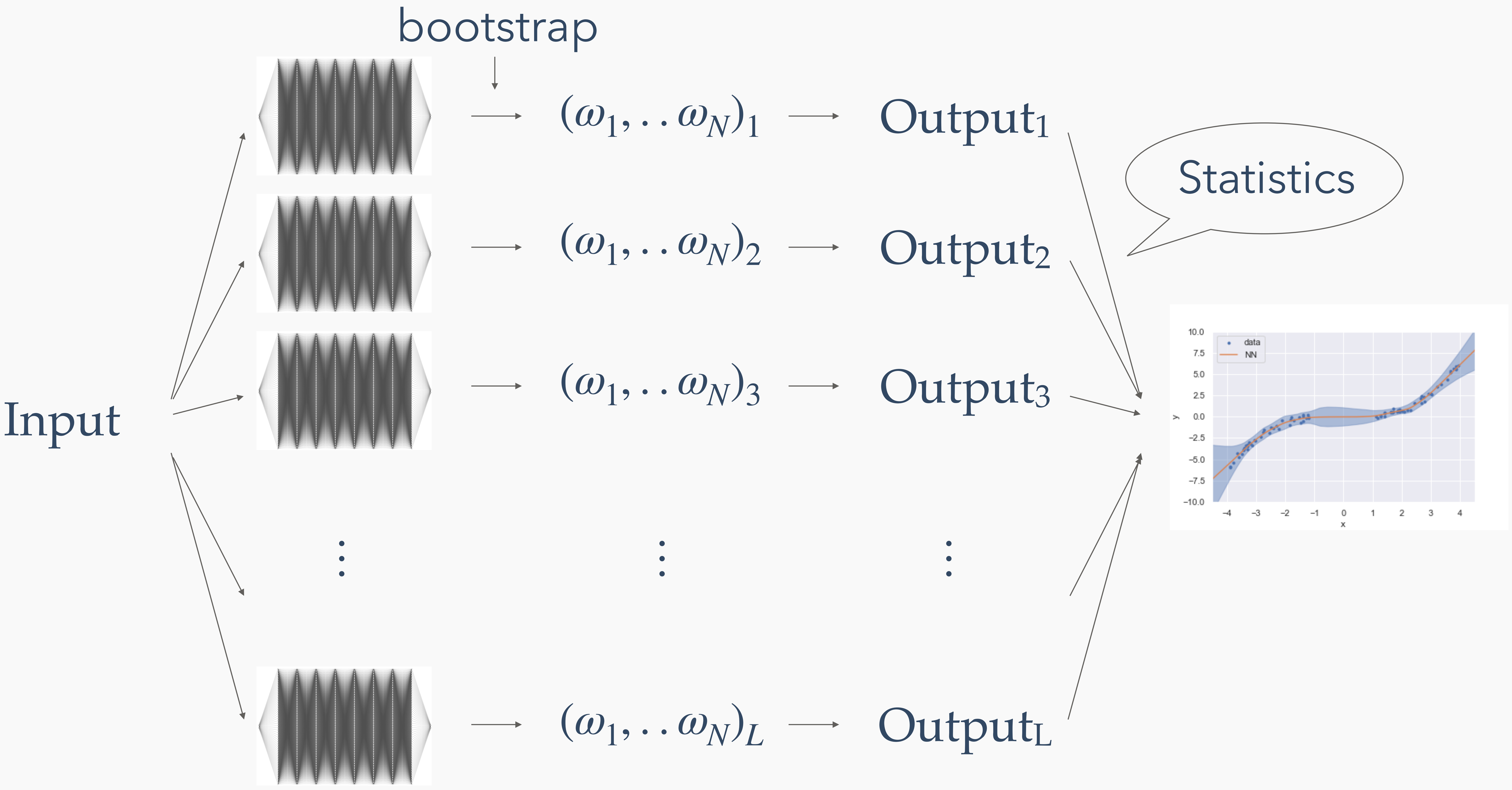
Bootstrap: L models



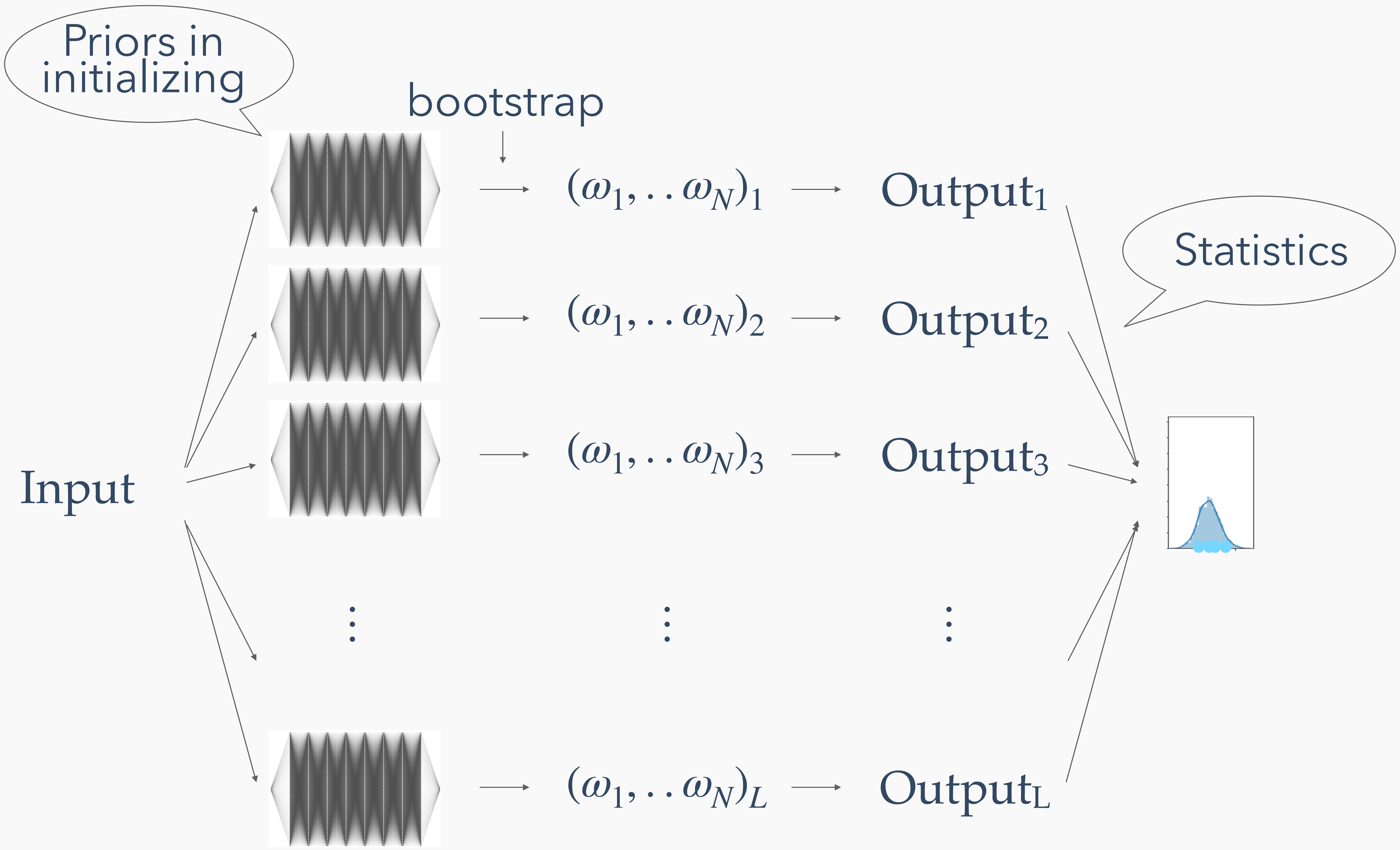
Bootstrap: L models



Bootstrap: L models

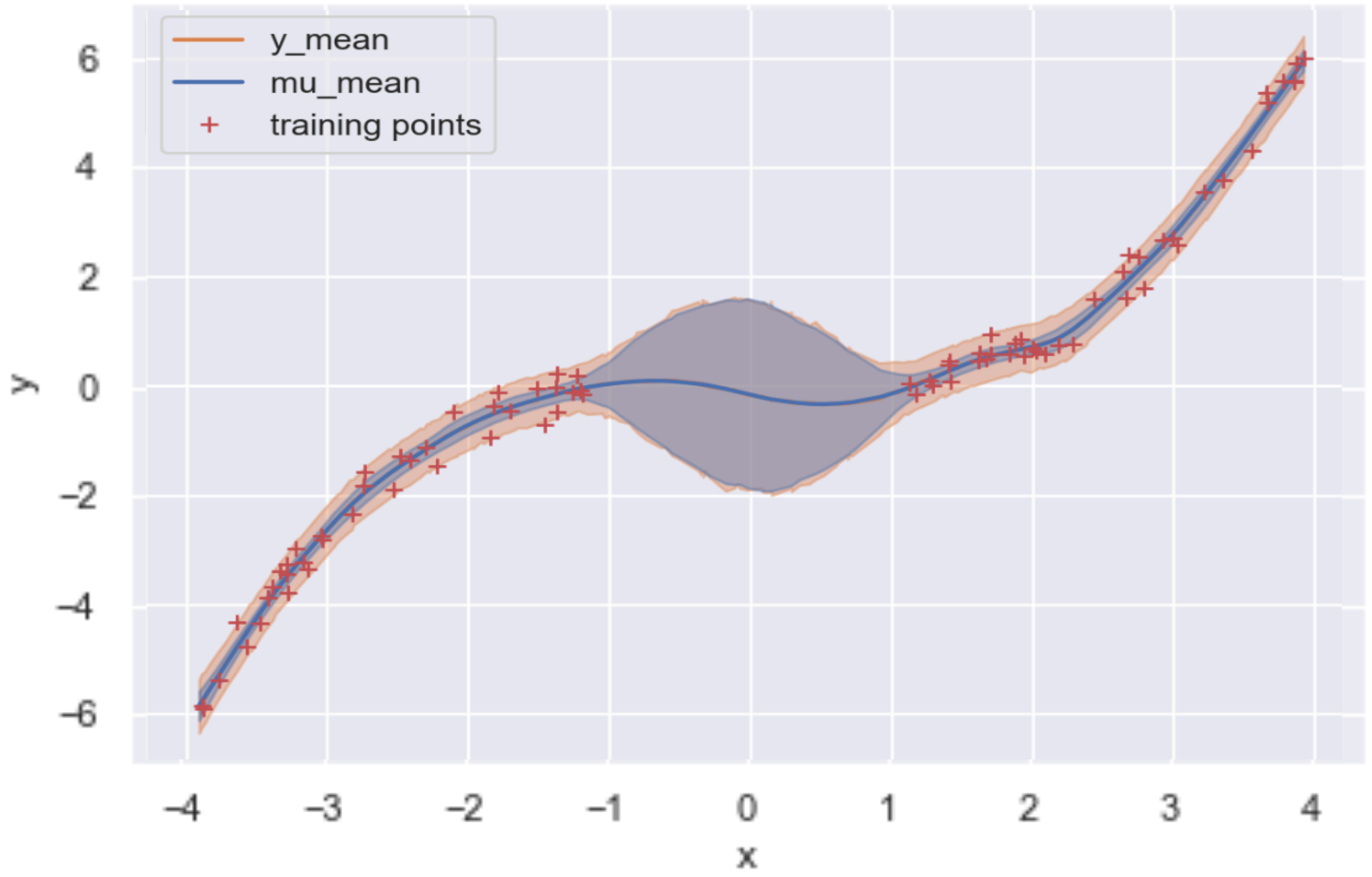


Bootstrap: L models



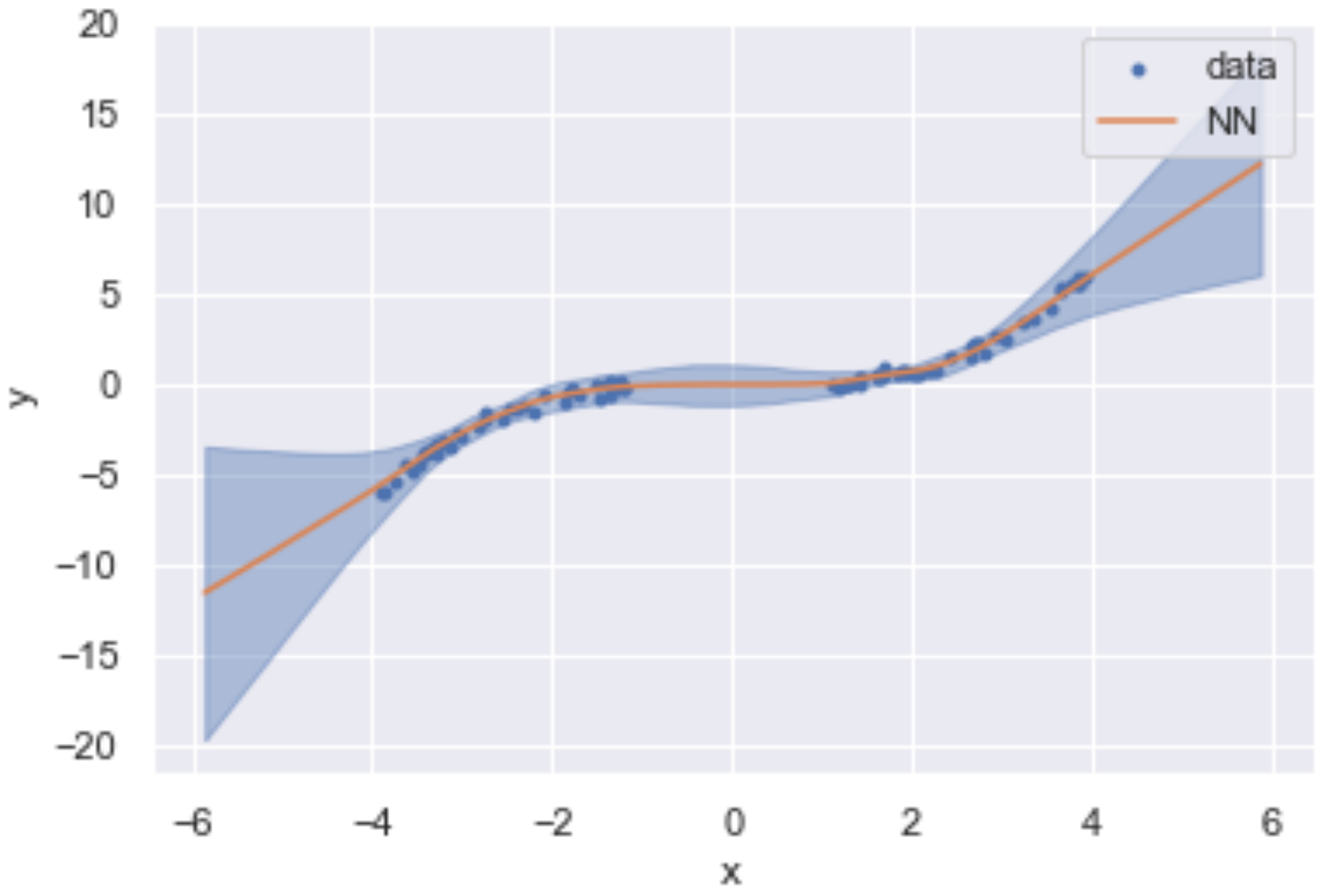
Bootstrap: L models

MCMC



Bootstrap

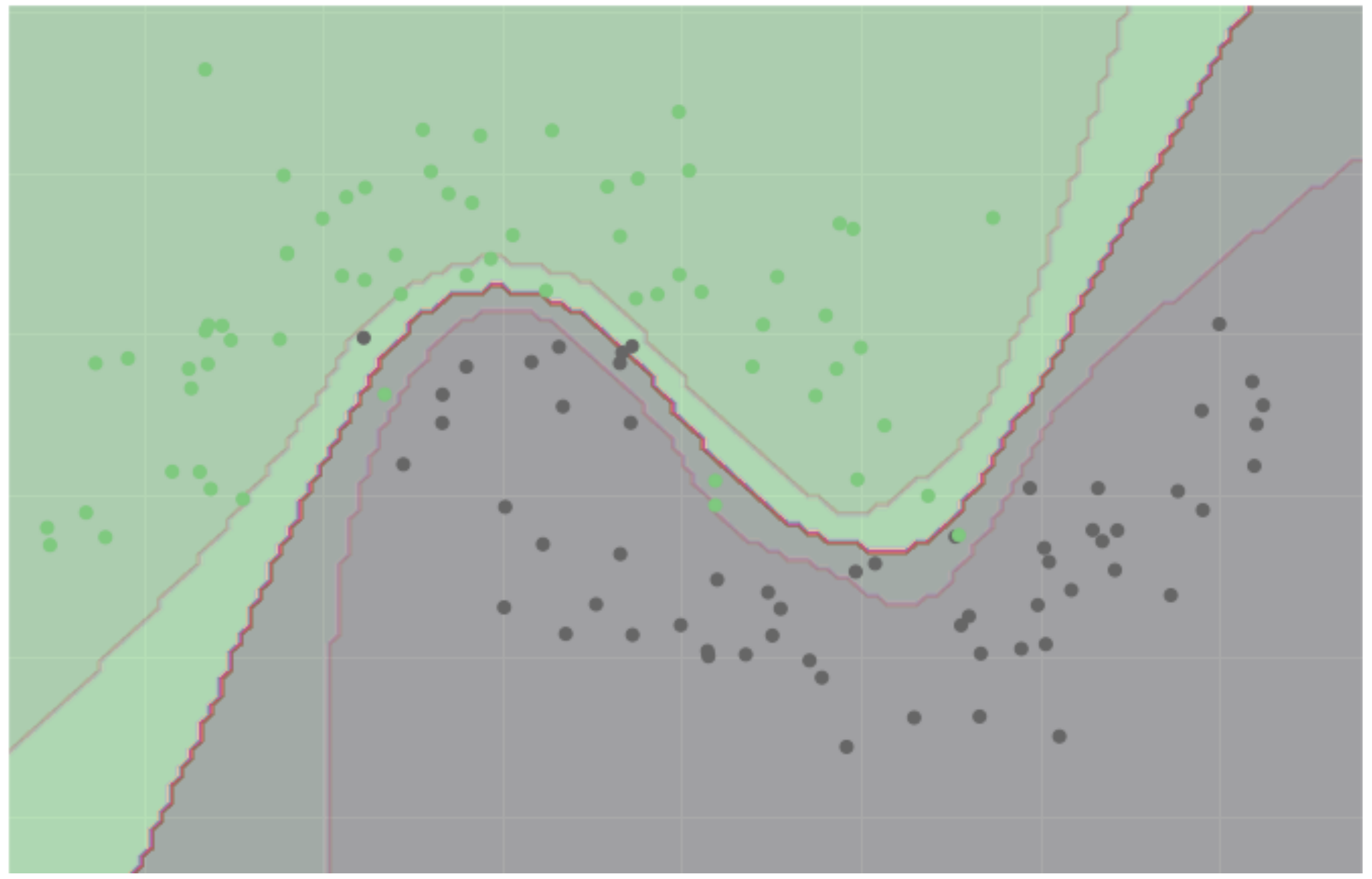
Regression



Model Mean

95% models

Classification



Fin de Recap

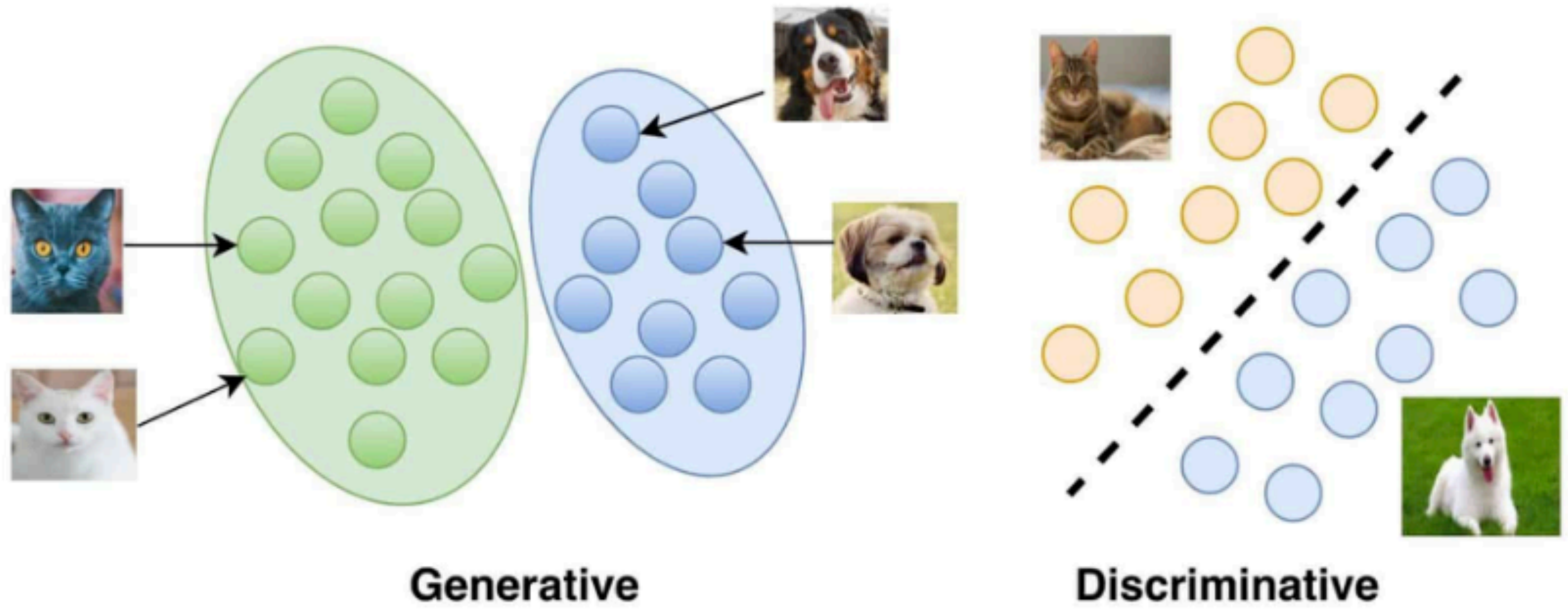
Generative AI for Astrophysics

Slide from Carolina Cuesta Lázaro

Modelos Generativos

$$p(x)$$

$$p(y|x)$$

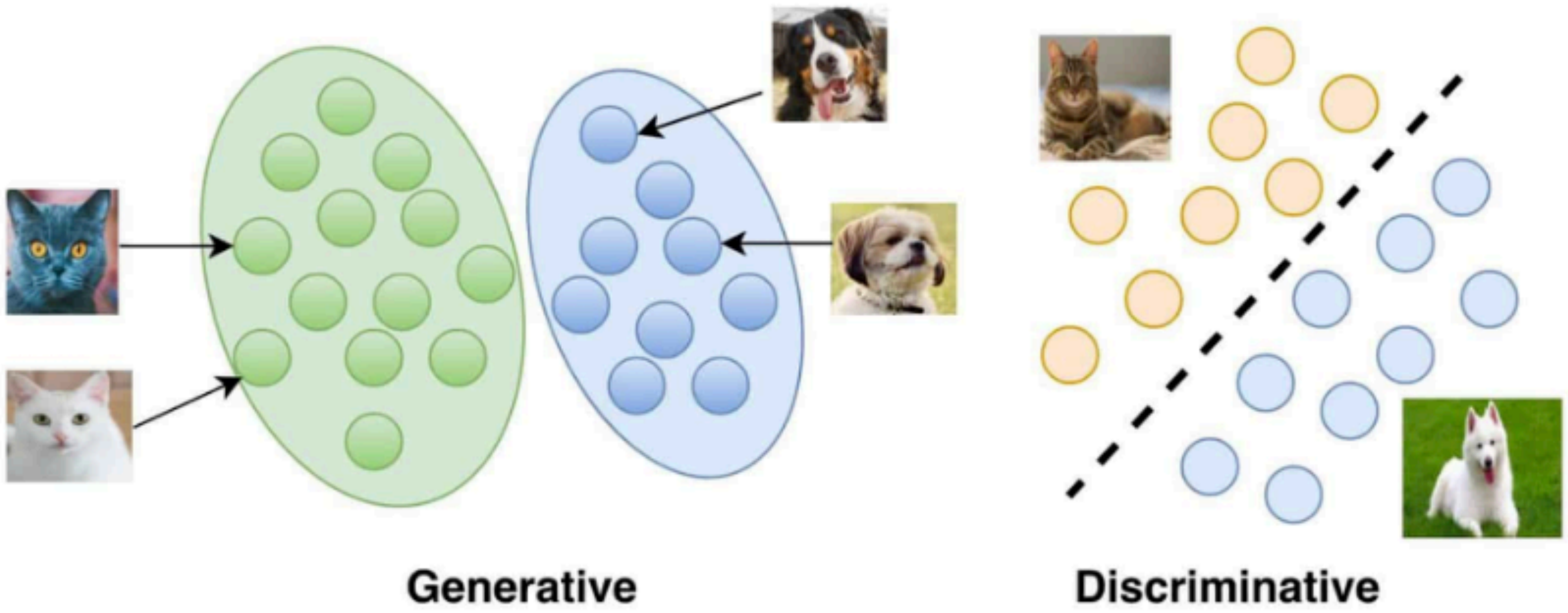


$$p(x|y)$$

Modelos Generativos

$$p(x)$$

$$p(y|x)$$



$$p(x|y)$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Modelos Generativos

$$p_\phi(x)$$

Modelos Generativos

Data

$$p_\phi(\boxed{x})$$

Slide from Carolina Cuesta Lázaro

Modelos Generativos

Data

$$p_{\phi}(\boxed{x})$$

A PDF that we can optimize

Slide from Carolina Cuesta Lázaro

Modelos Generativos

Maximize the likelihood of the data

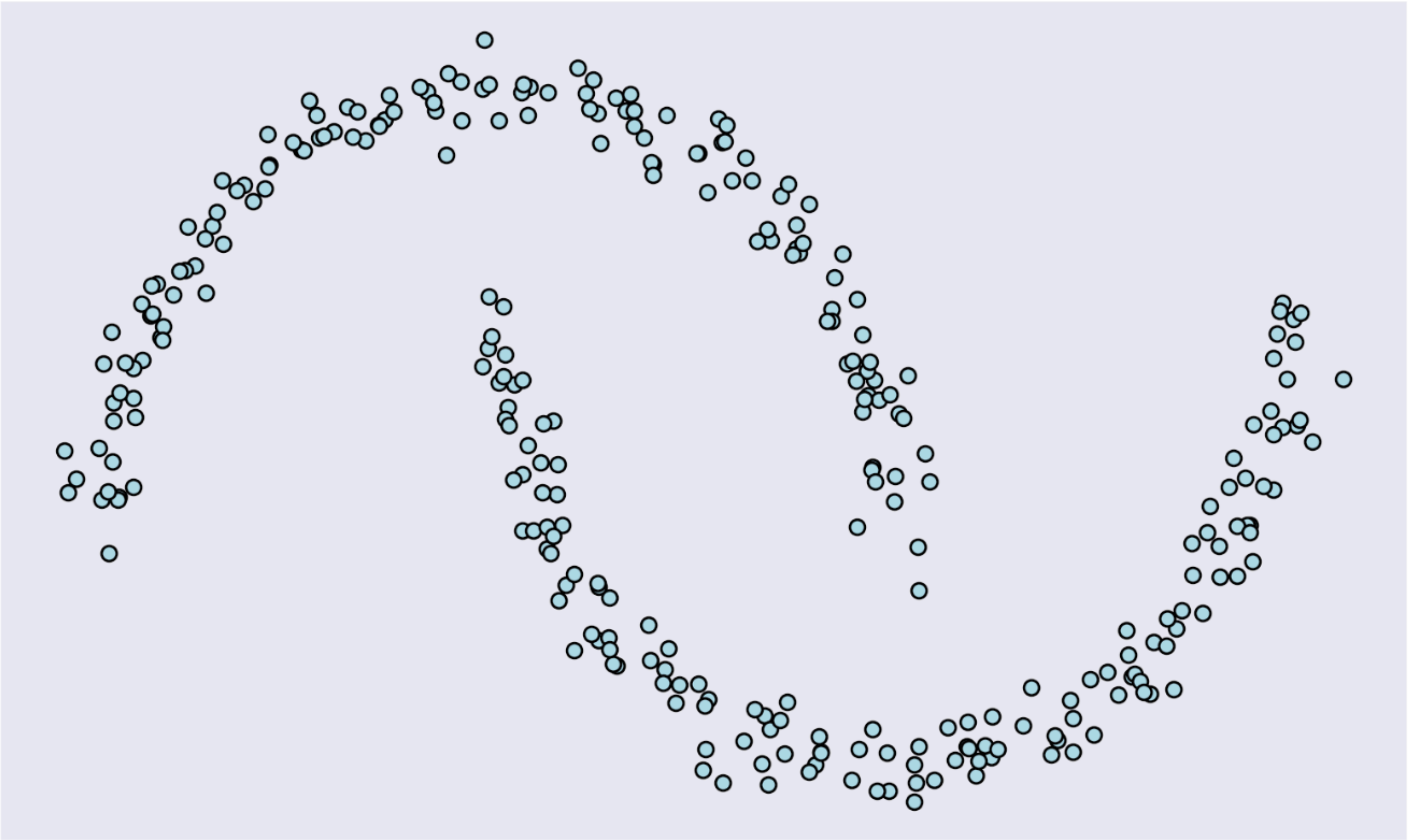
Data

$$\boxed{p_{\phi}(\boxed{x})}$$

A PDF that we can optimize

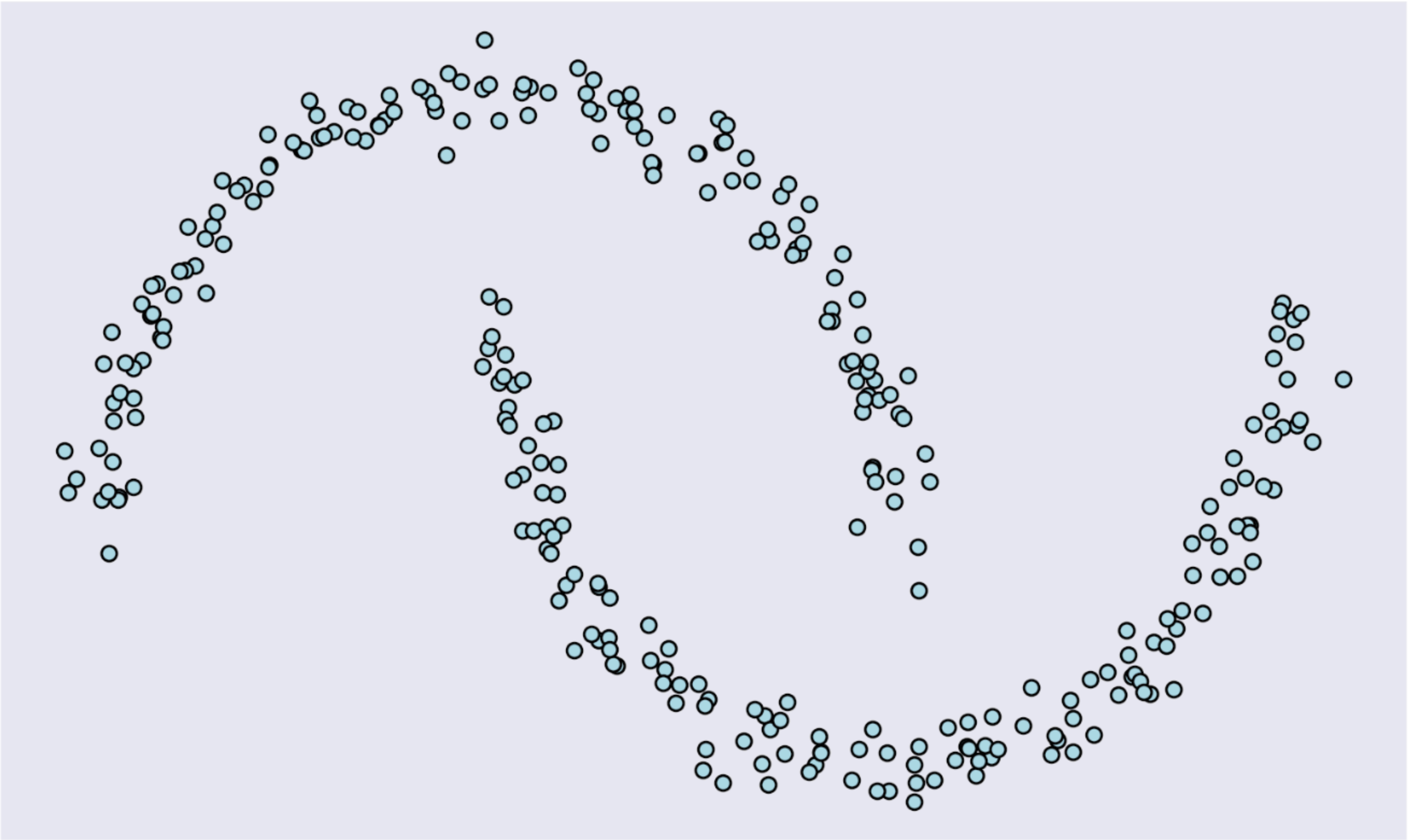
Modelos Generativos

Datos de Entrenamiento x_{train}

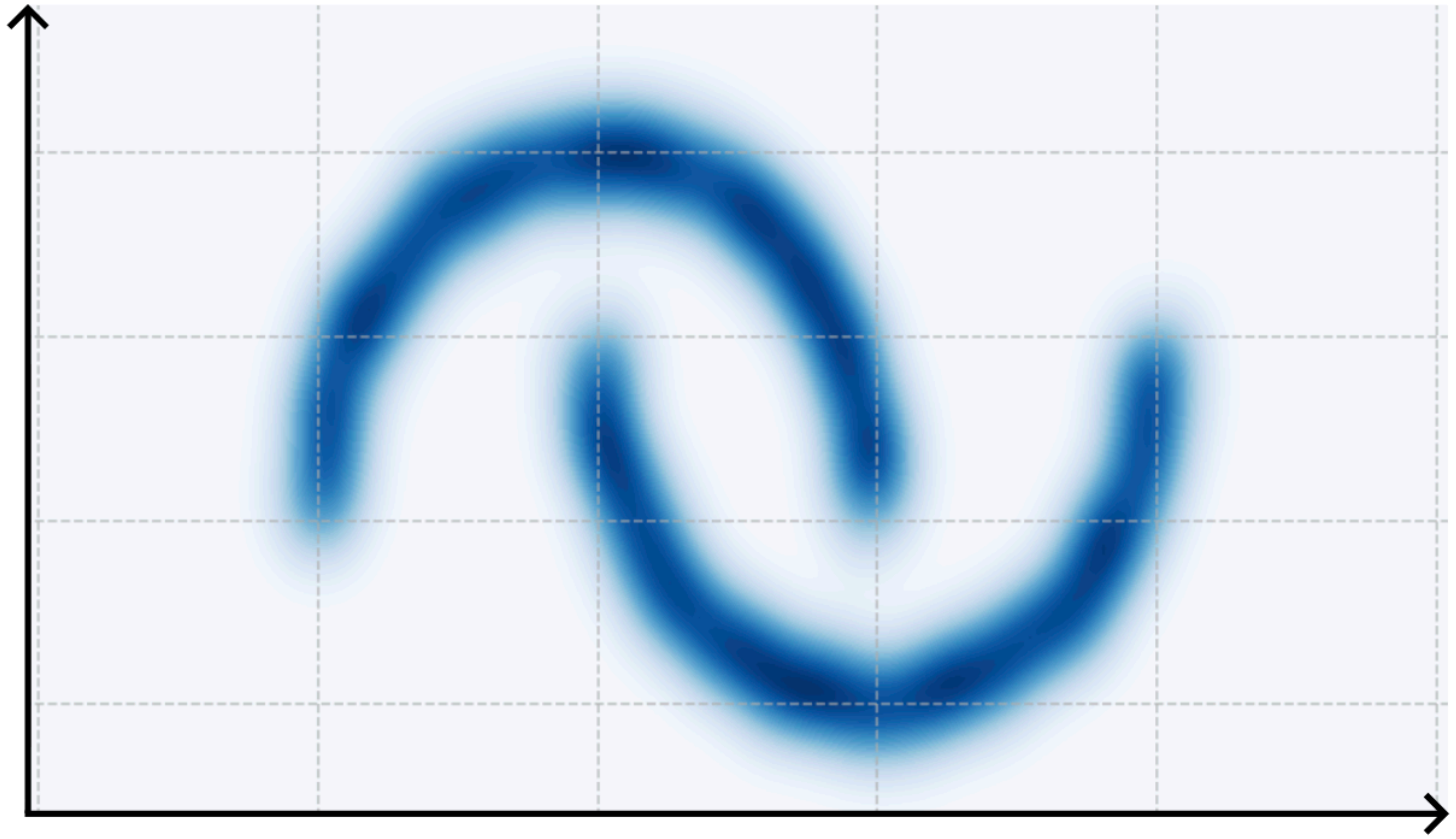


Modelos Generativos

Datos de Entrenamiento x_{train}

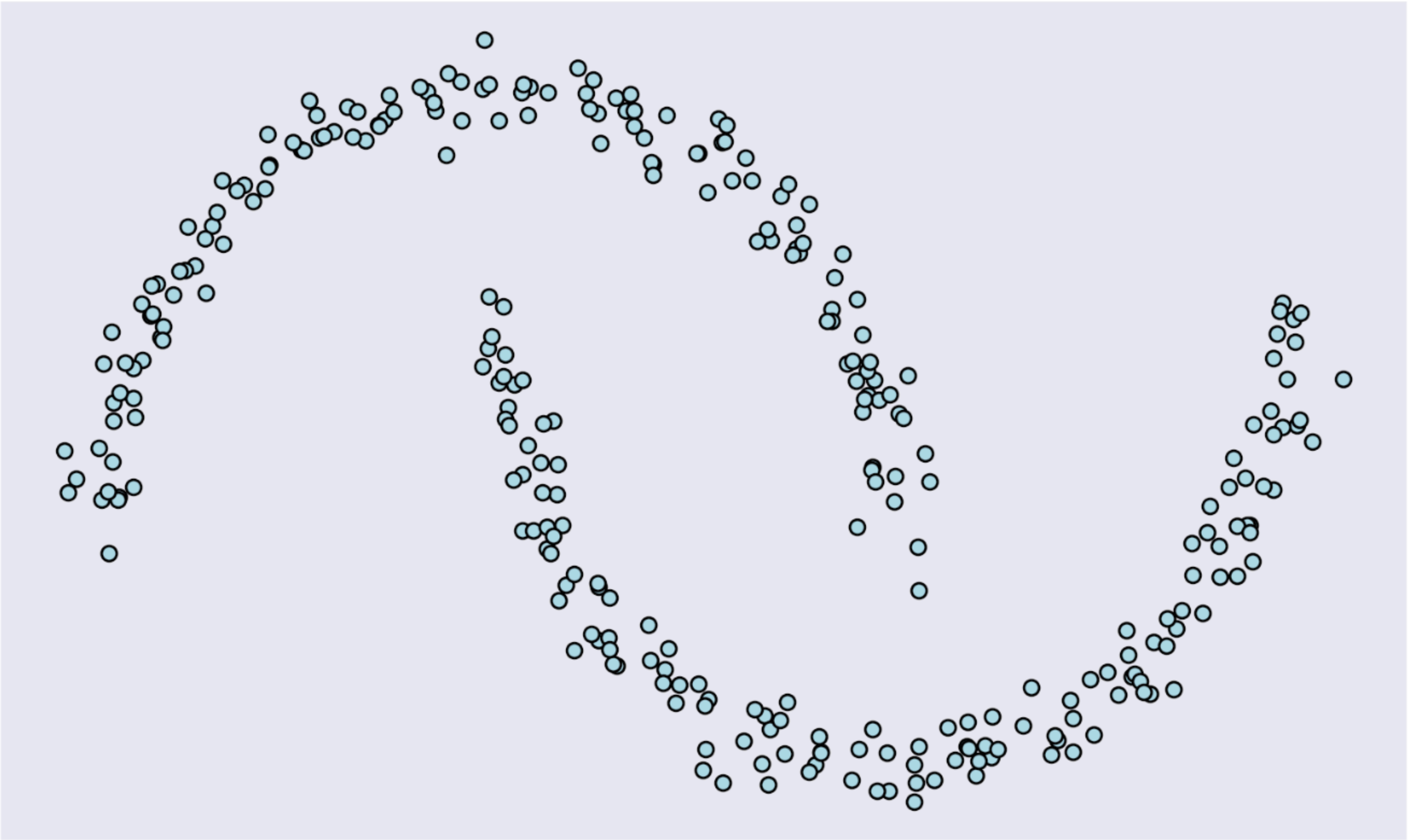


Modelo Paramétrico $p_\phi(x)$

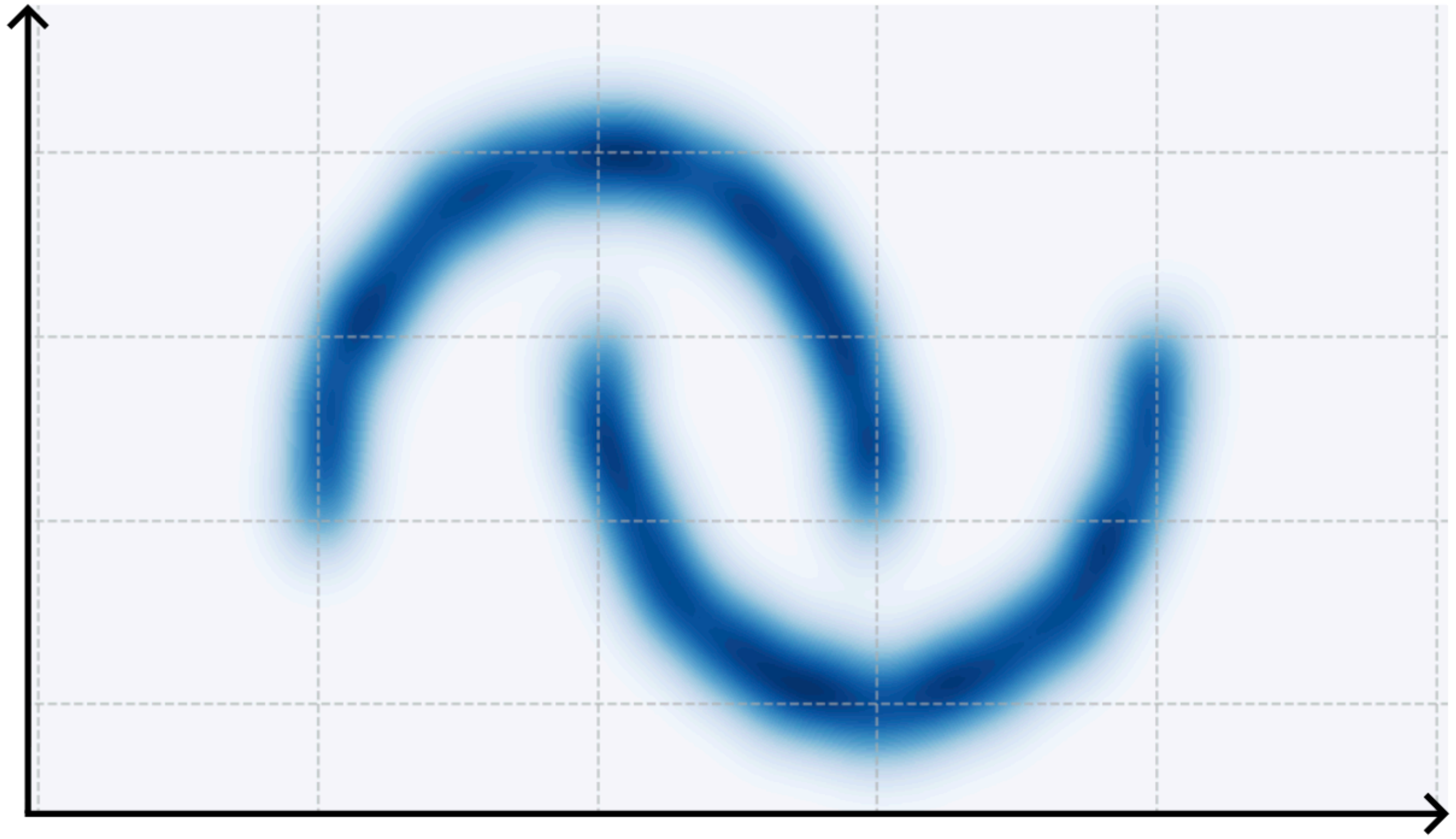


Modelos Generativos

Datos de Entrenamiento x_{train}



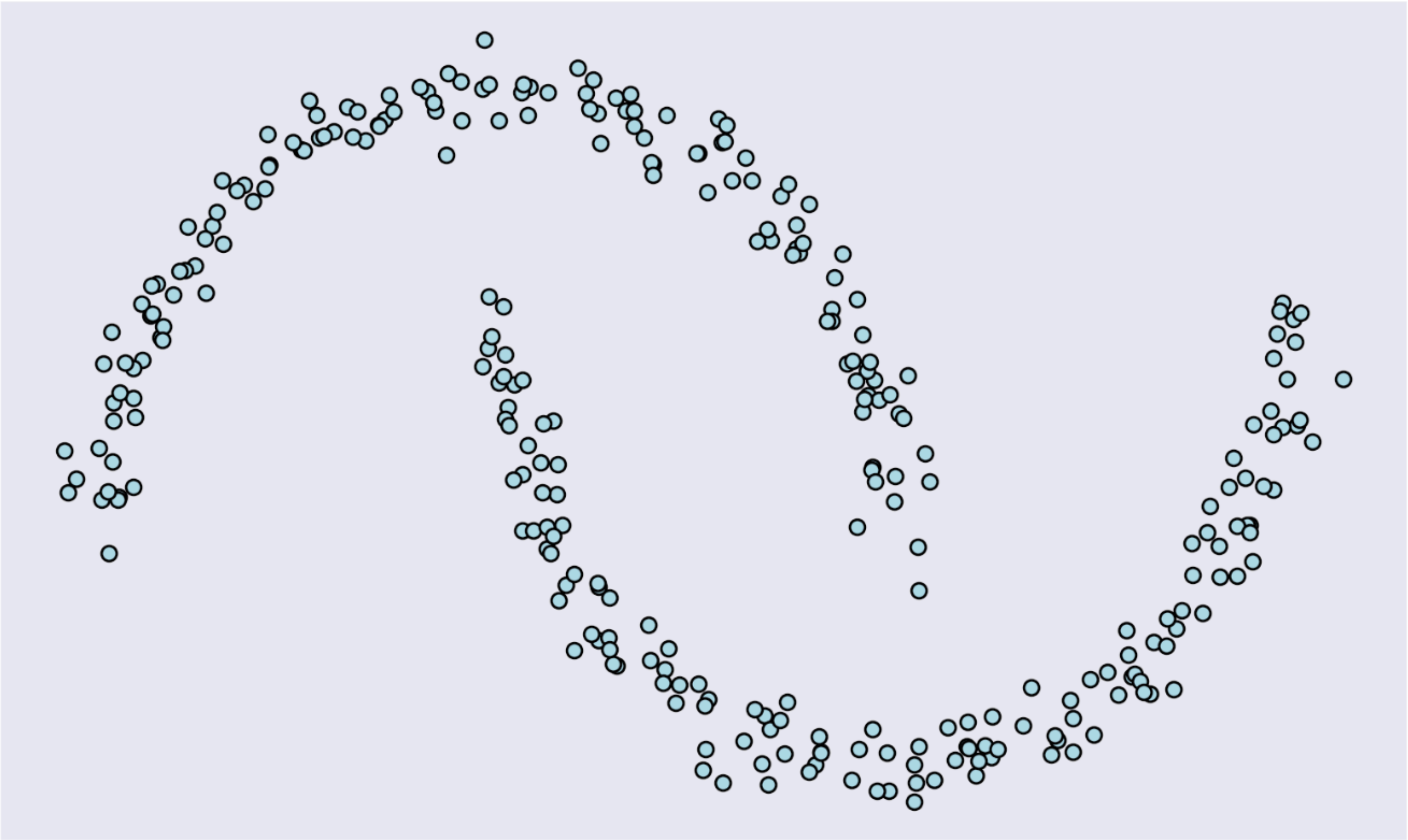
Modelo Paramétrico $p_\phi(x)$



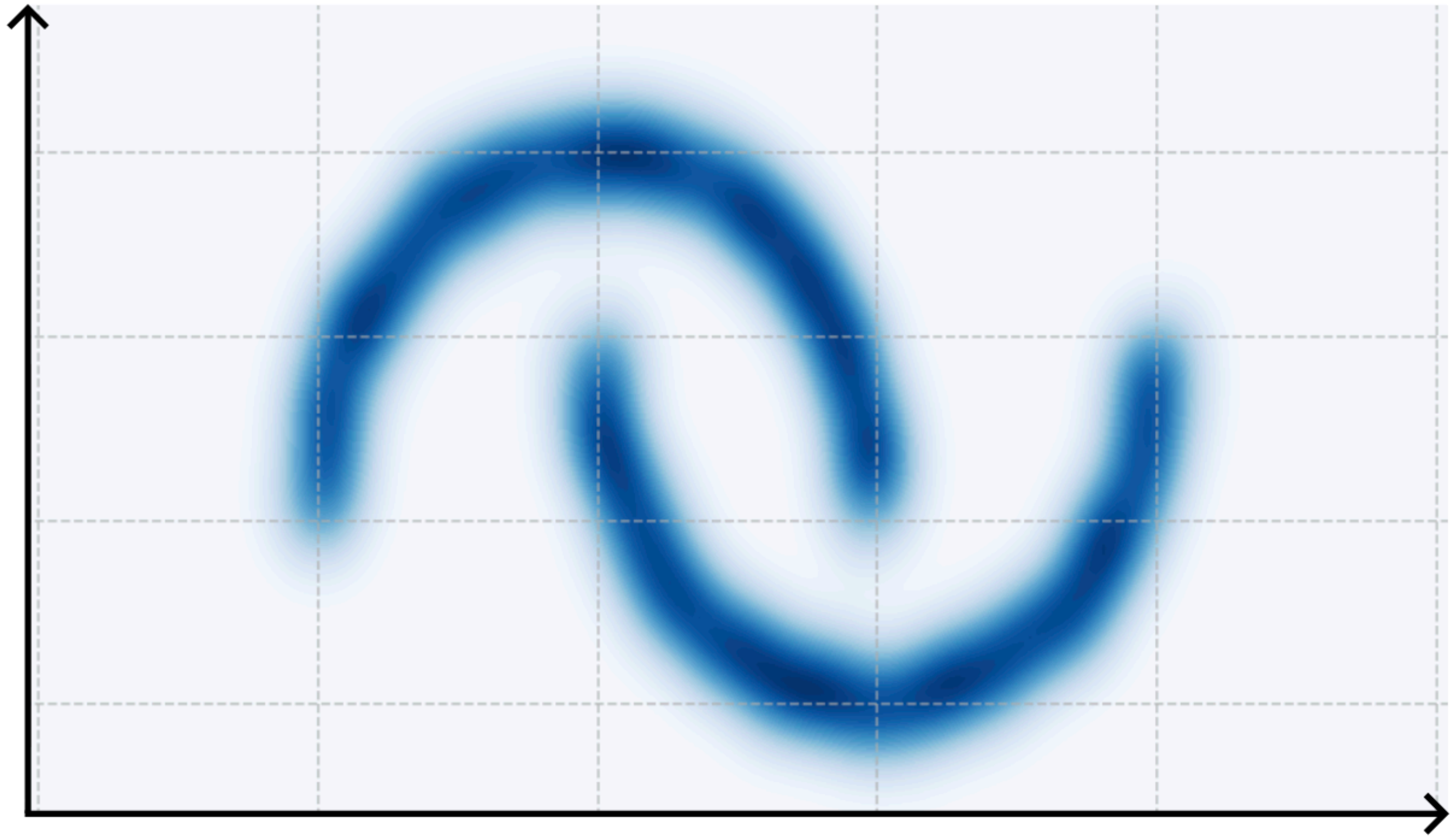
Maximizar la likelihood de los datos de entrenamiento

Modelos Generativos

Datos de Entrenamiento x_{train}



Modelo Paramétrico $p_\phi(x)$



Maximizar la likelihood de los datos de entrenamiento

$$\hat{\phi} = arg\,max[\log p_\phi(x_{train})]$$

Por qué Modelos Generativos

1. No Cheating

Por qué Modelos Generativos

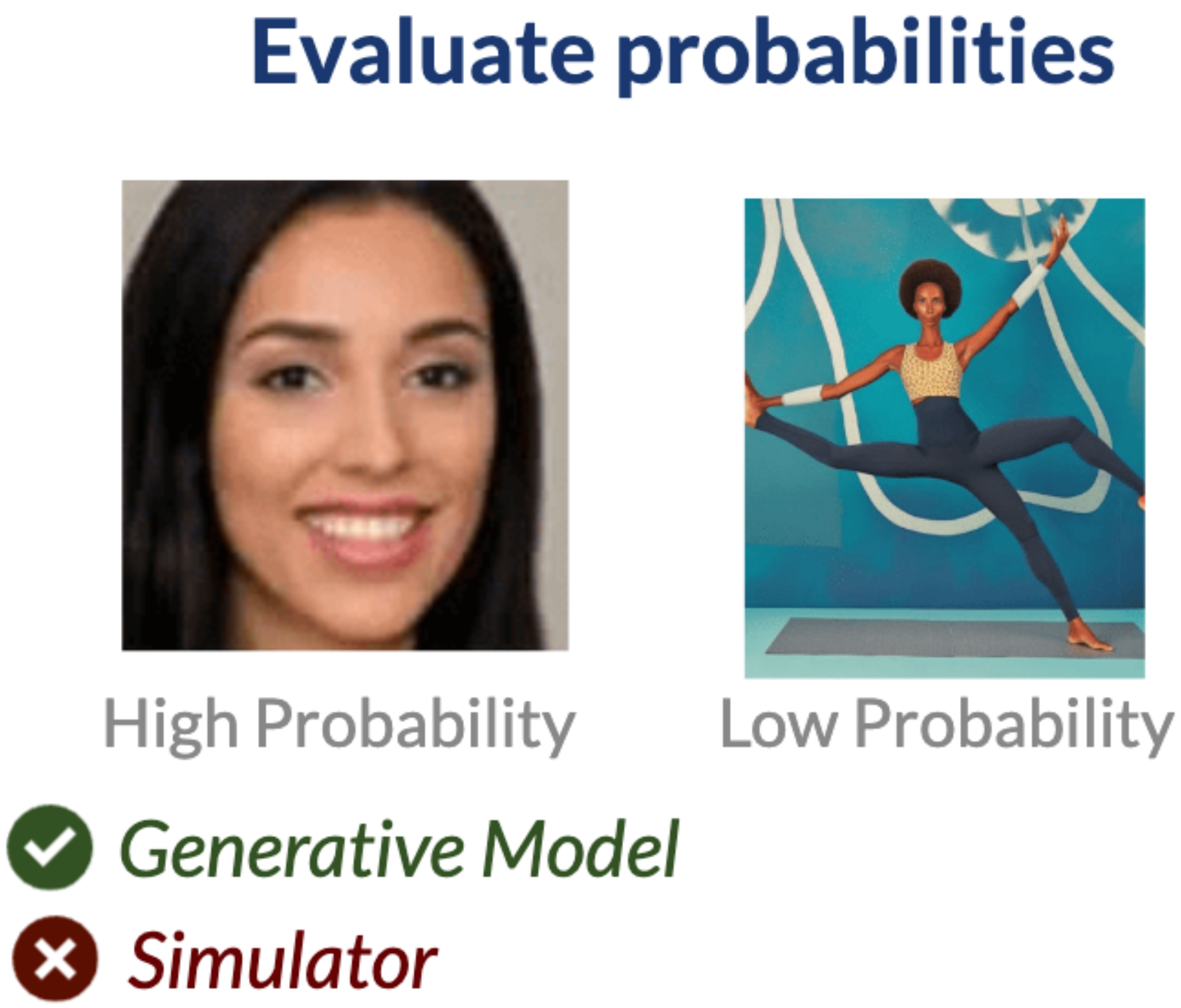
2. Simulador

Generate Novel Samples

- ✔ *Generative Model*
- ✔ *Simulator*

Por qué Modelos Generativos

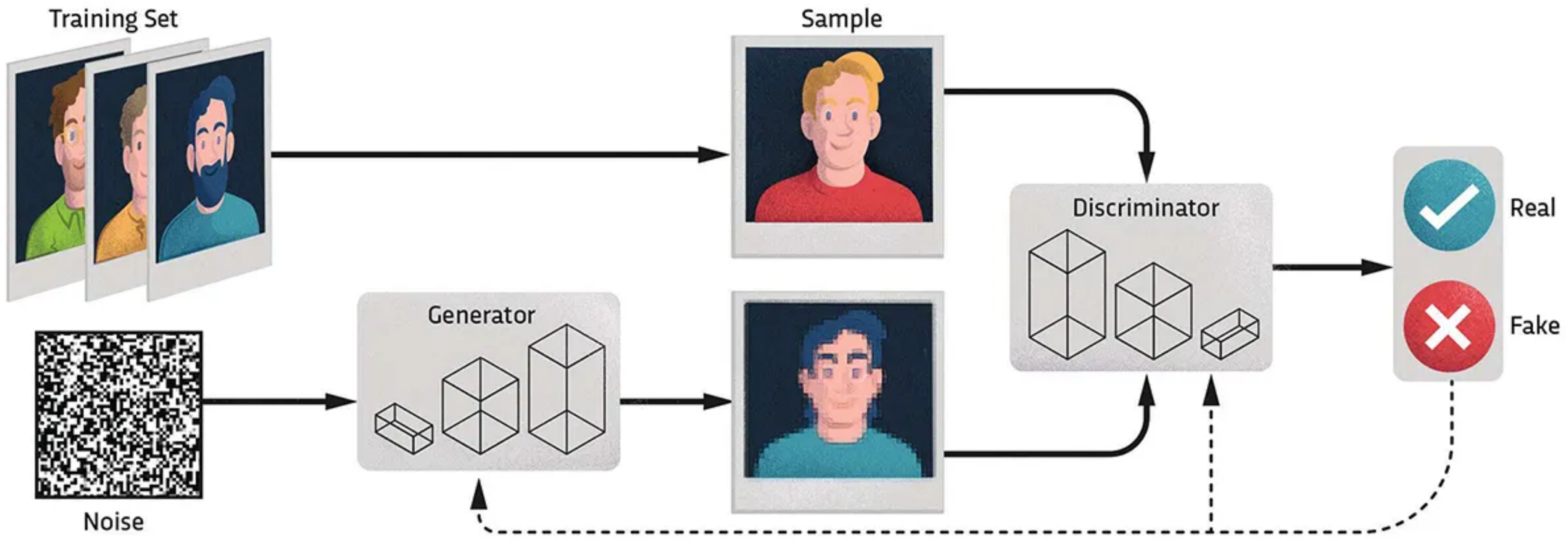
3. Evaluar de Probabilidades



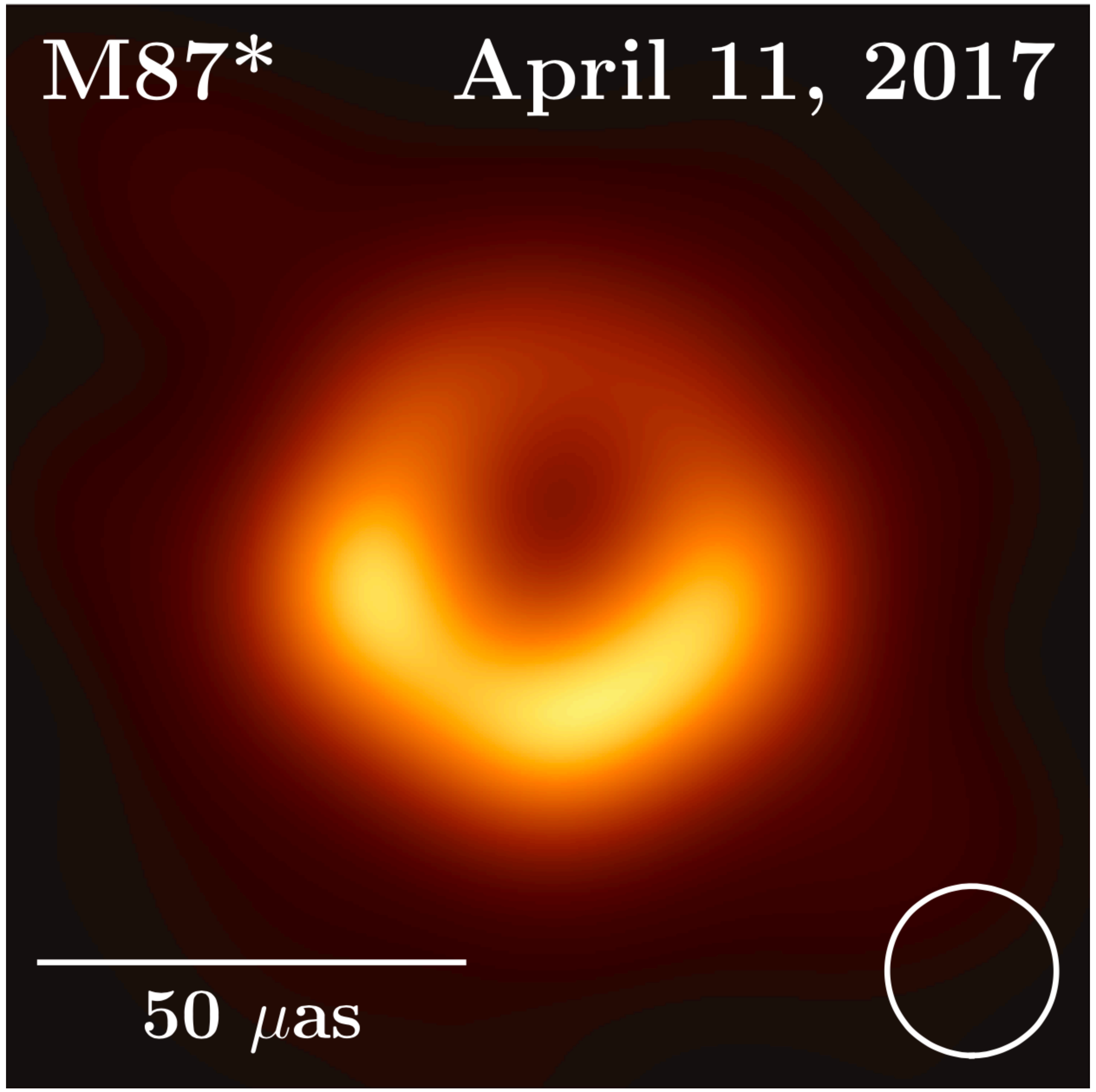
Modelos Generativos:

- Redes Adversarias Generativas (GANs)
- Autocodificadores Variaciones (VAE)
- Flujos Normalizantes (NFs)
- Modelos de Difusión
- Modelos de Flow Matching

Redes Generativas Adversarias

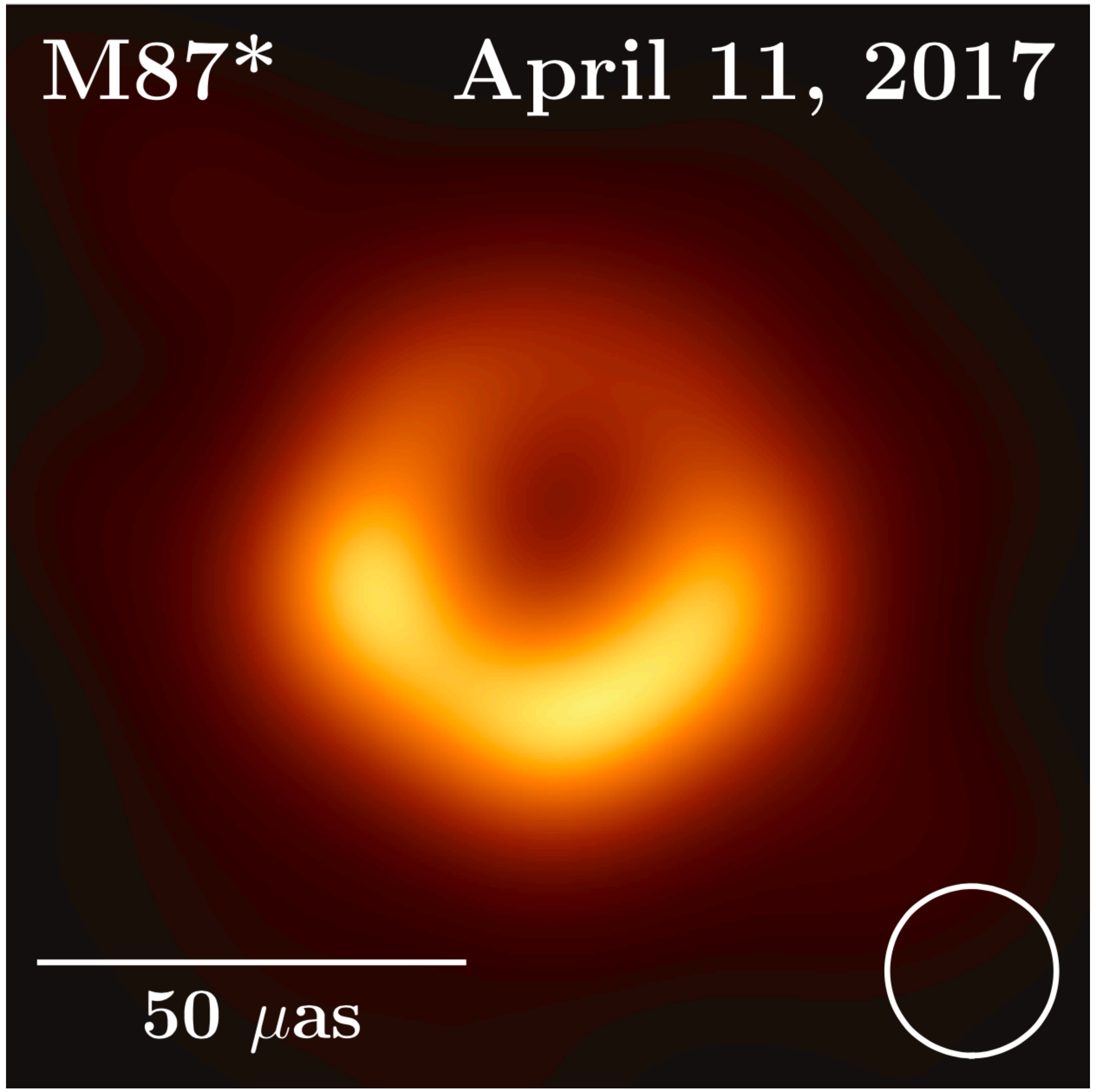


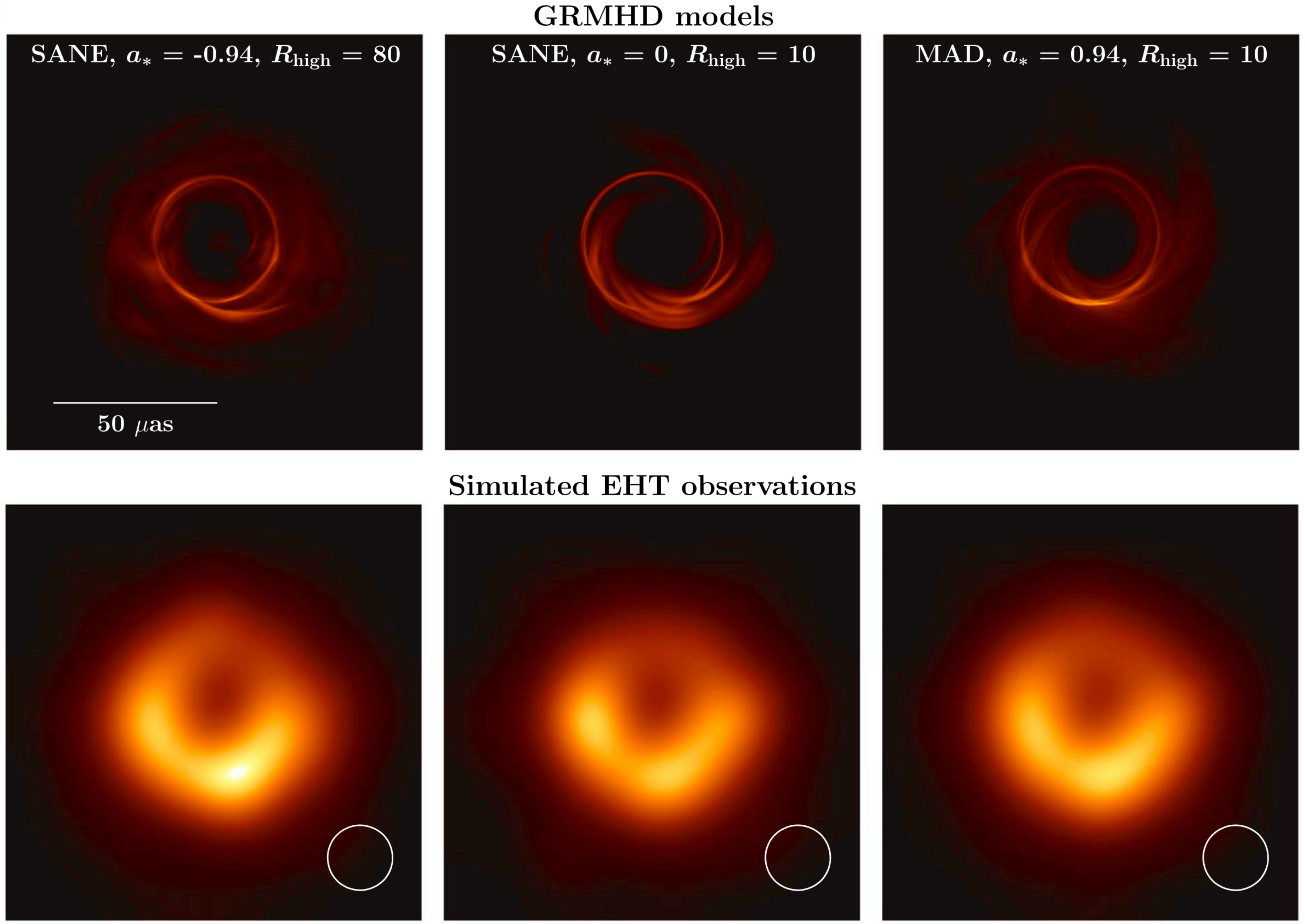
Redes Generativas Adversarias



Event Horizon Telescope Collaboration

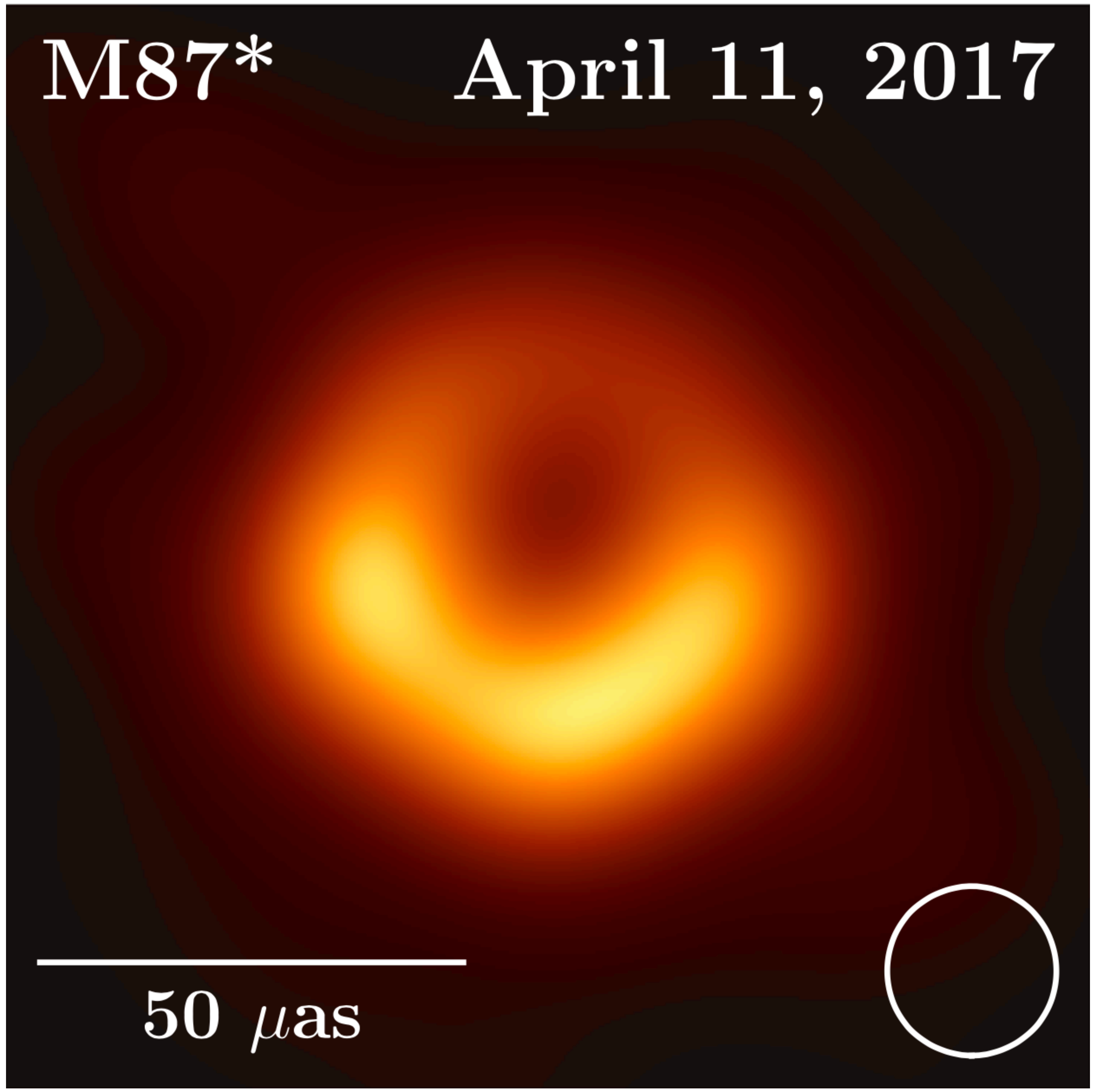
Redes Generativas Adversarias

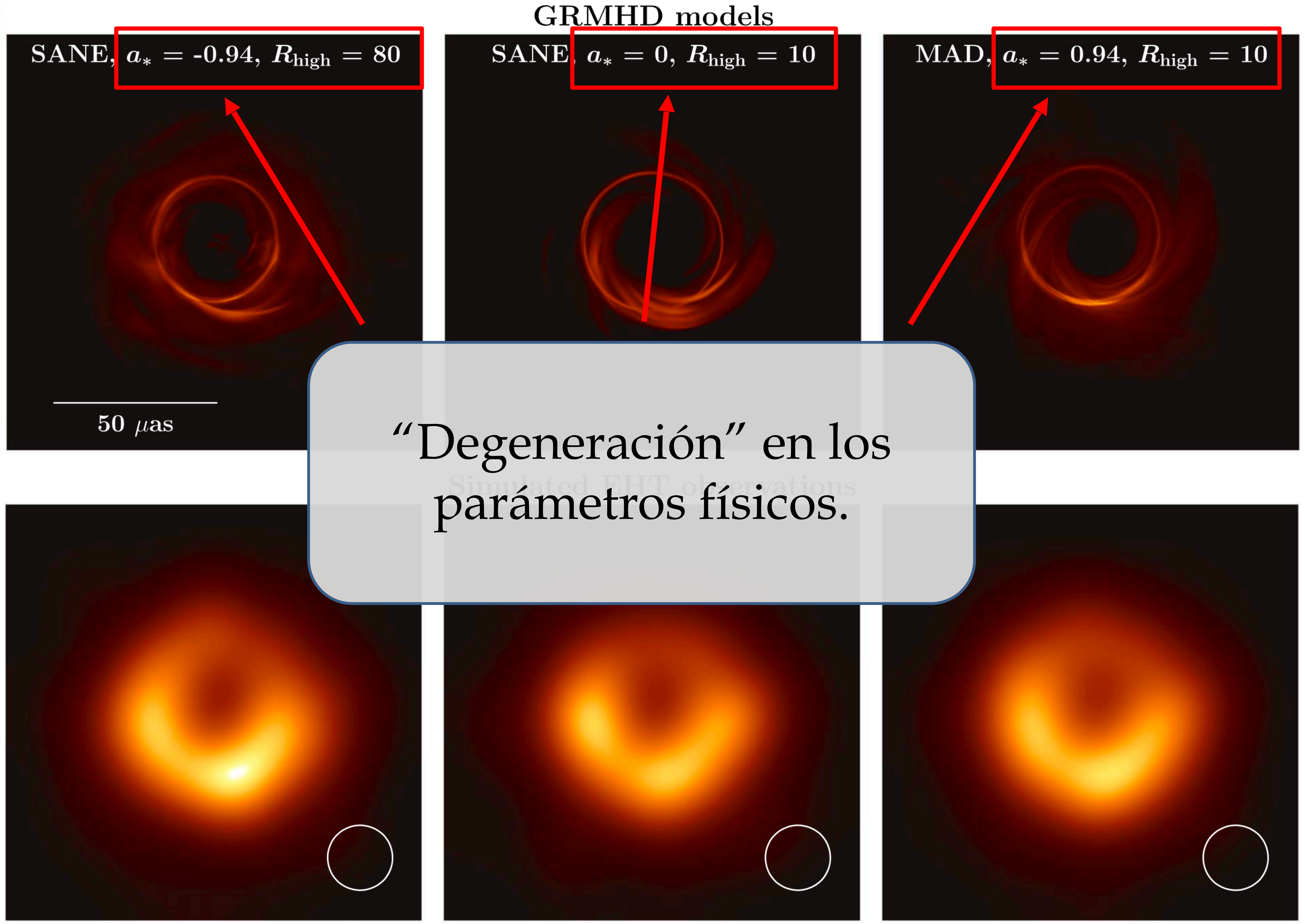




Event Horizon Telescope Collaboration

Redes Generativas Adversarias

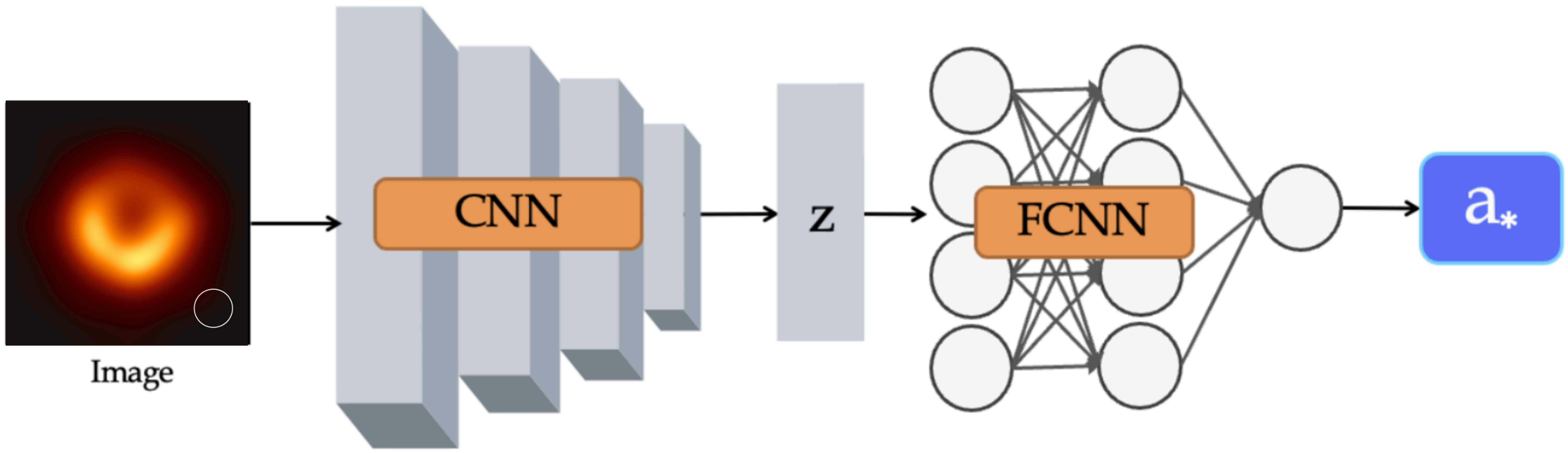




"Degeneración" en los parámetros físicos.

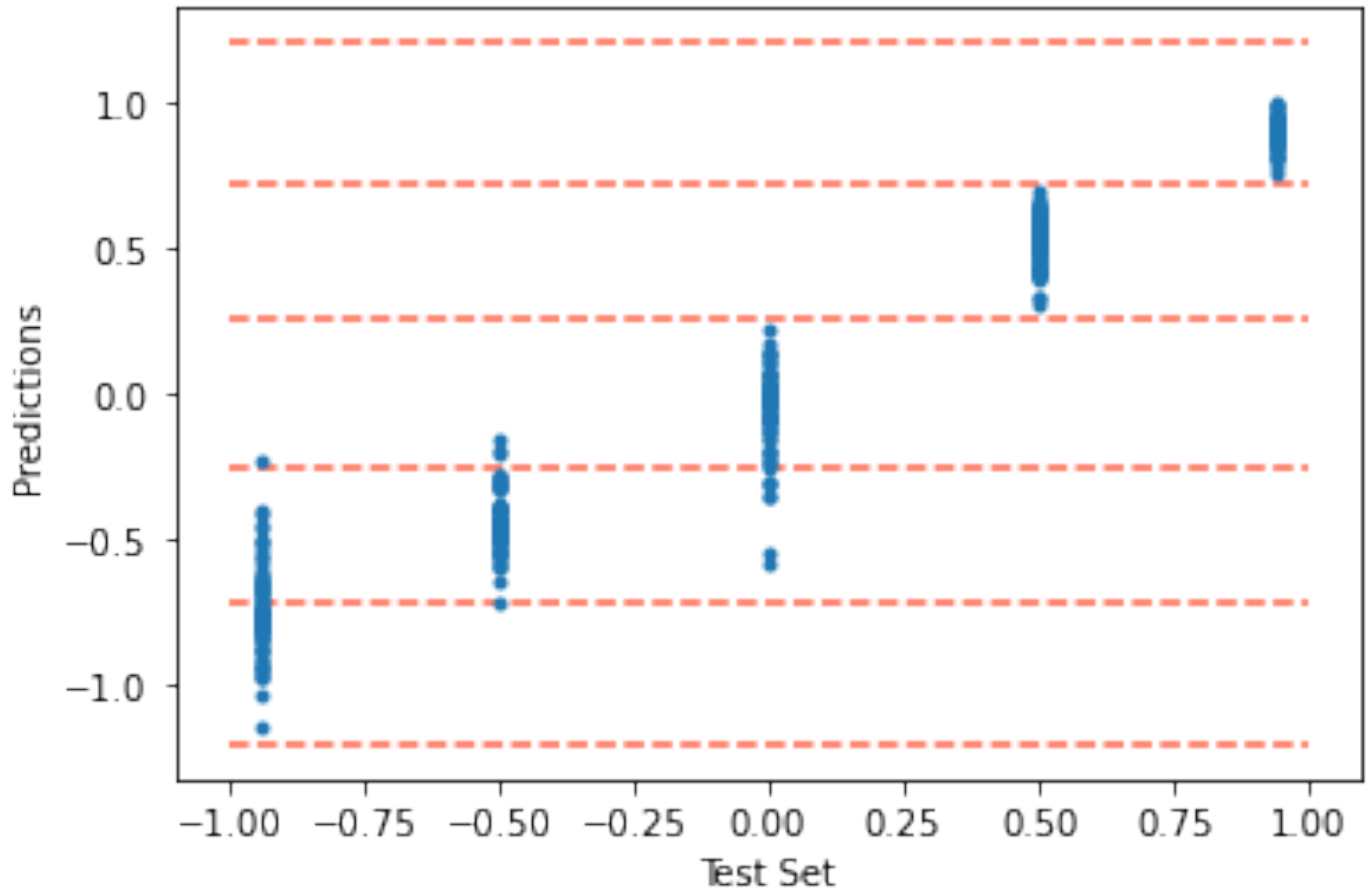
Event Horizon Telescope Collaboration

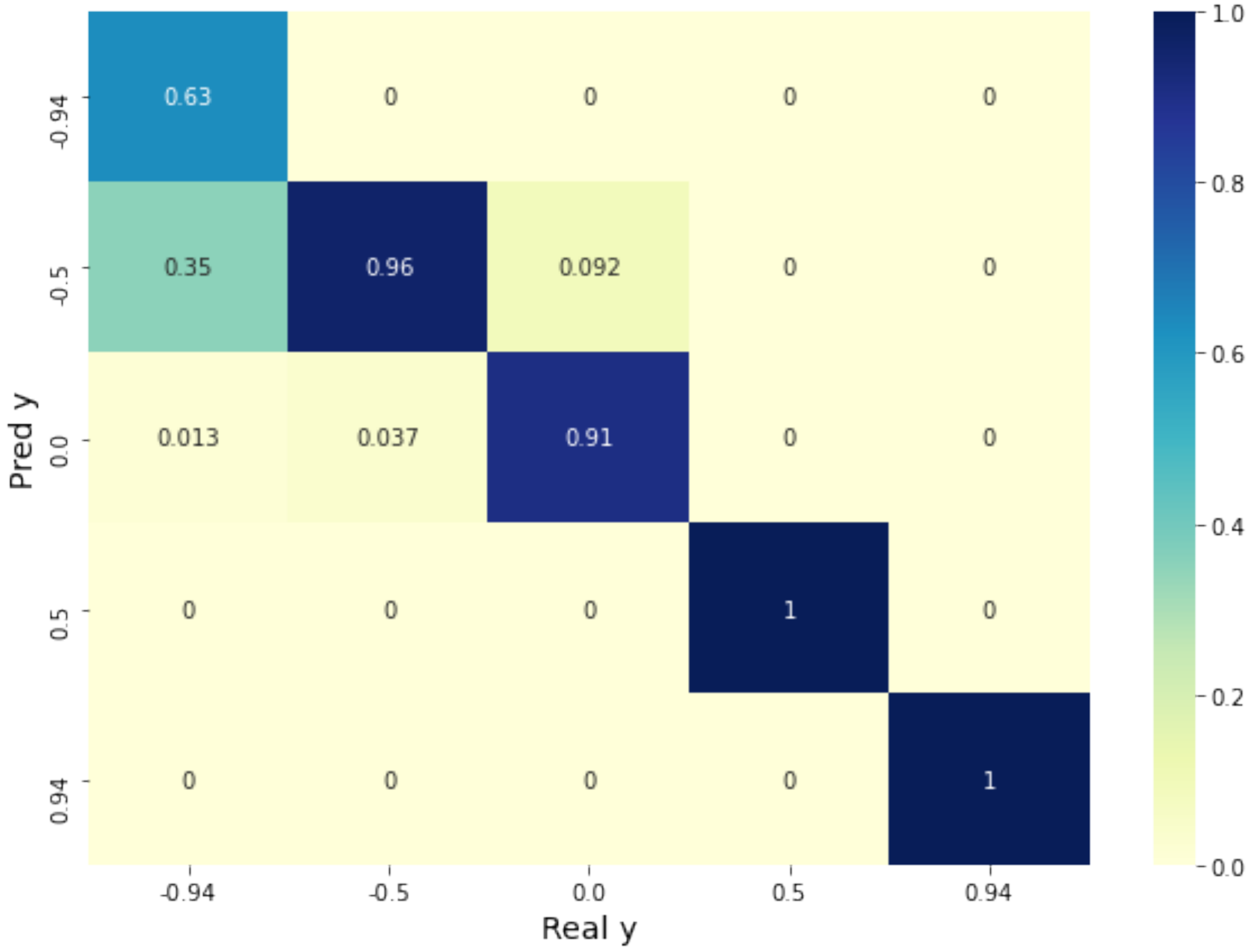
Redes Generativas Adversarias



Tsui et al. 2024

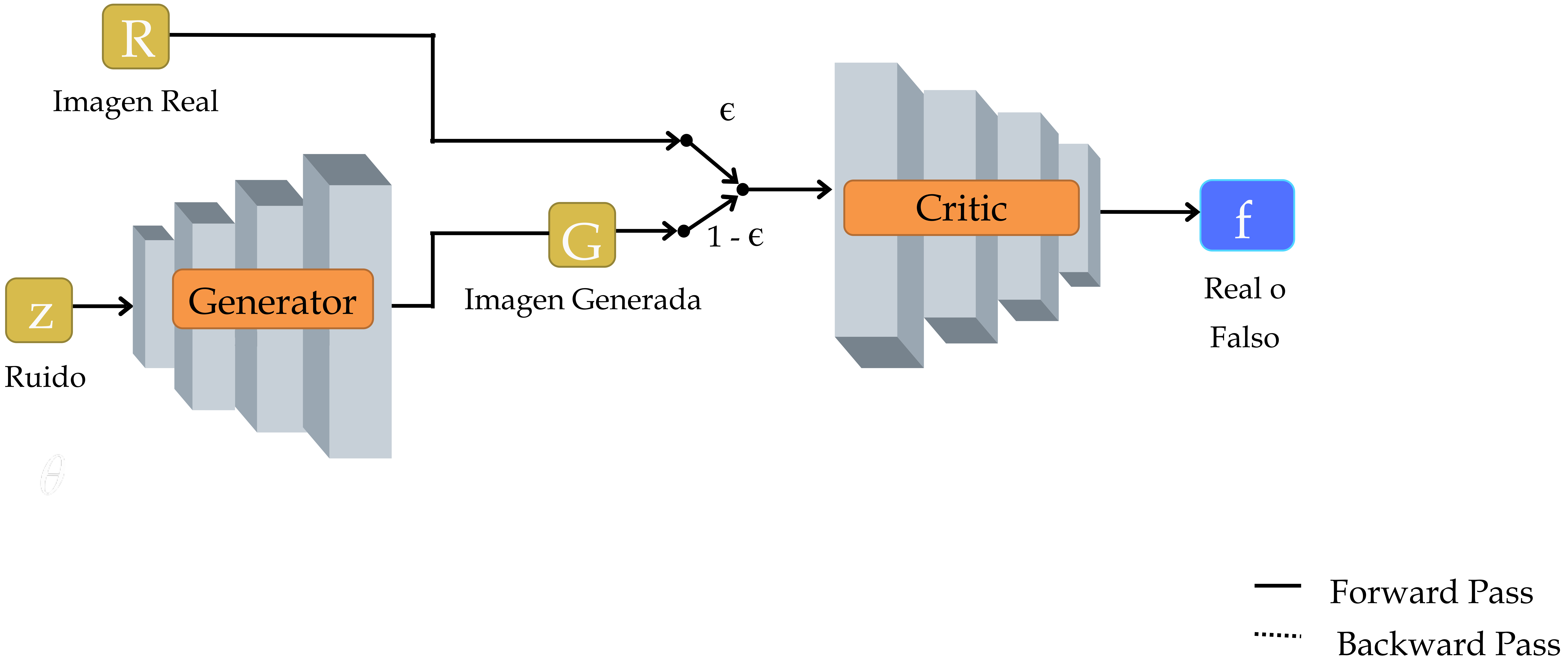
Redes Generativas Adversarias



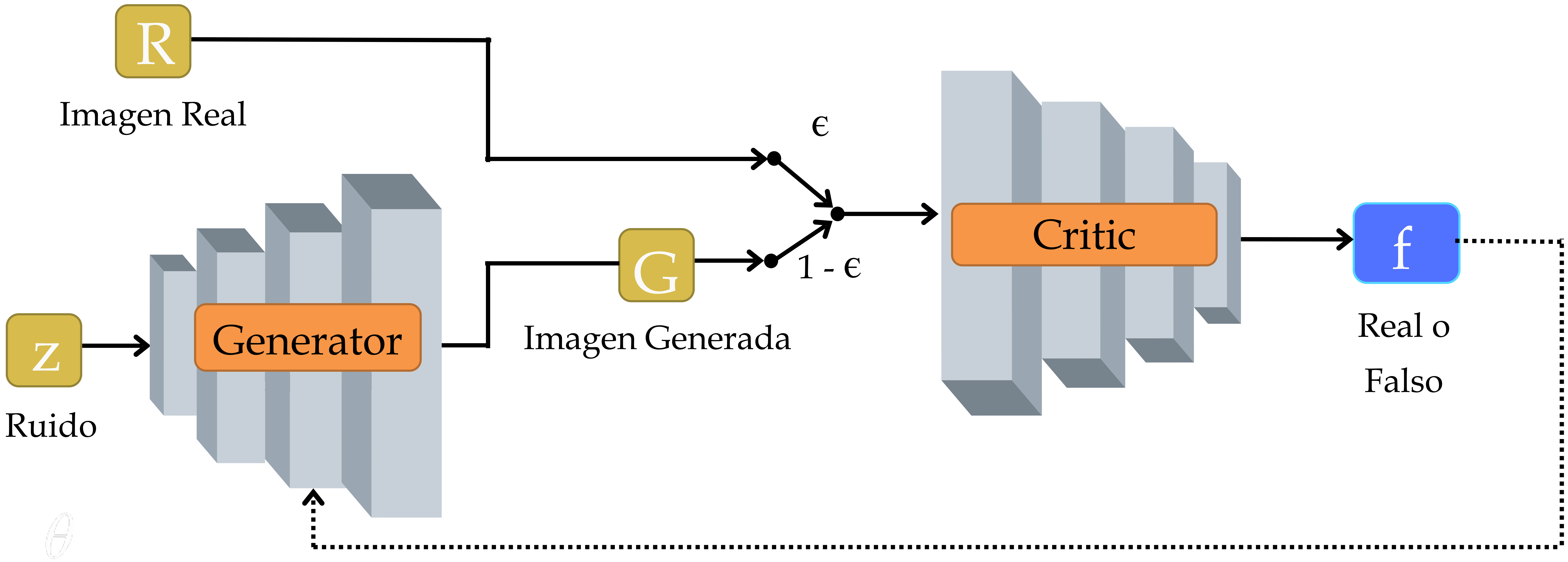


Tsui et al. 2024

Redes Generativas Adversarias



Redes Generativas Adversarias

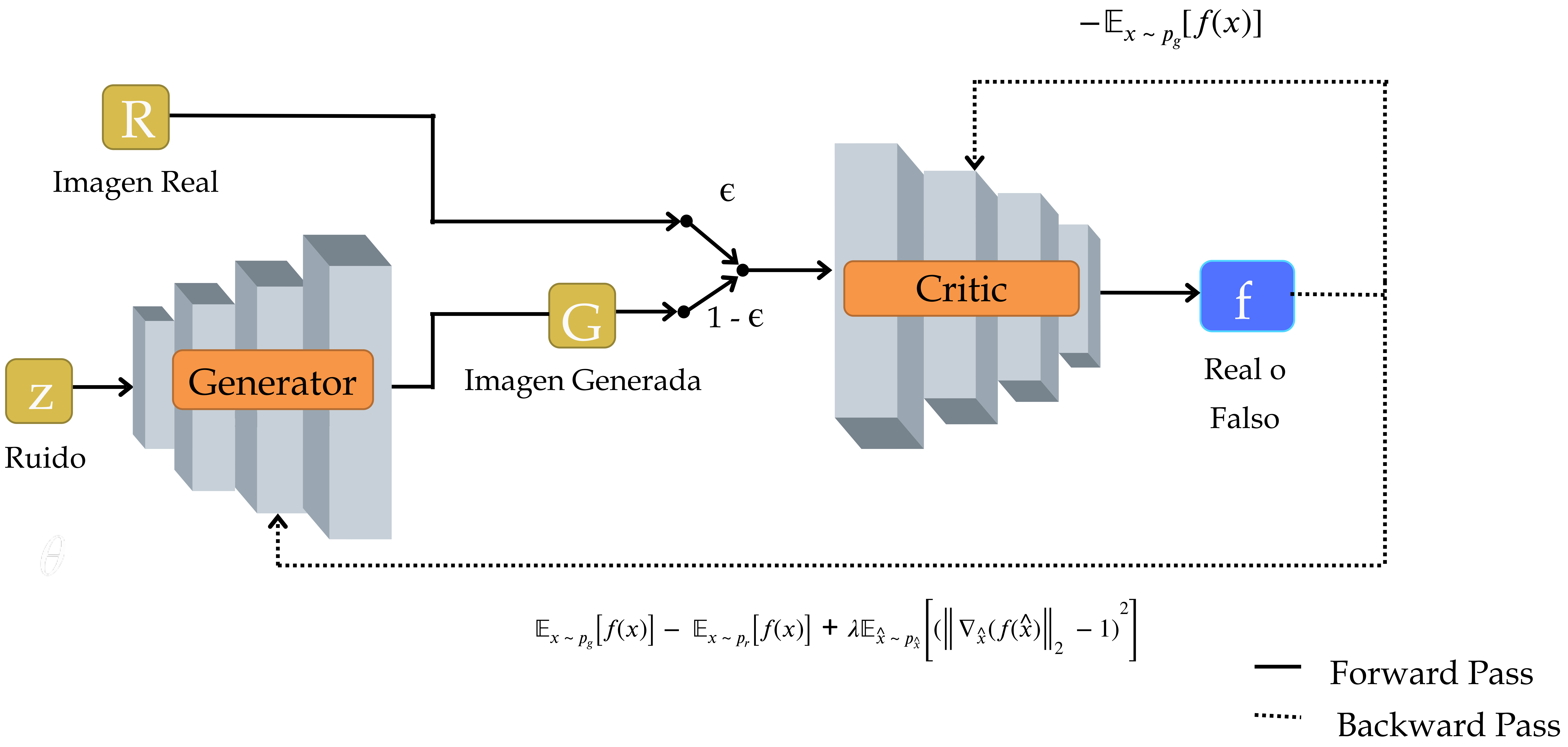


$$\mathbb{E}_{x \sim p_g}[f(x)] - \mathbb{E}_{x \sim p_r}[f(x)] + \lambda \mathbb{E}_{\hat{x} \sim p_{\hat{x}}}\left[(\|\nabla_{\hat{x}}(f(\hat{x})\|_2 - 1)^2\right]$$

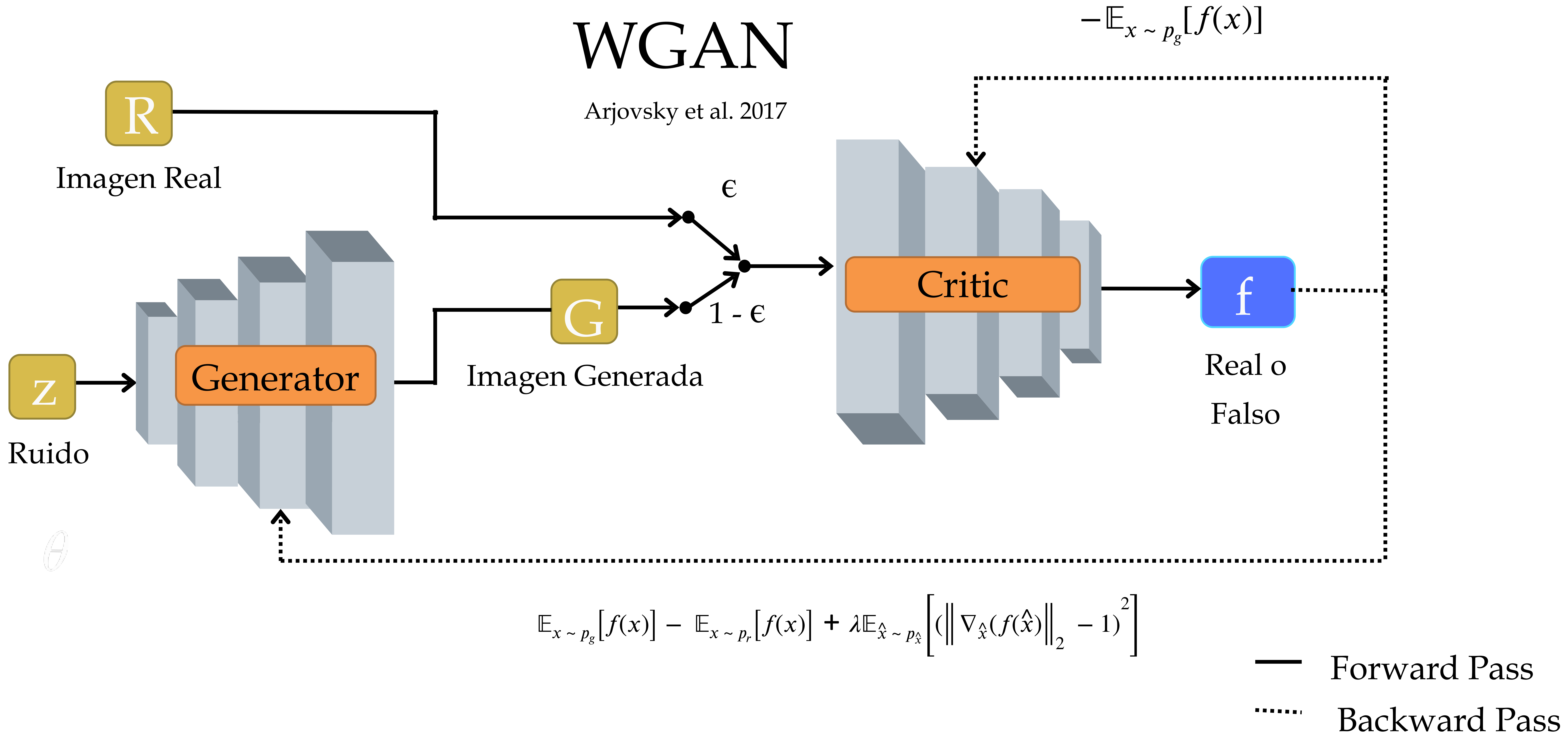
Forward Pass

Backward Pass

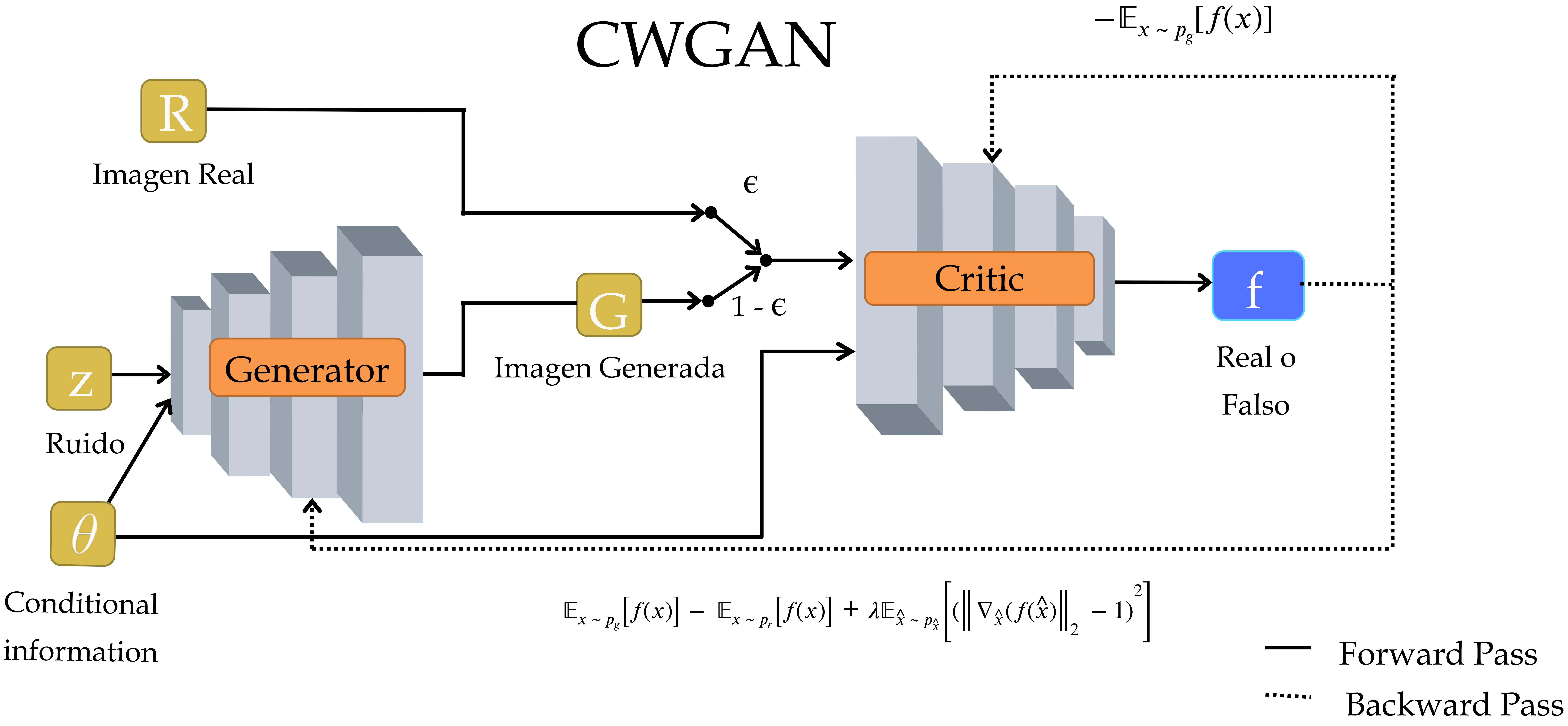
Redes Generativas Adversarias



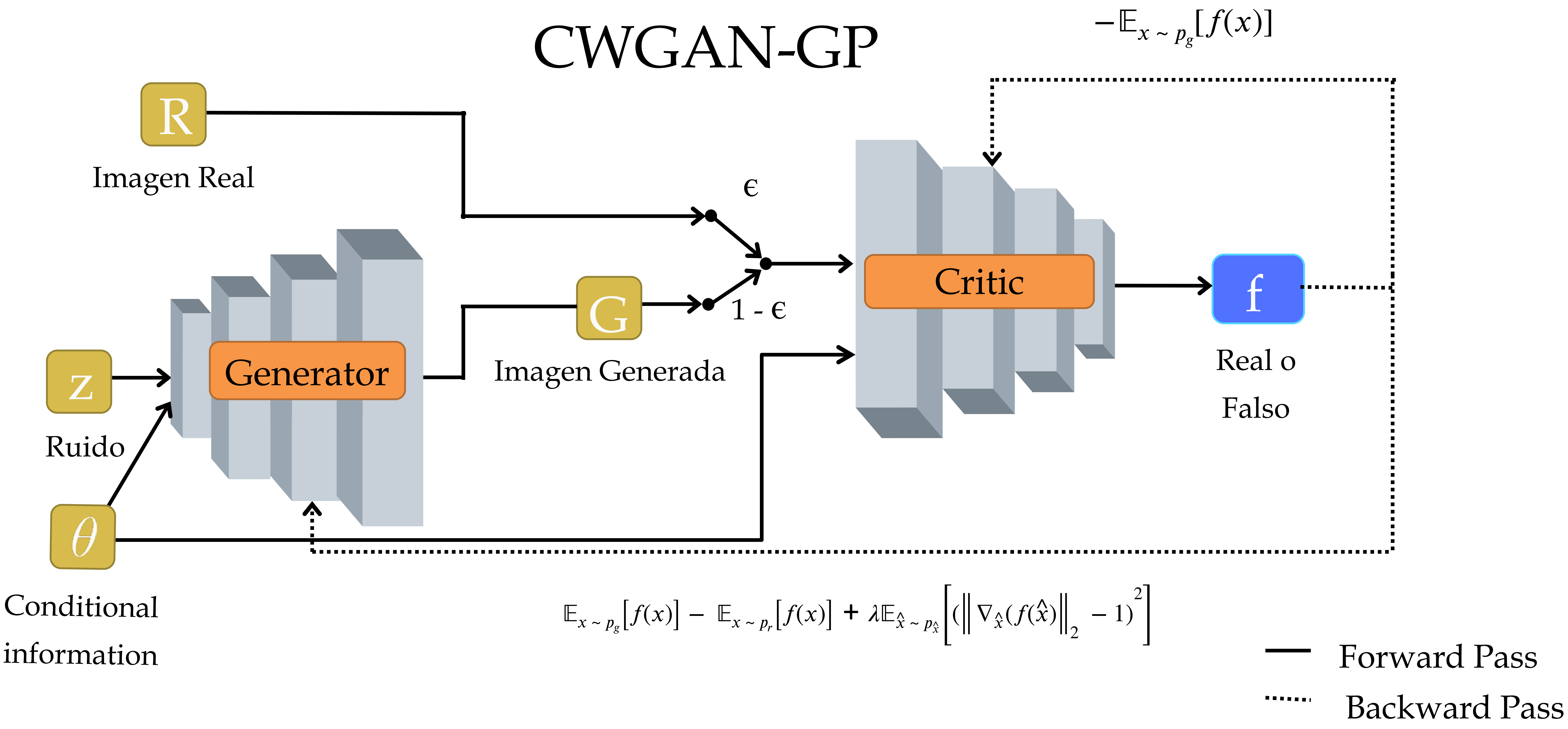
W Redes Generativas Adversarias



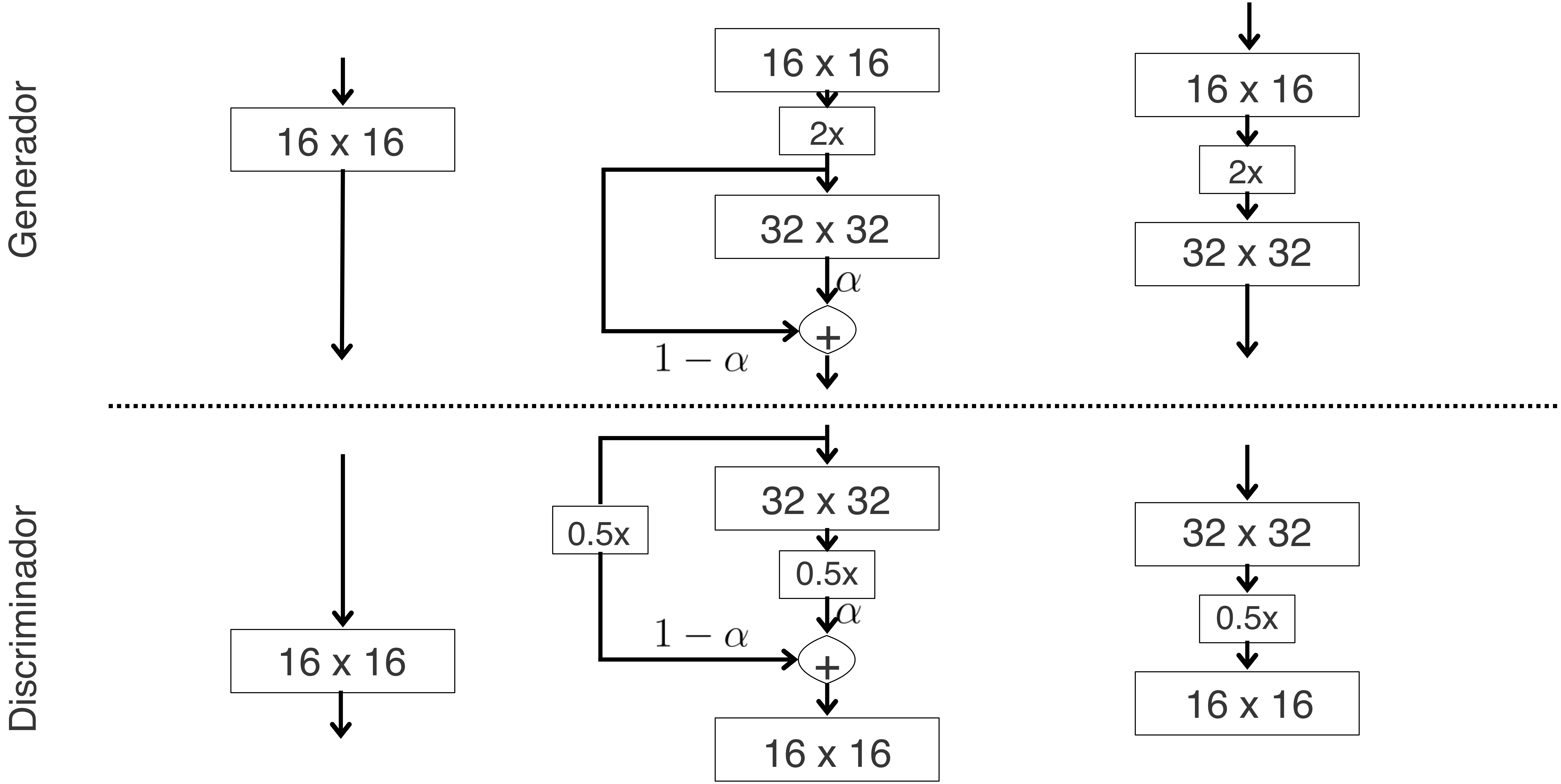
CW Redes Generativas Adversarias



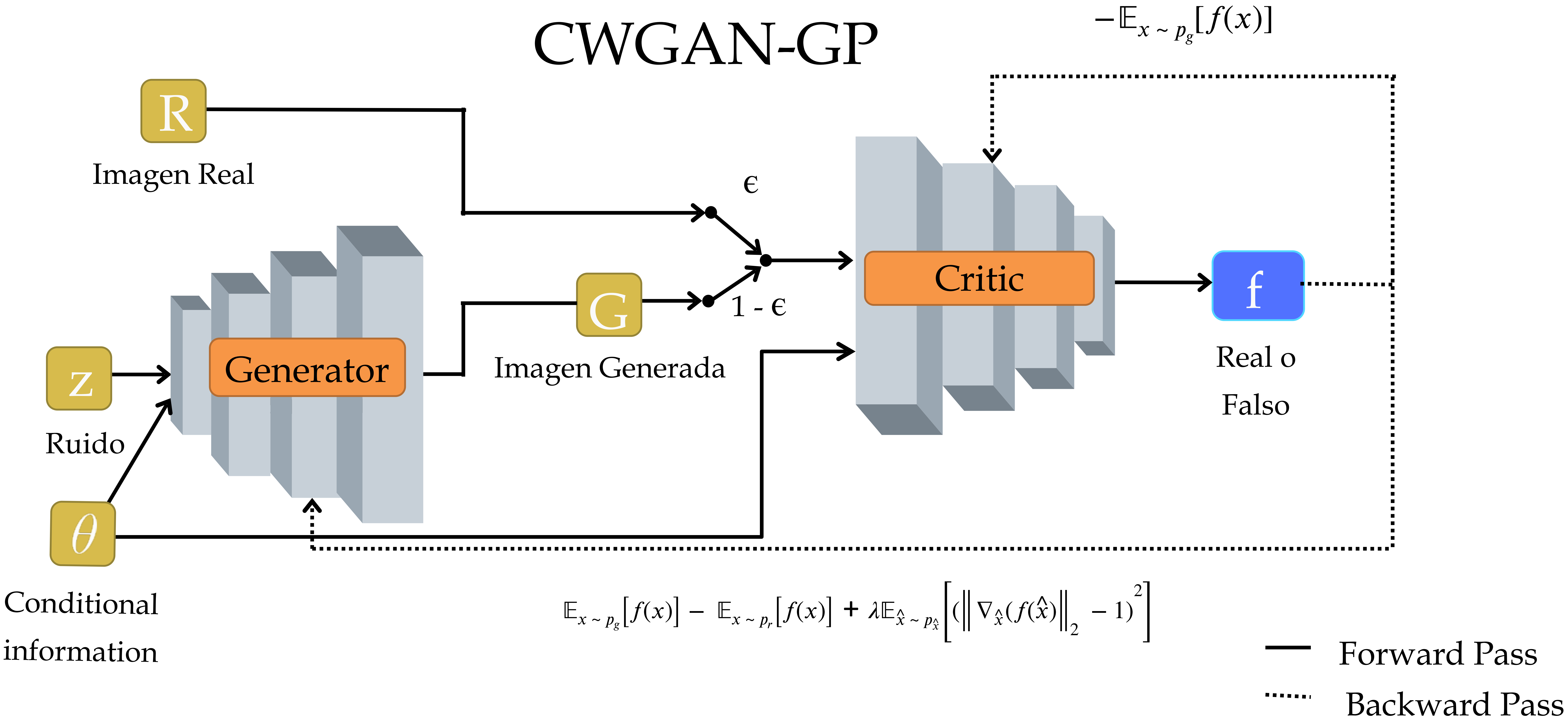
C Redes Generativas Adversarias



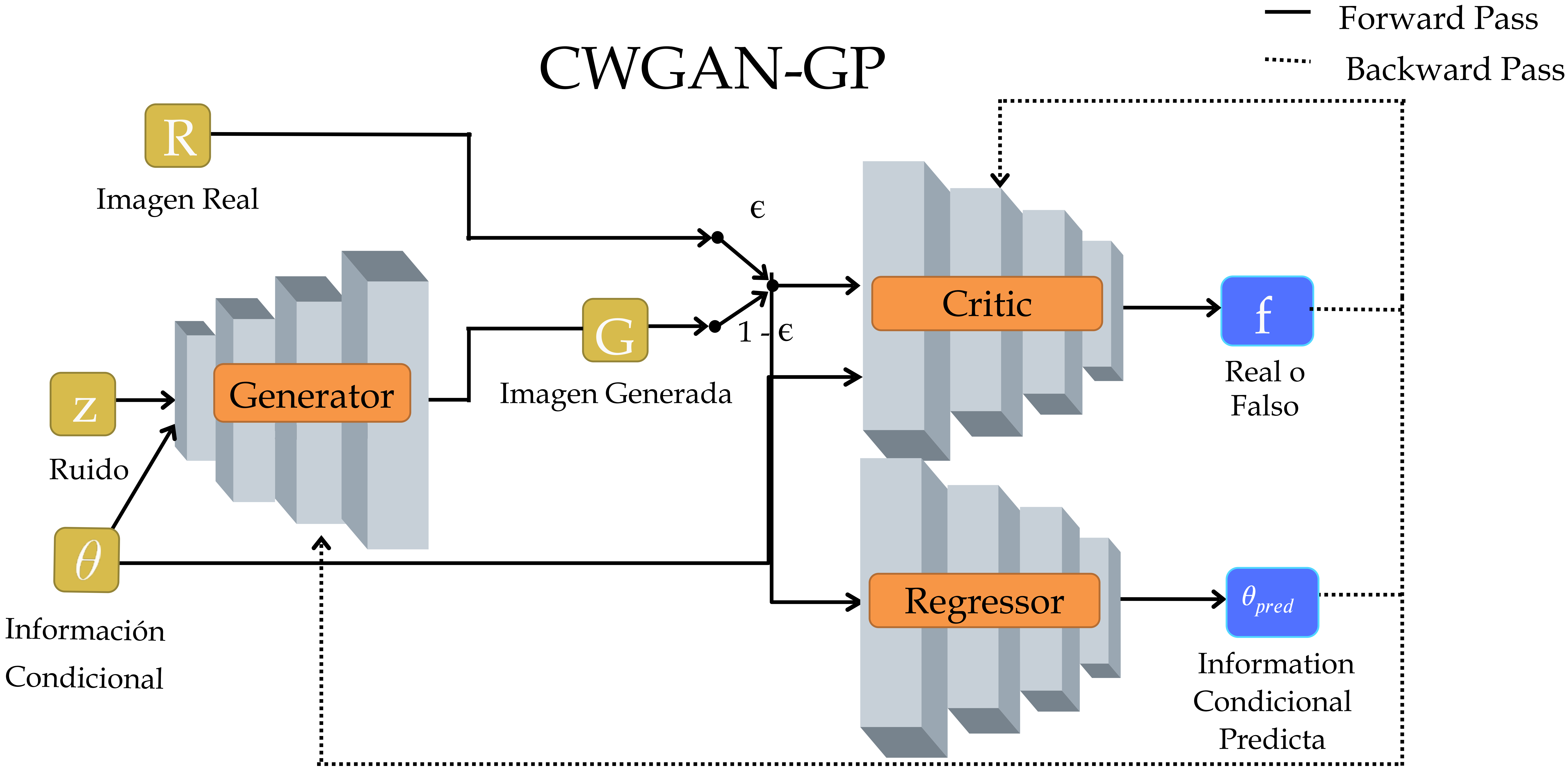
Redes Generativas Adversarias Progressivas



C Redes Generativas Adversarias

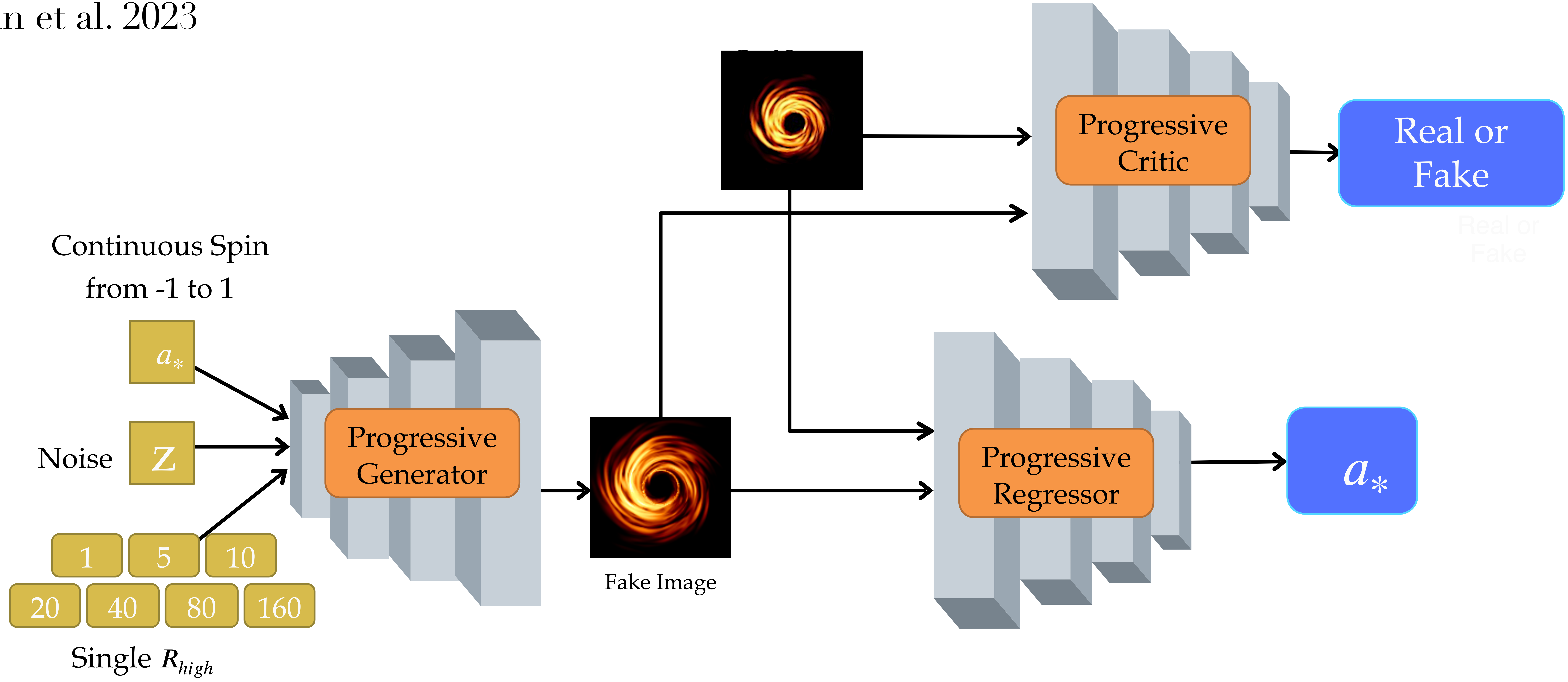


CW Redes Generativas Adversarias Versátiles



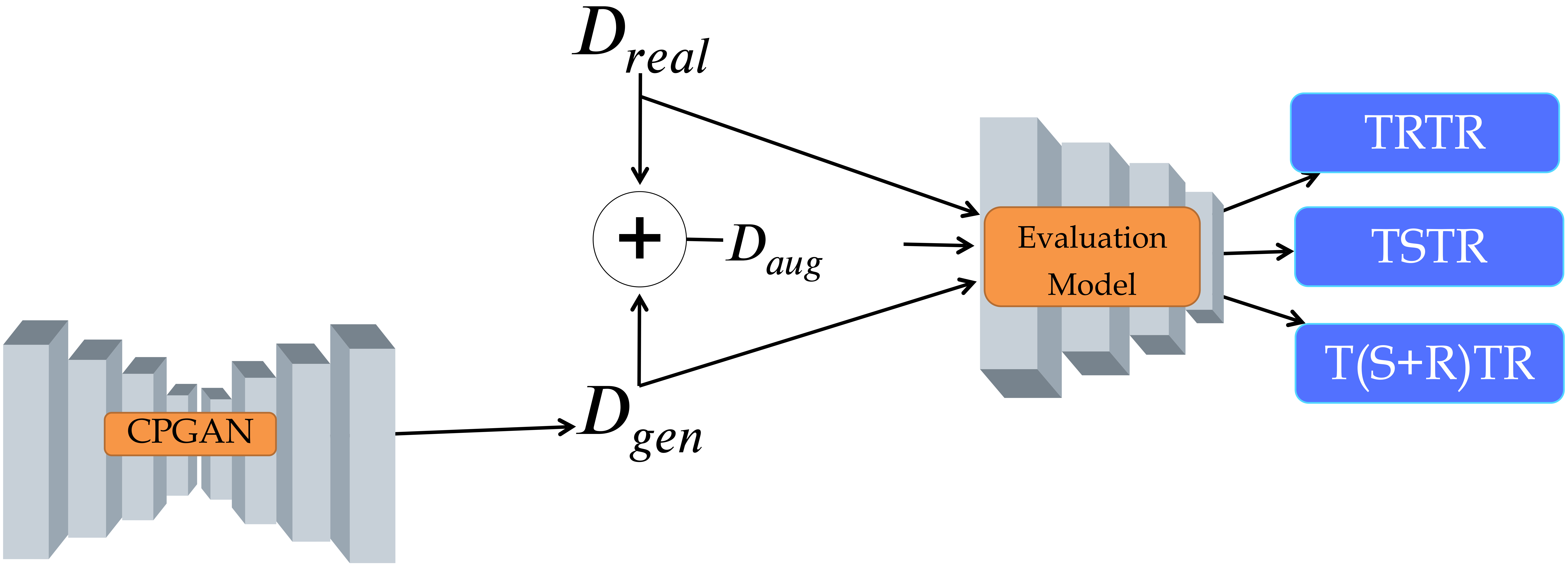
EHT VCWGAN

Mohan et al. 2023



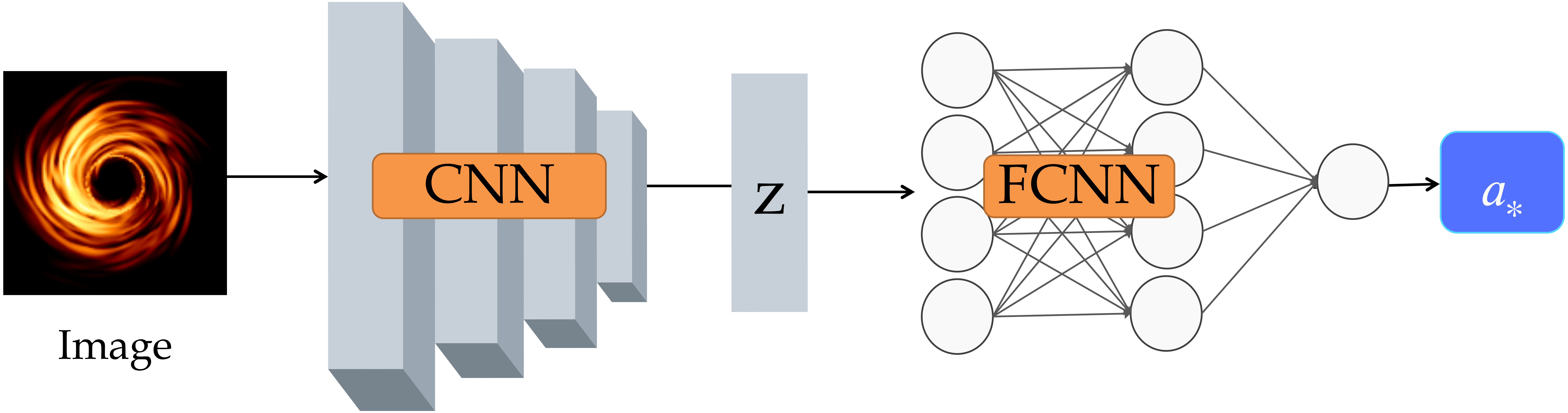
EHT VCWGAN: Evaluation

Mohan et al. 2023



EHT VCWGAN: Evaluation

Tsui et al. 2024



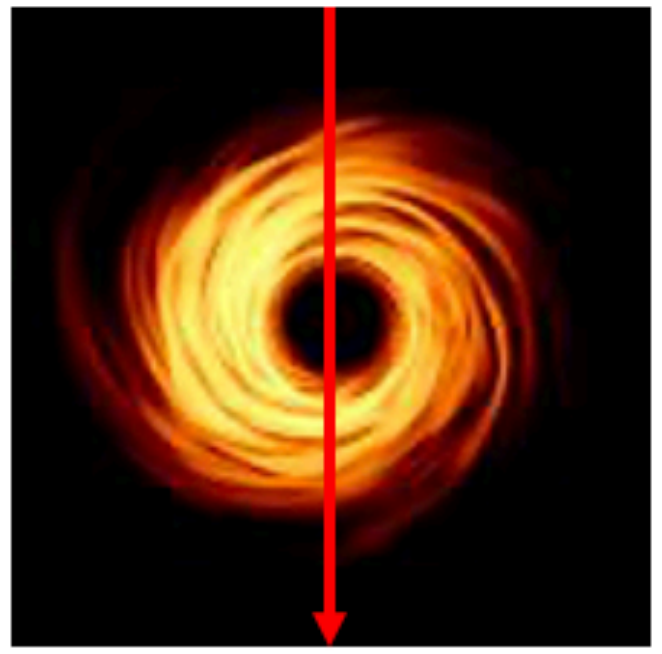
EHT VCWGAN: Evaluation

	TRTR	TSTR	(TR+TS)TR
1	0.975	0.978	0.985
5	0.926	0.909	0.931
10	0.917	0.935	0.943
20	0.988	0.879	0.915
40	0.976	0.903	0.981
80	0.905	0.971	0.963
160	0.938	0.910	0.950

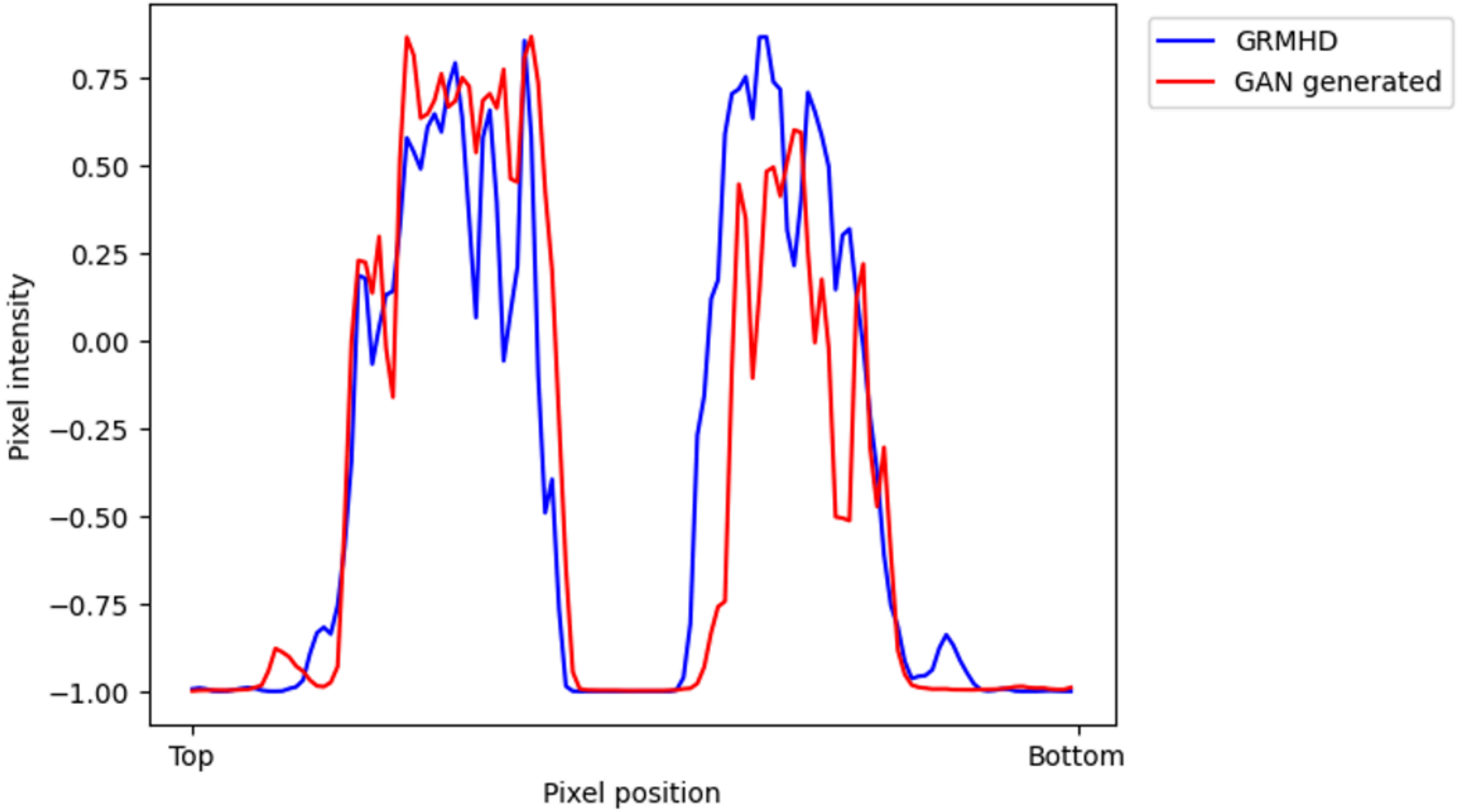
EHT VCWGAN: Results

Pixel flux from top to bottom for $R_{high} = 1$ and $a_* = 0.5$

GRMHD



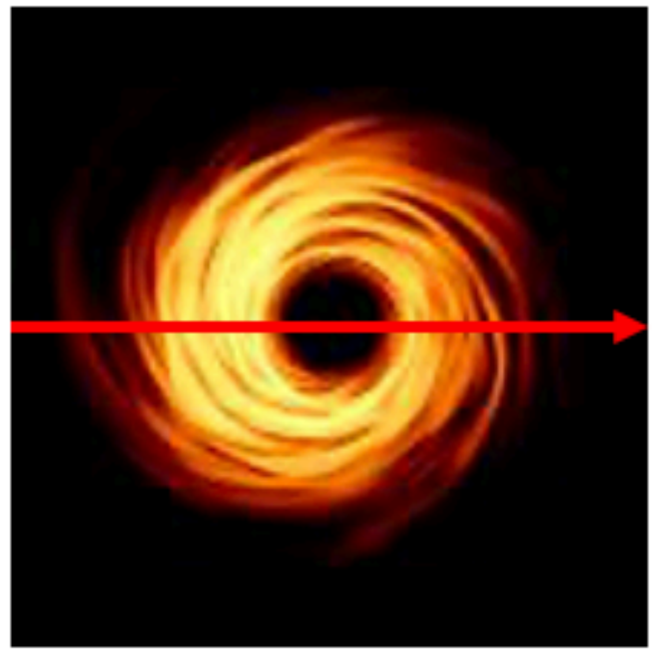
GAN-generated



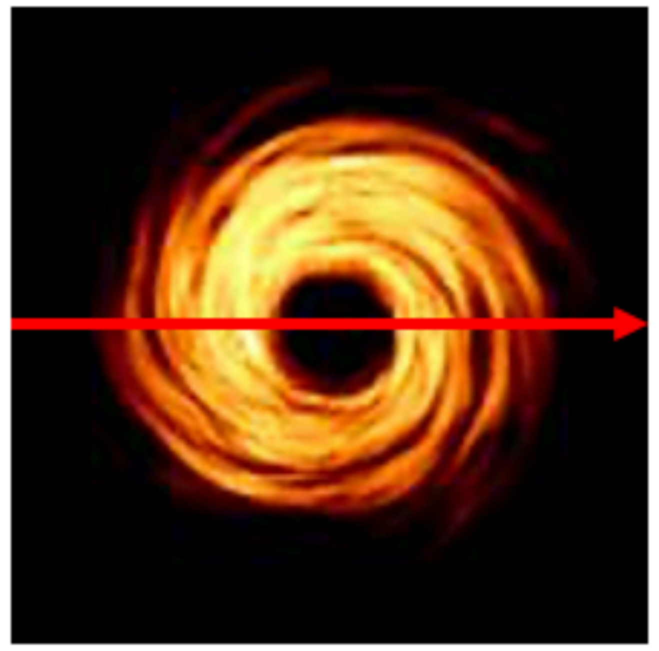
EHT VCWGAN: Results

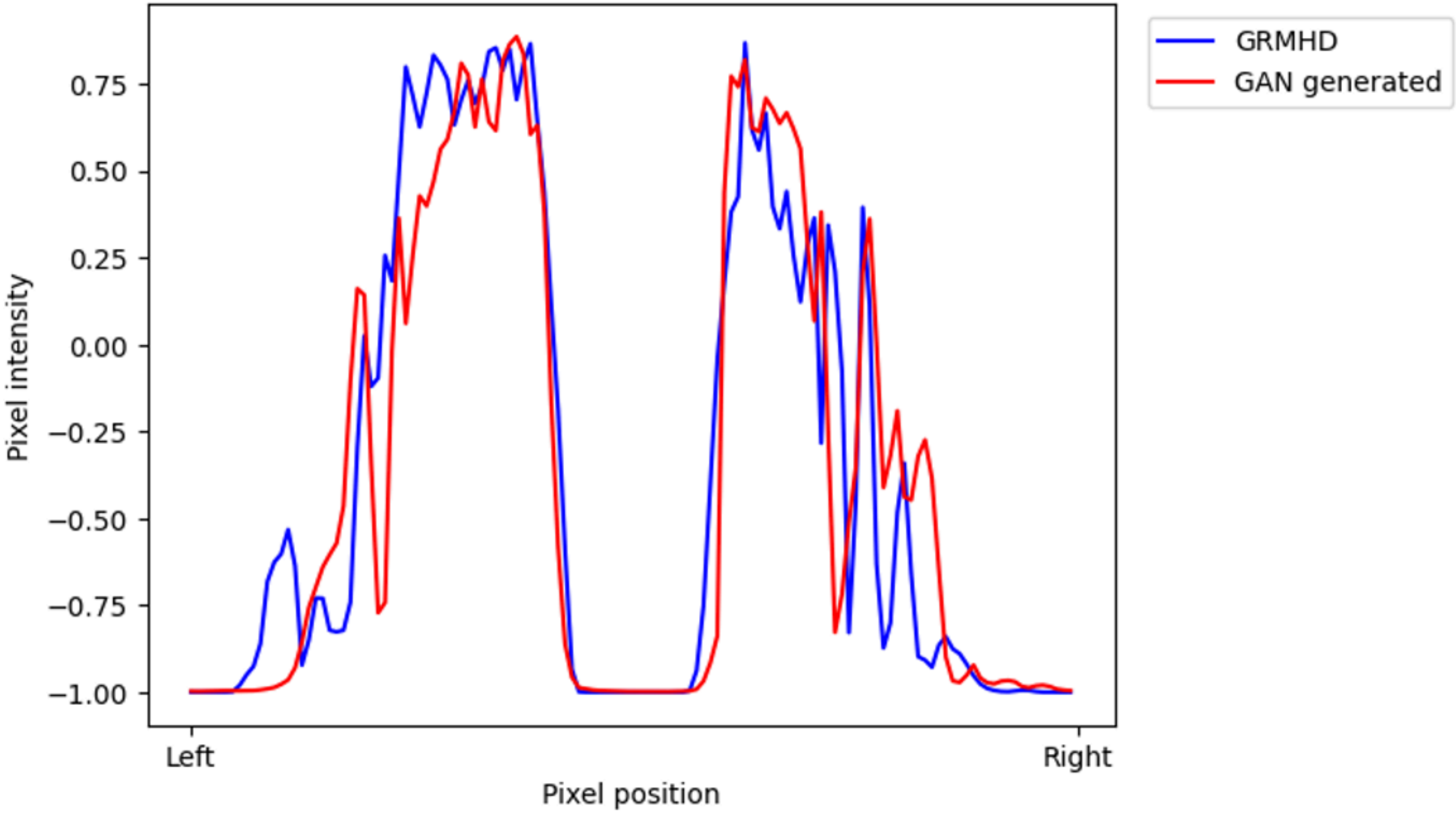
Pixel flux from left to right for $R_{high} = 1$ and $a_* = 0.5$

GRMHD



GAN-generated





EHT VCWGAN: Results

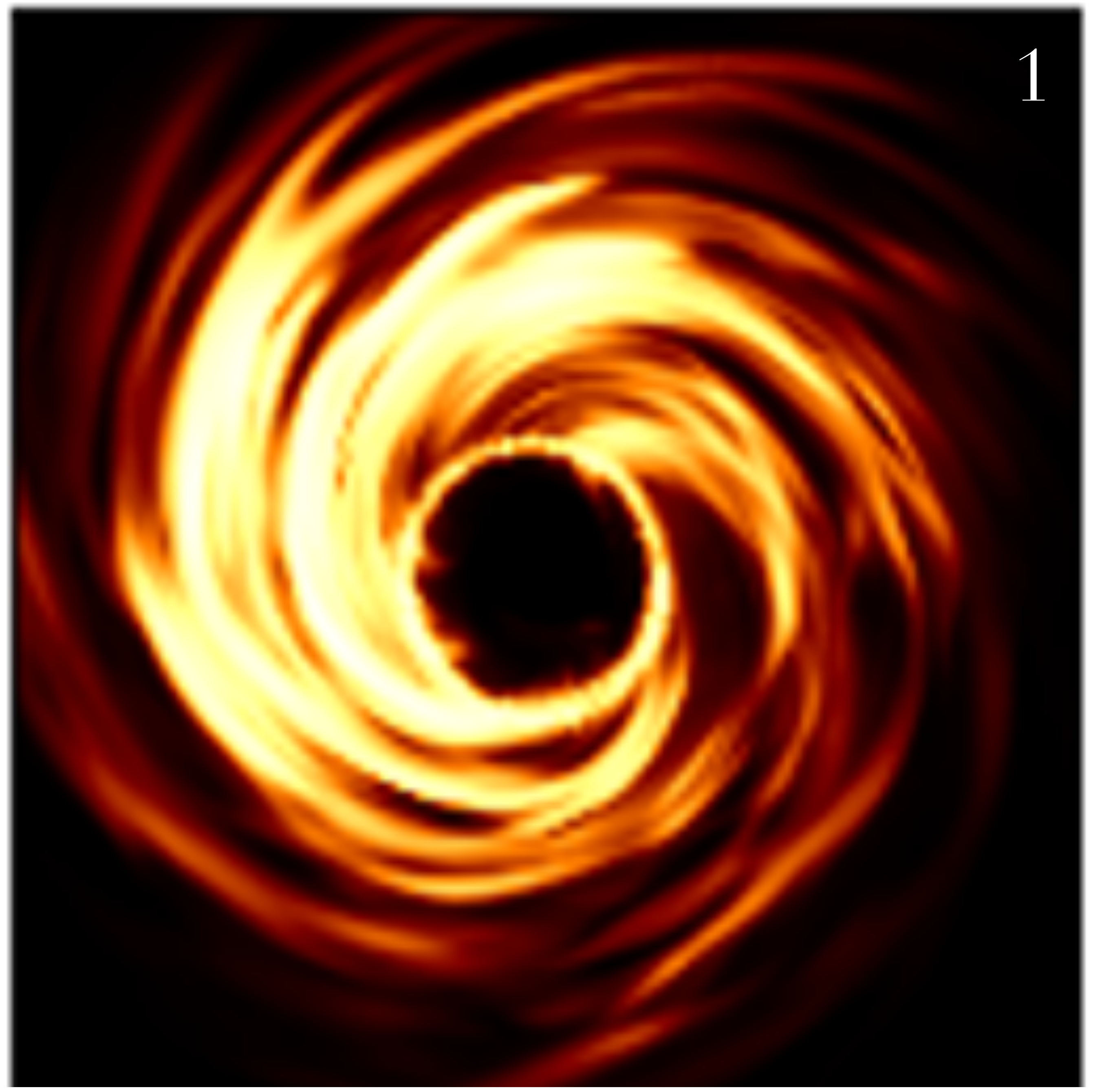
The loss function used to train our CPGAN is as follows:

$$L_G = \mathbb{E}_{x_r \sim p_r}\left[D(x_r)\right] - \mathbb{E}_{x_g \sim p_g}\left[D(x_g)\right] + \sum_{i=1}^{N}\left(a_* - R\left(x_g\right)\right)^2$$

$$L_D = -L_G + \lambda \mathbb{E}_{\hat{x} \sim p_{\hat{x}}}\left[\left(\left\| \nabla_{\hat{x}}(D(\hat{x})\right\|_2 - 1\right)^2\right]$$

$$L_R = \sum_{i=1}^{N}\left(a_* - R\left(x_g\right)\right)^2 + \sum_{i=1}^{N}\left(a_* - R(x_r)\right)^2$$

EHT VCWGAN: Results

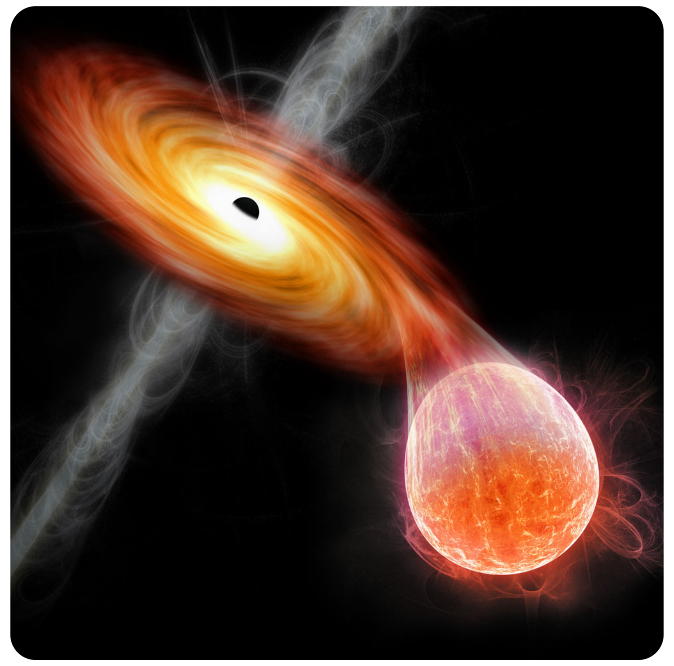


GRMHD

GenAI

GRMHD

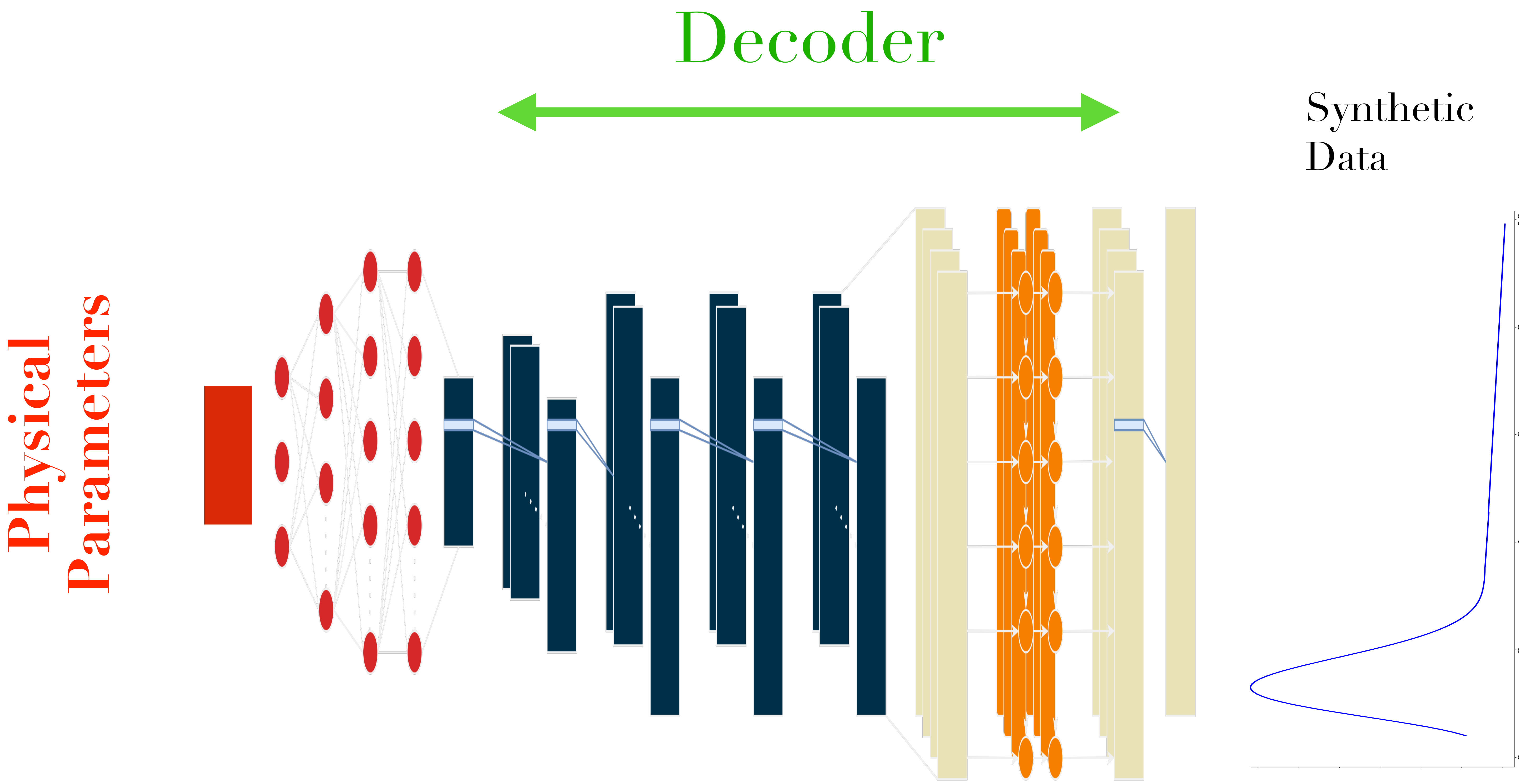
Modelo Físico Semi Supervisado



Physical Parameters

Tregidga et al. 2023

Modelo Físico Semi Supervisado

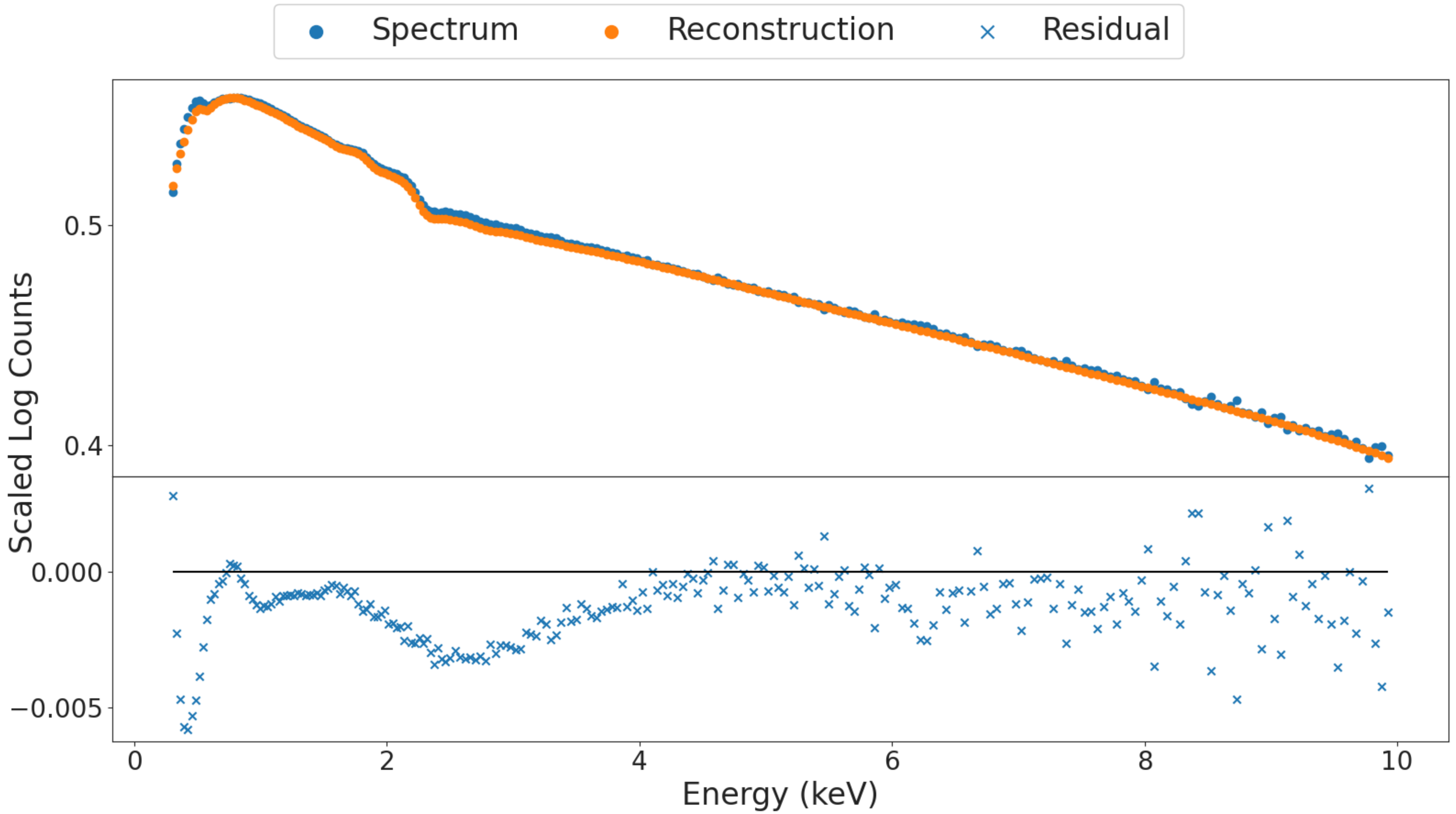


Tregidga et al. 2023

Modelo Físico Semi Supervisado

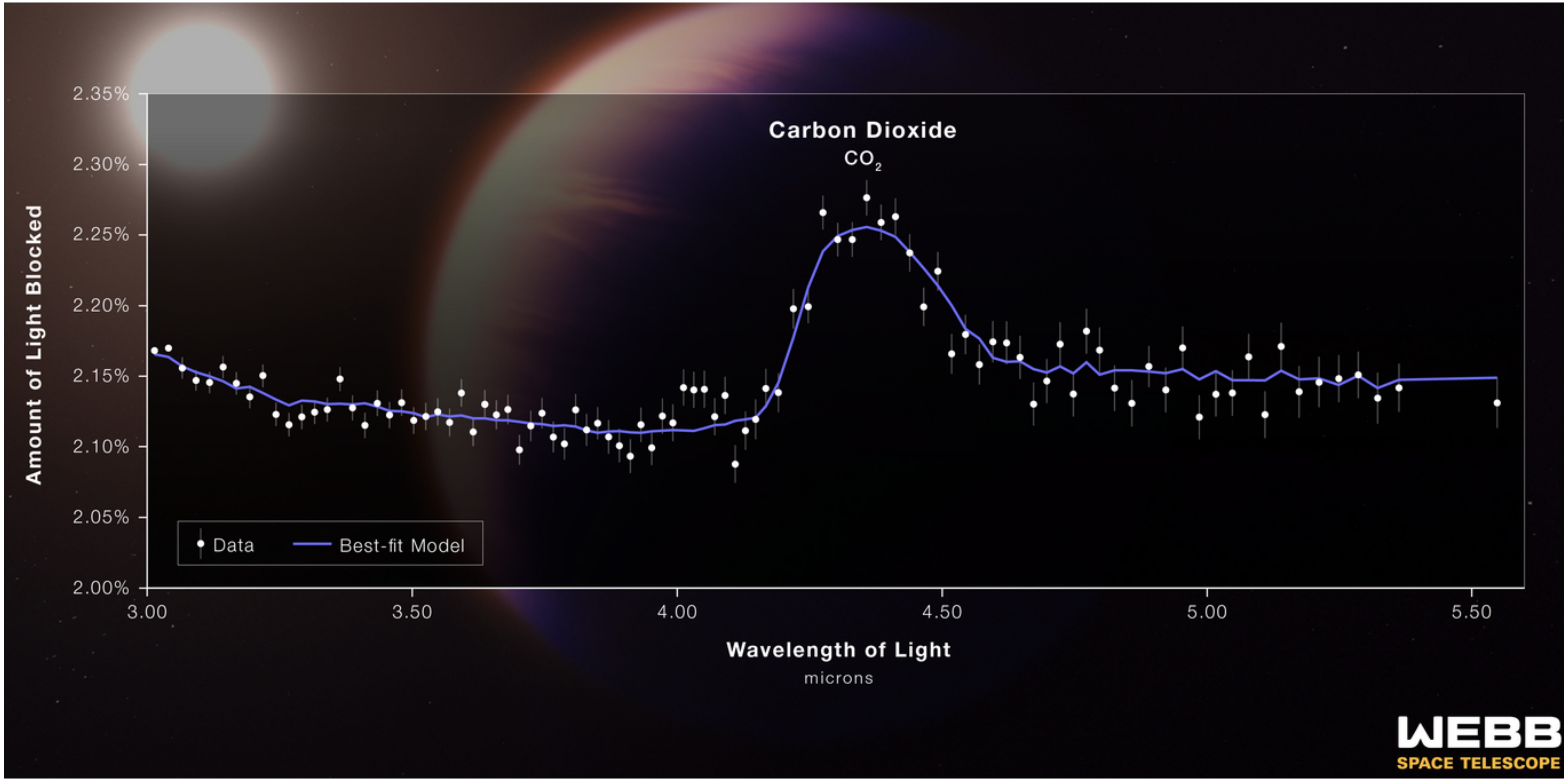
Tregidga et al. 2023

Modelo Físico Semi Supervisado

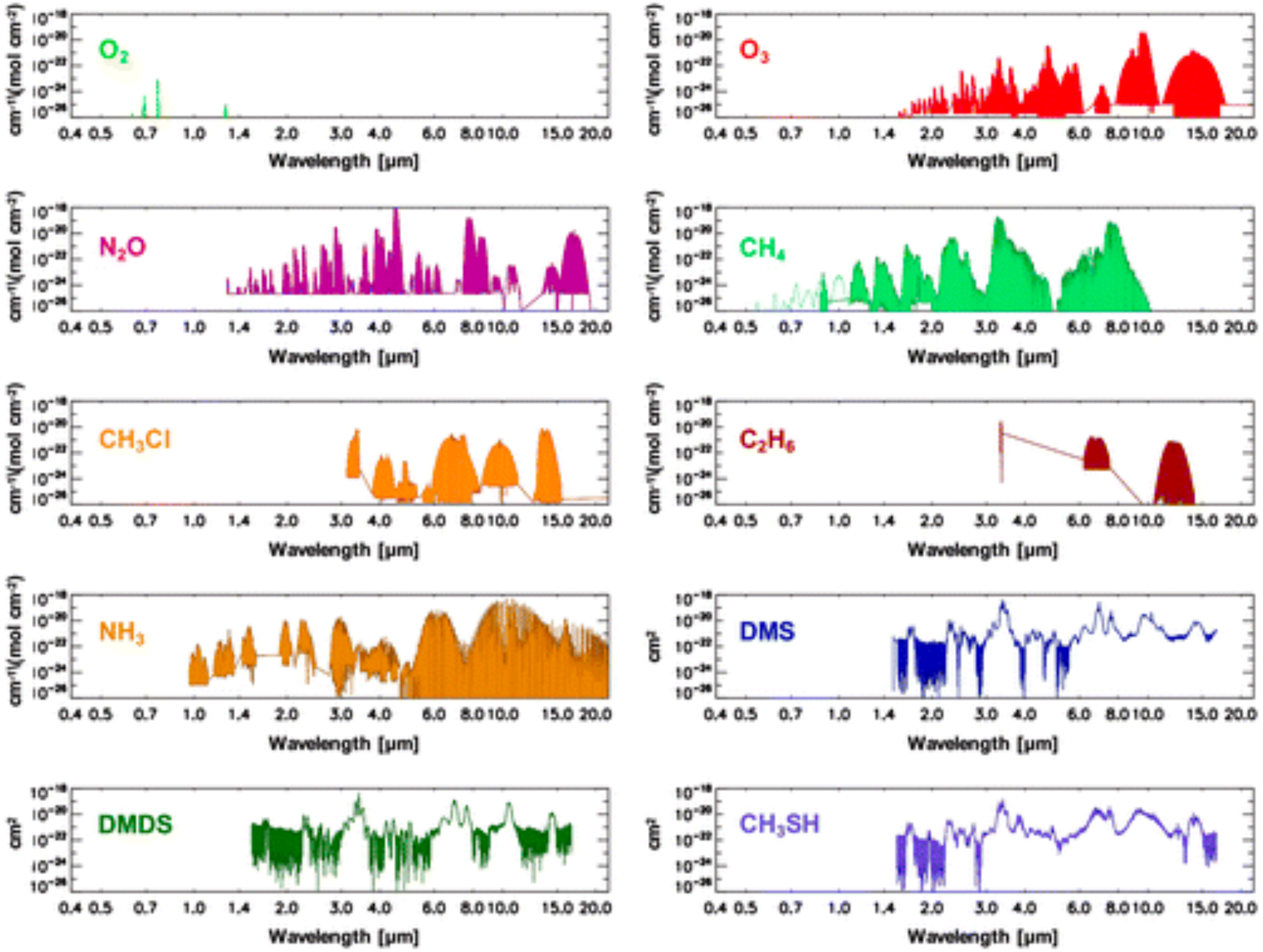


Tregidga et al. 2023

Modelo Físico Semi Supervisado



Modelo Físico Semi Supervisado

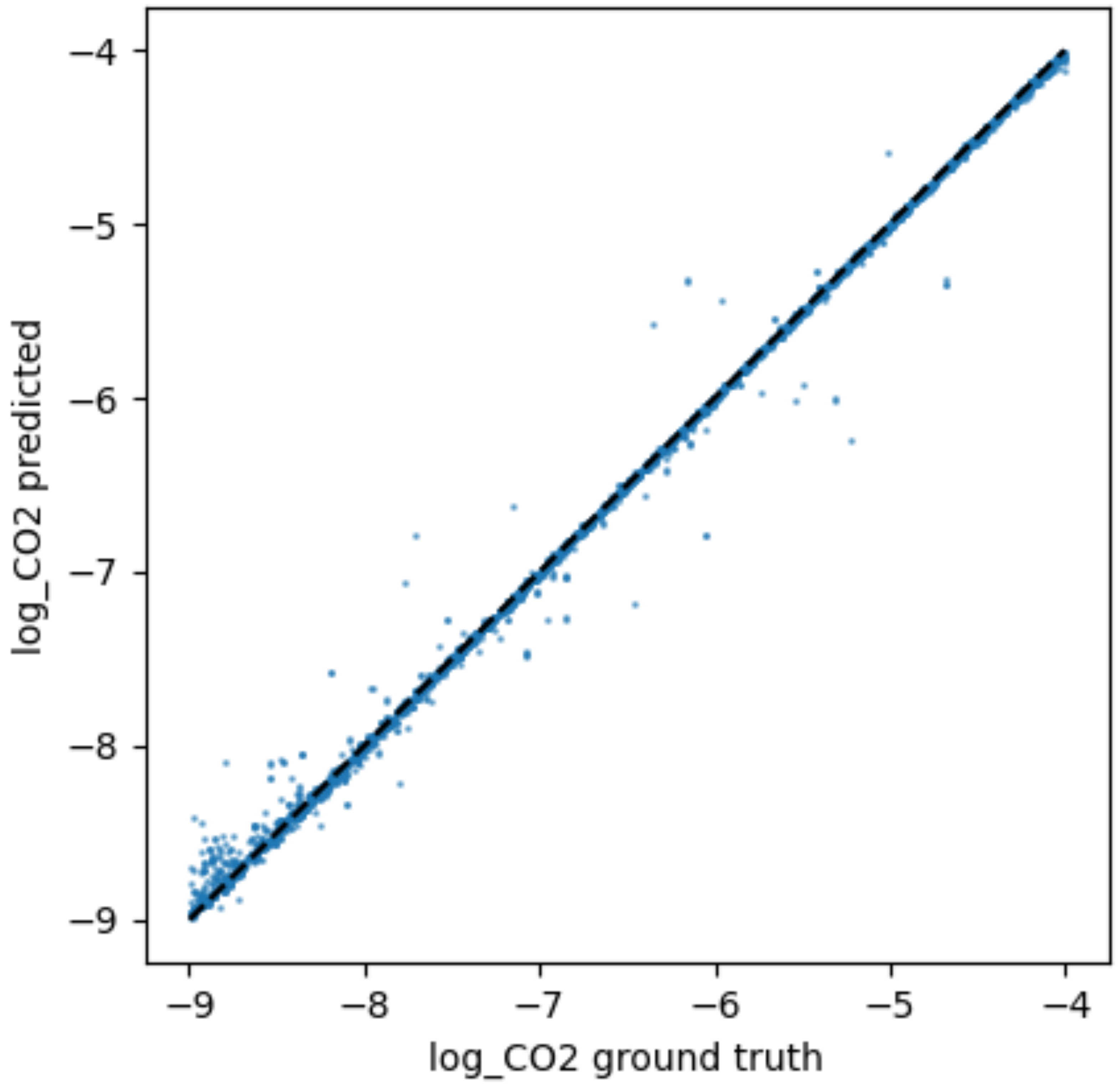


Modelo Físico Semi Supervisado

Figure 2: Reconstruction of a spectrum by the decoder

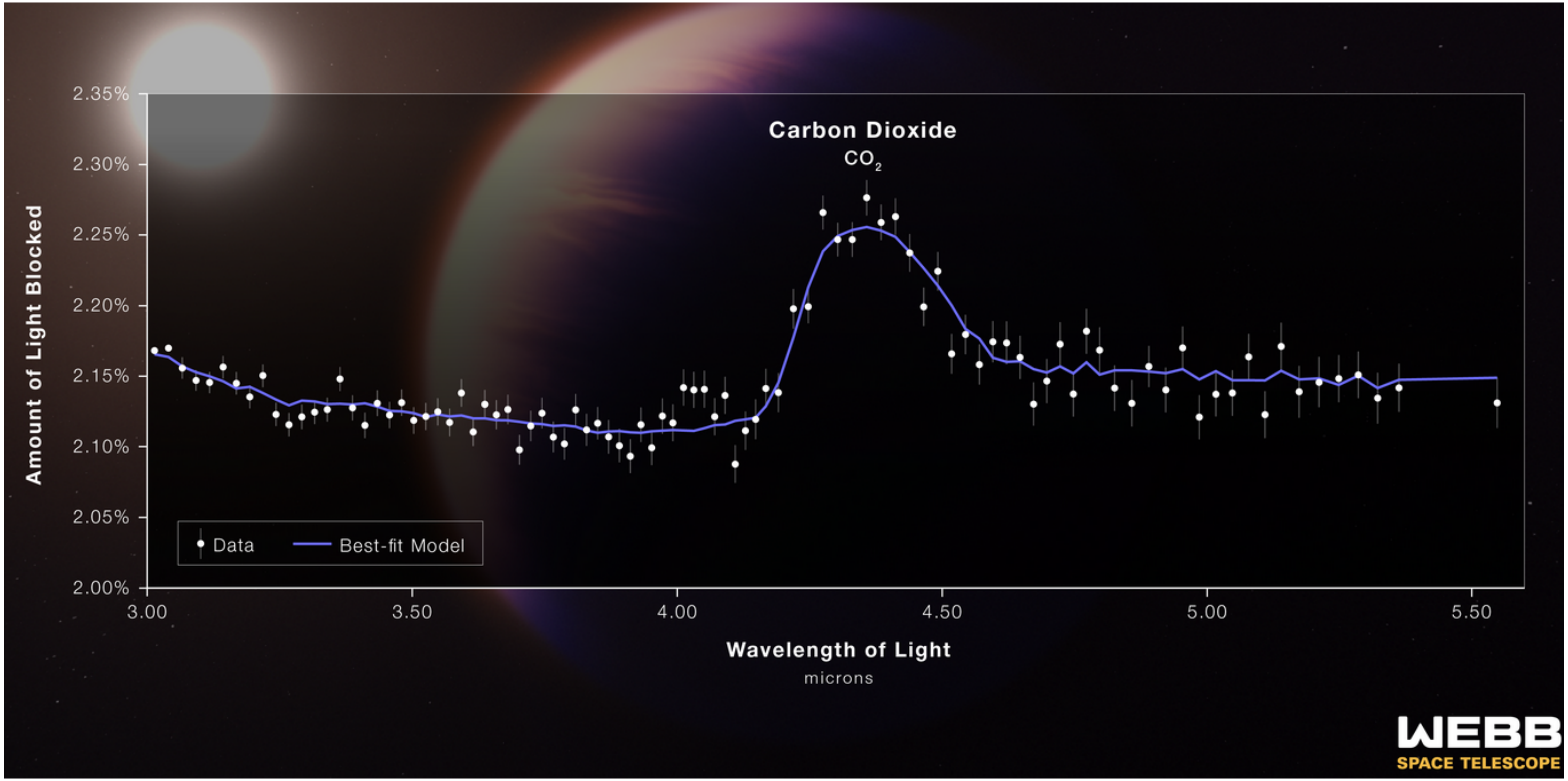
Modelo Físico Semi Supervisado

The encoder can reliably predict the target parameters

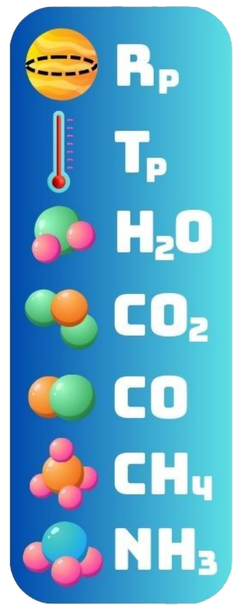


Prediction of the mass fraction of CO2 in the atmosphere by the encoder

Modelo Físico Semi Supervisado



Modelo Físico Semi Supervisado



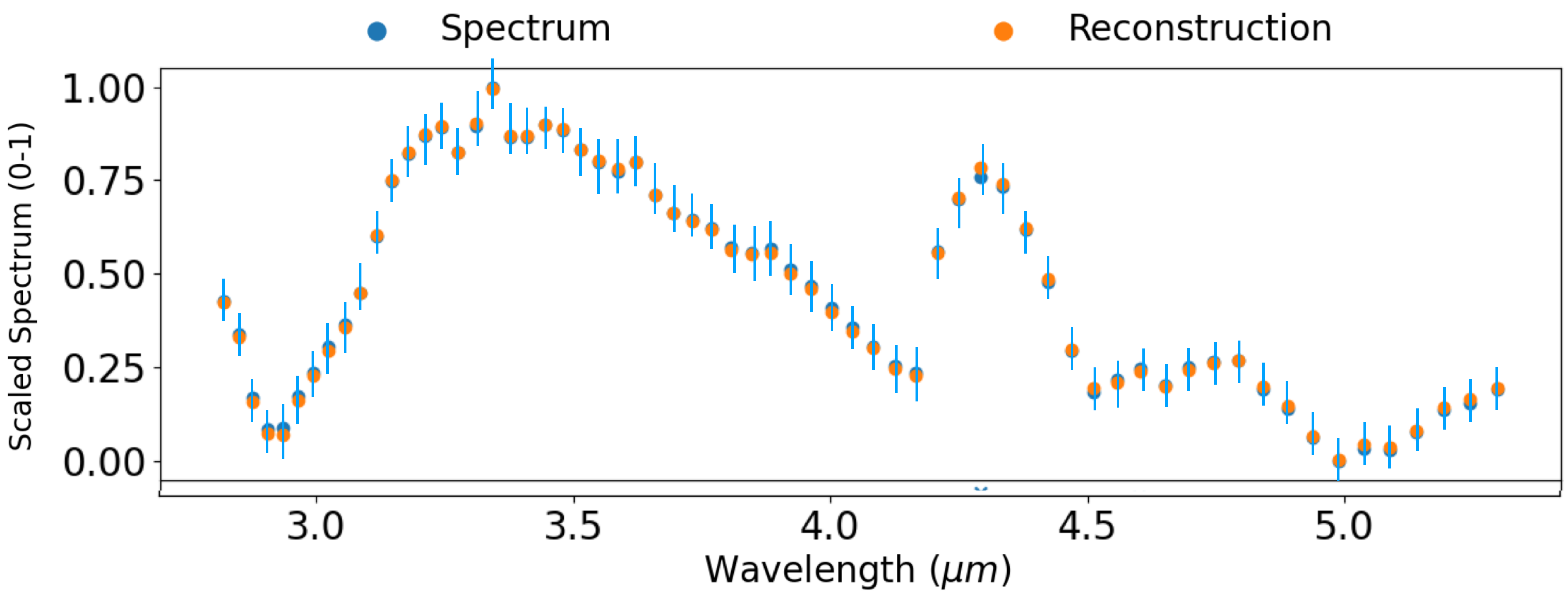
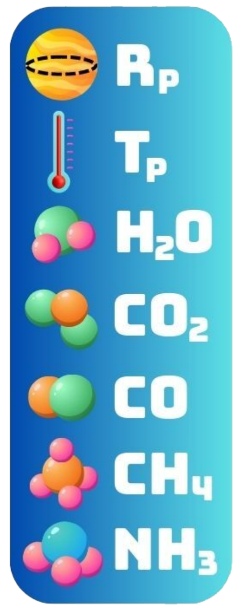


Figure 2: Reconstruction of a spectrum by the decoder

Modelo Físico Semi Supervisado



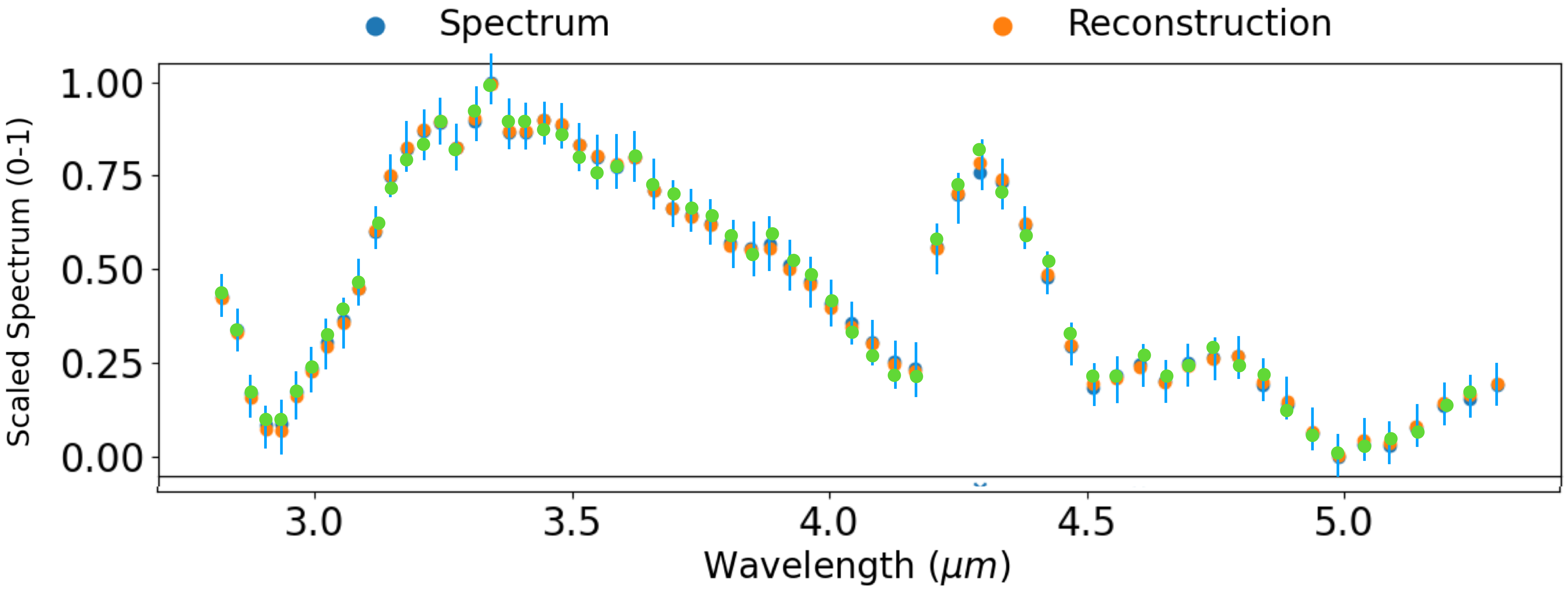
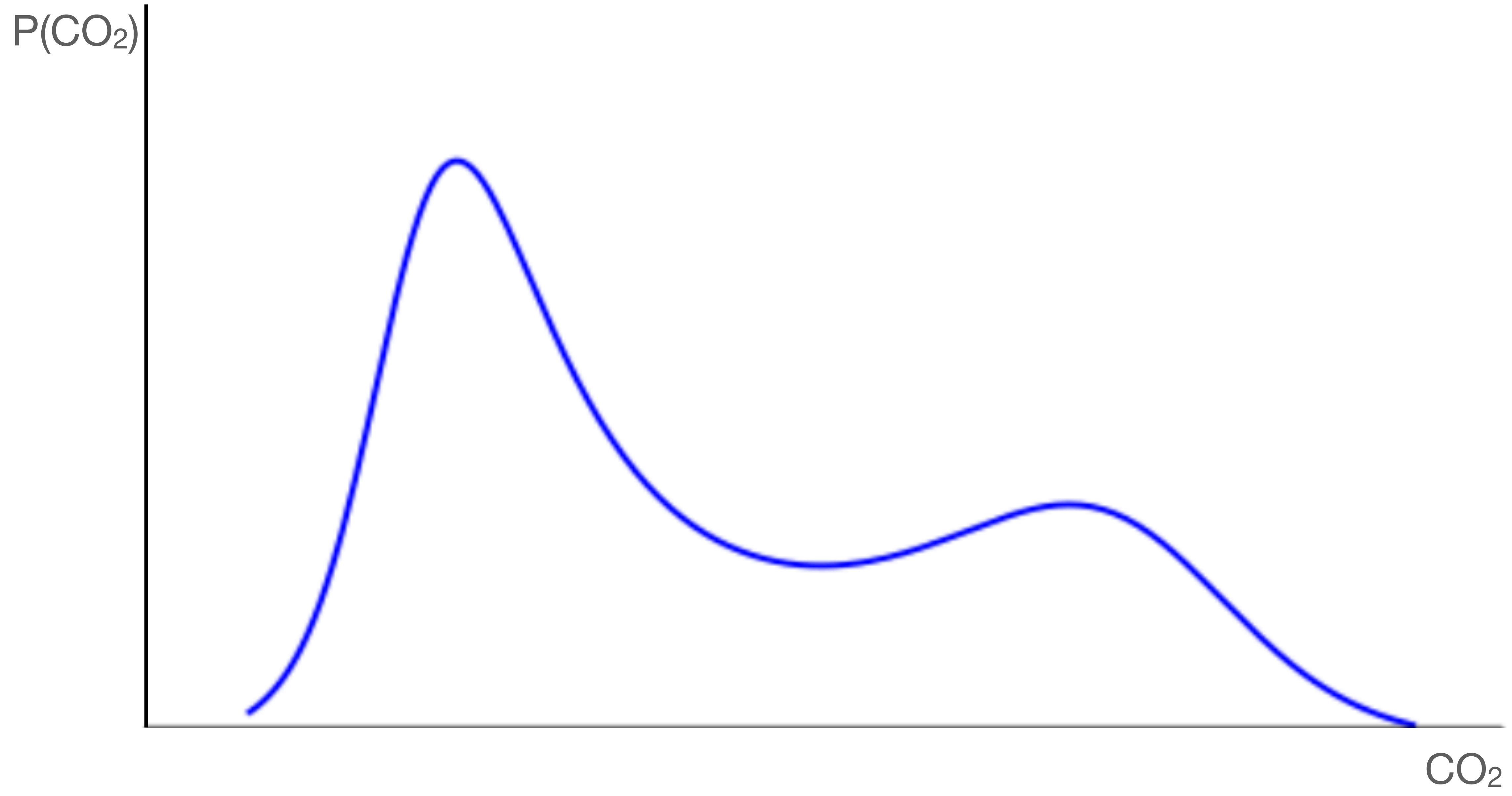
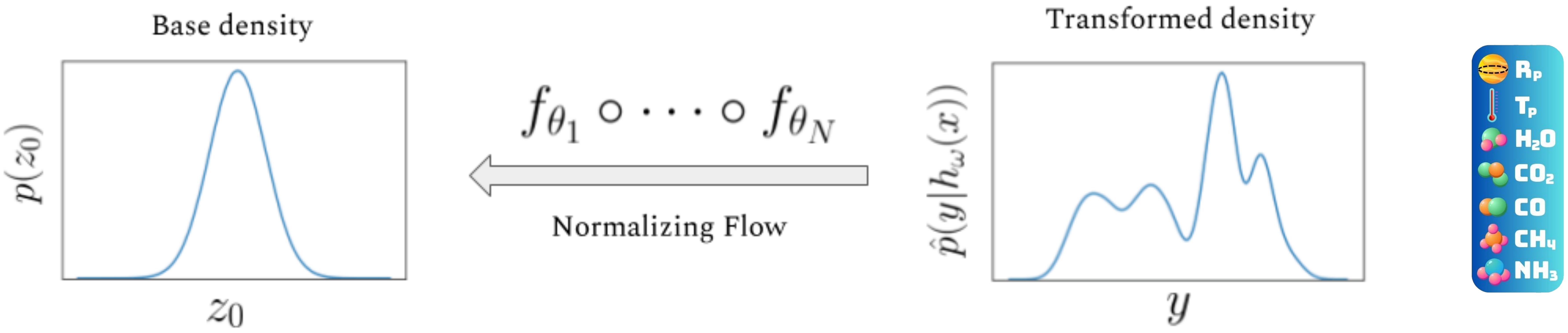


Figure 2: Reconstruction of a spectrum by the decoder

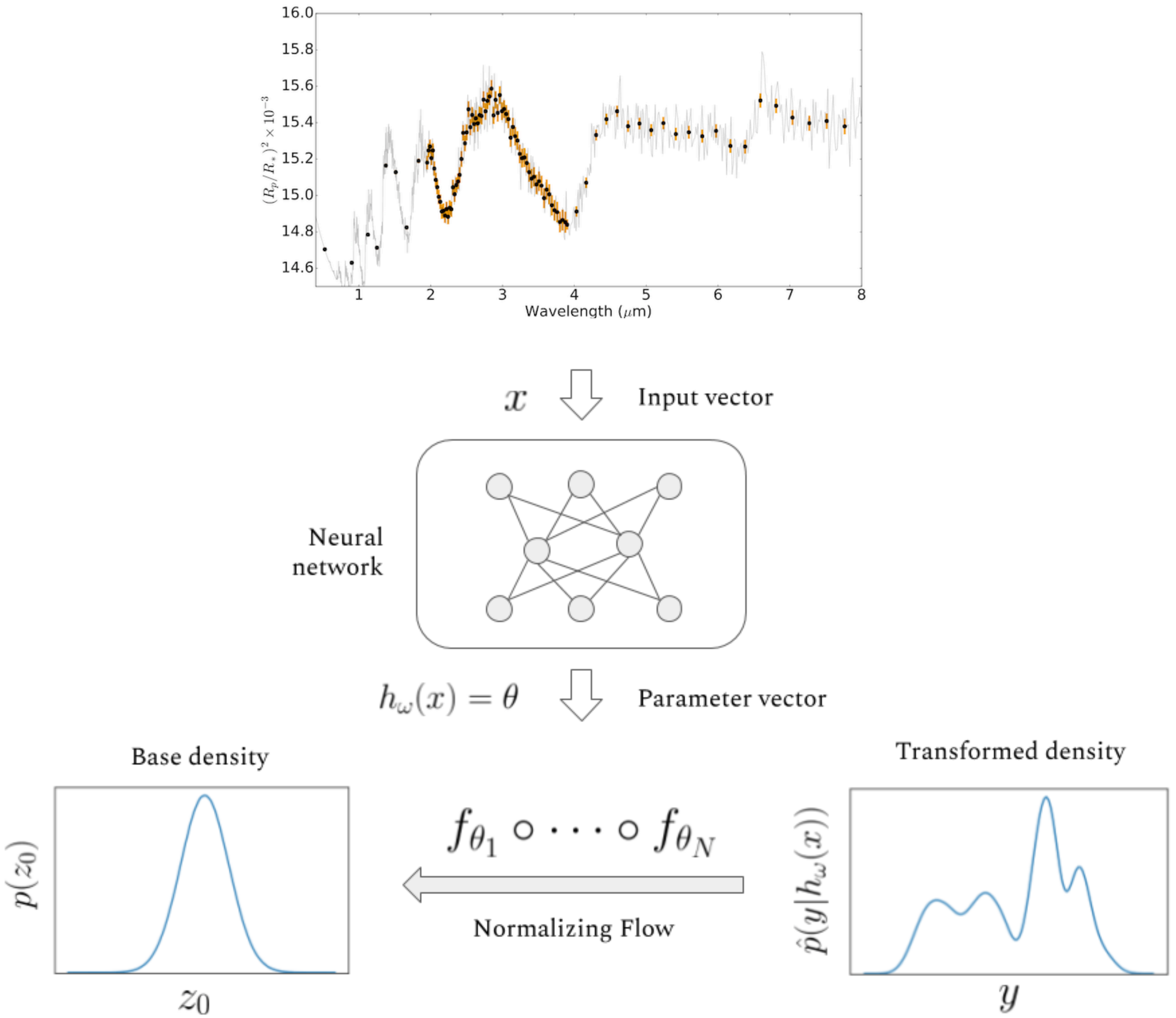
Modelo Probabilistico



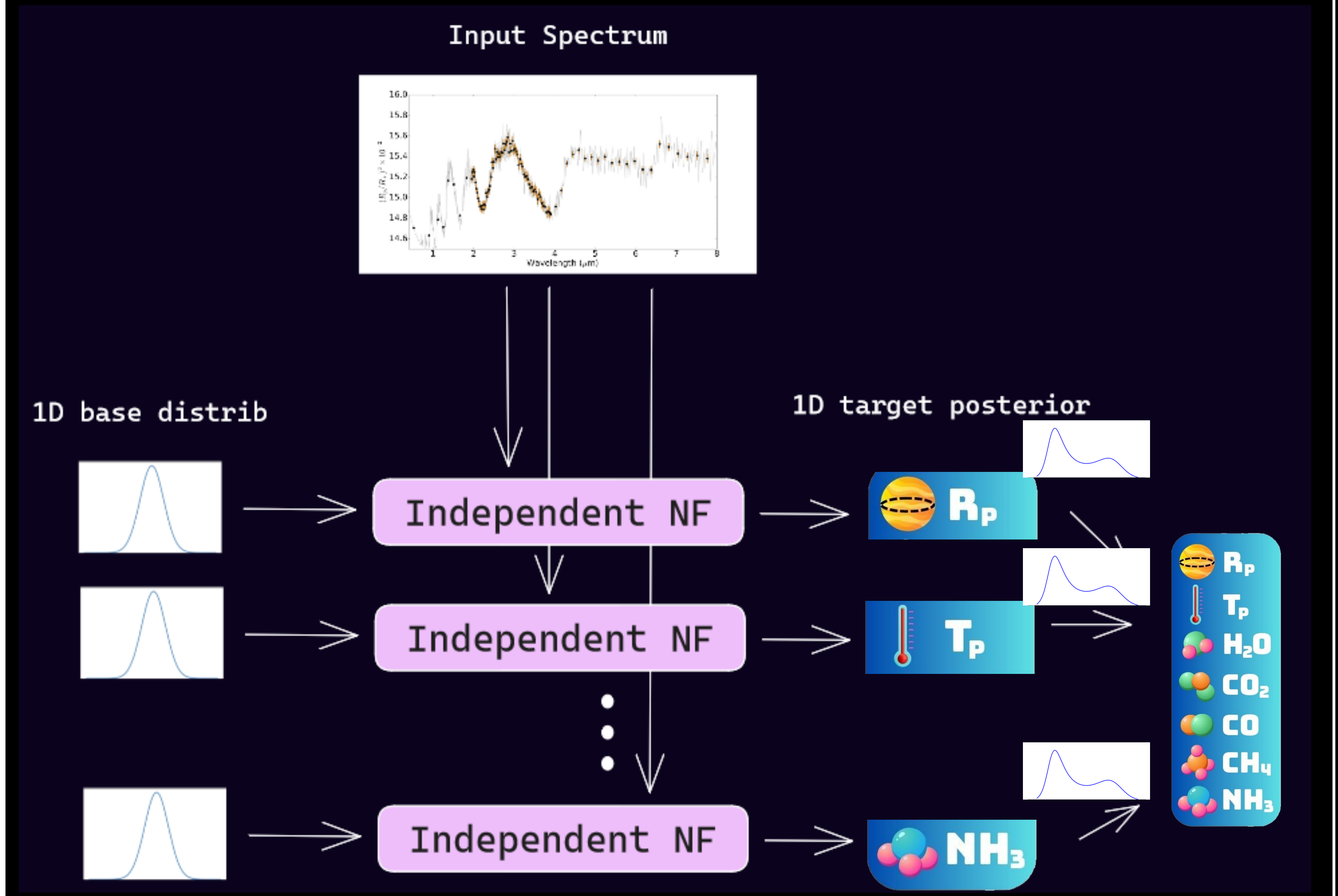
Modelo Probabilistico: Flujos Normalizantes



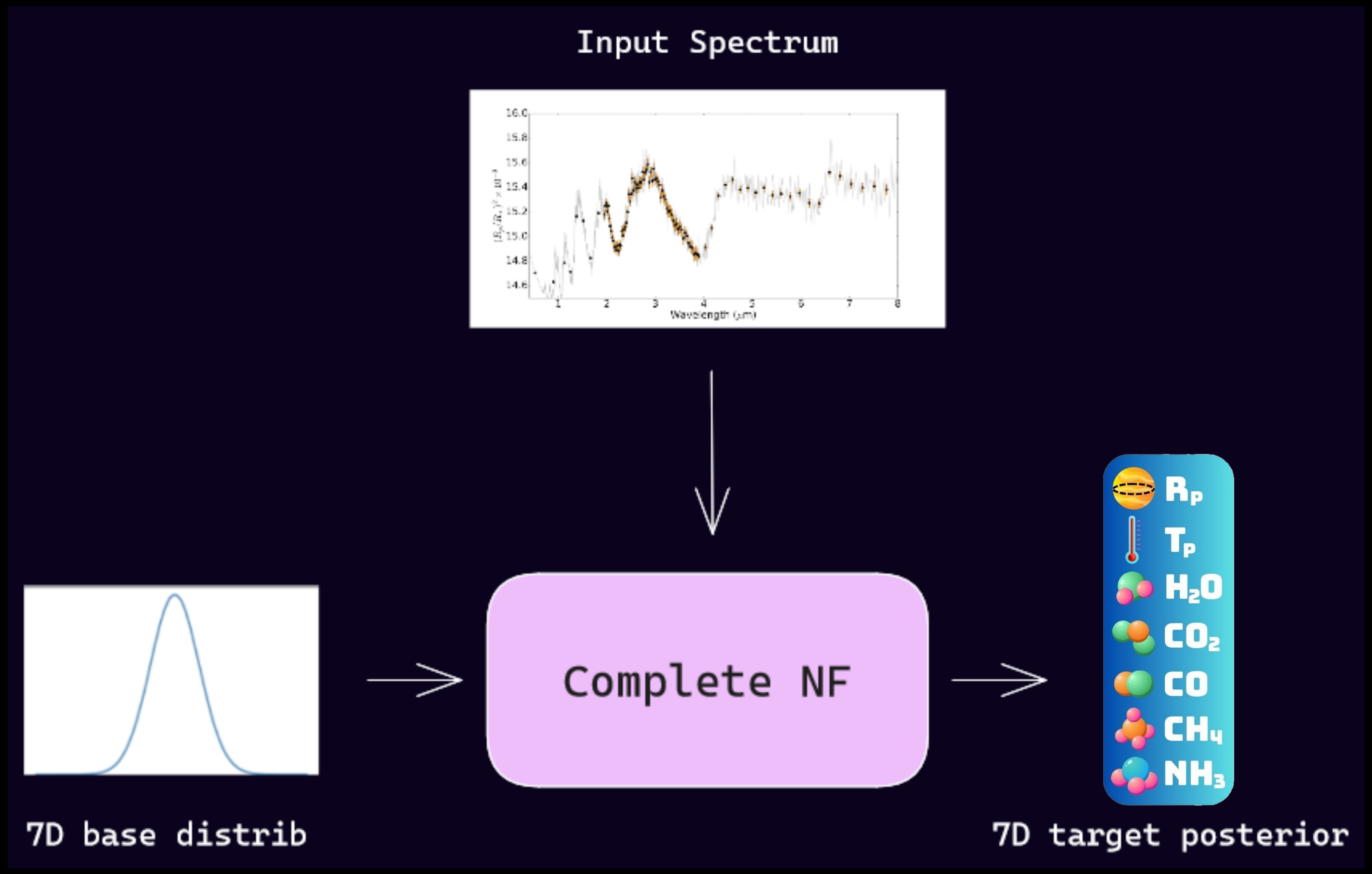
Modelo Probabilistico: Flujos Normalizantes



Modelo Probabilistico: Flujos Normalizantes



A set of independent Normalising Flows



One complete Normalising Flow

Modelos de Difusión

DALL-E generated

Modelos de Difusión

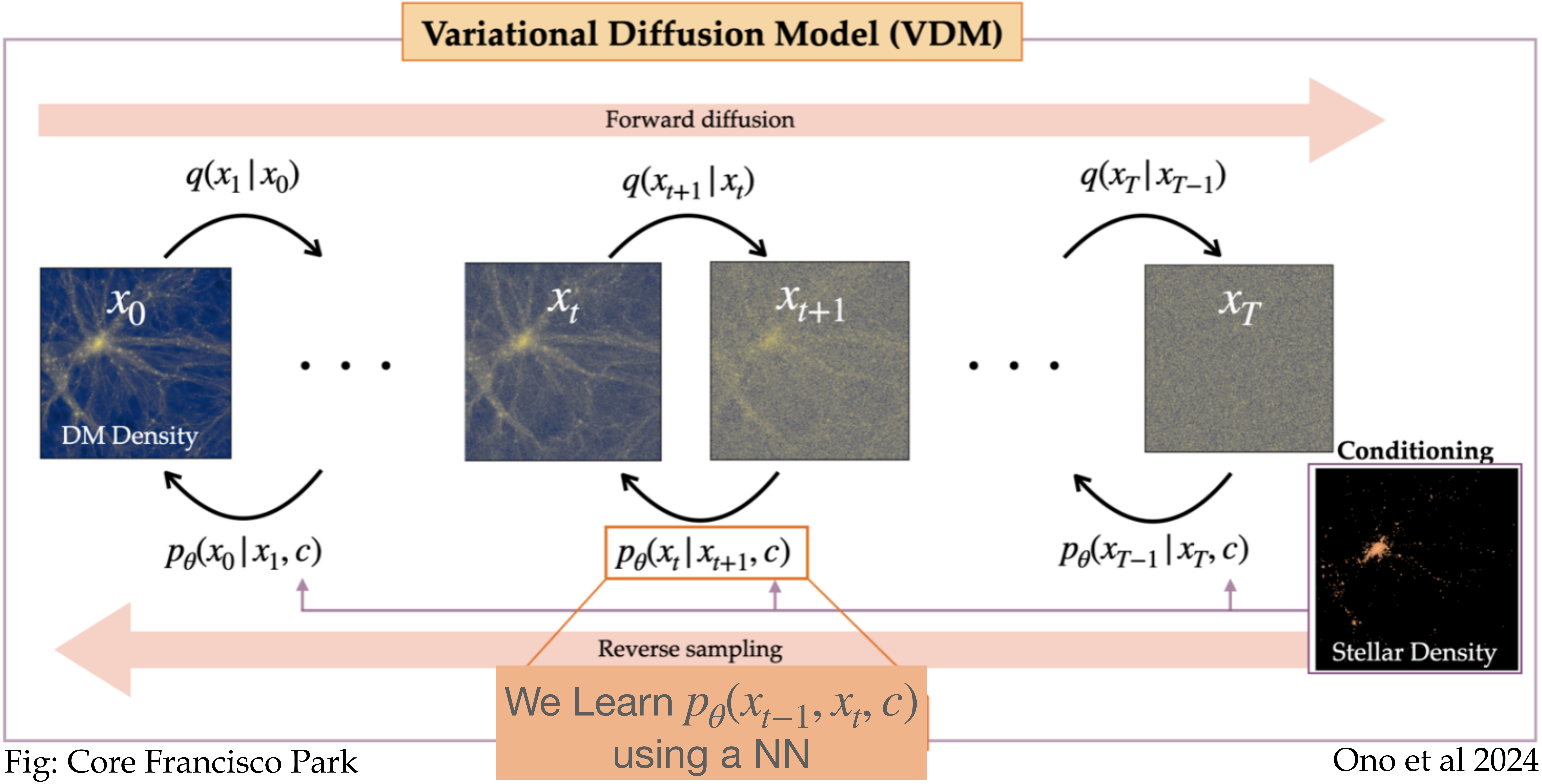
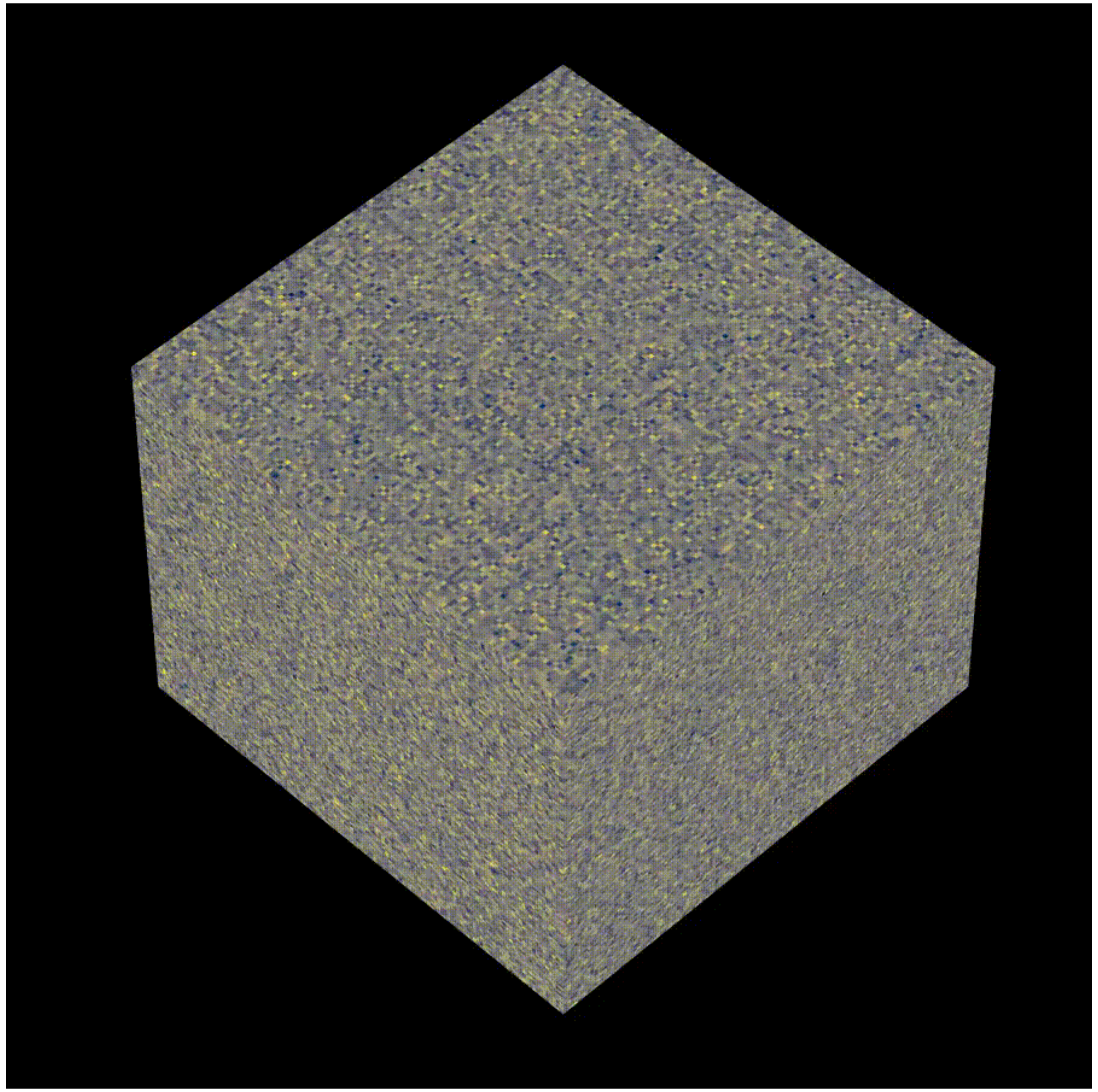


Fig: Core Francisco Park

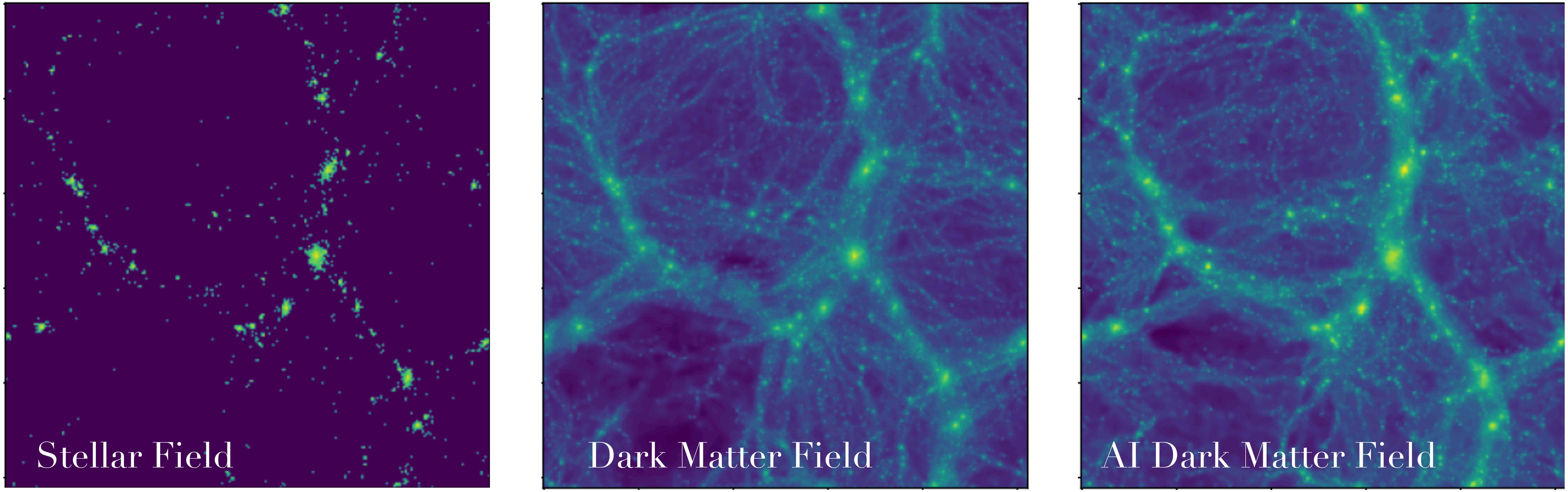
Ono et al 2024

Modelos de Difusión



Modelos de Difusión

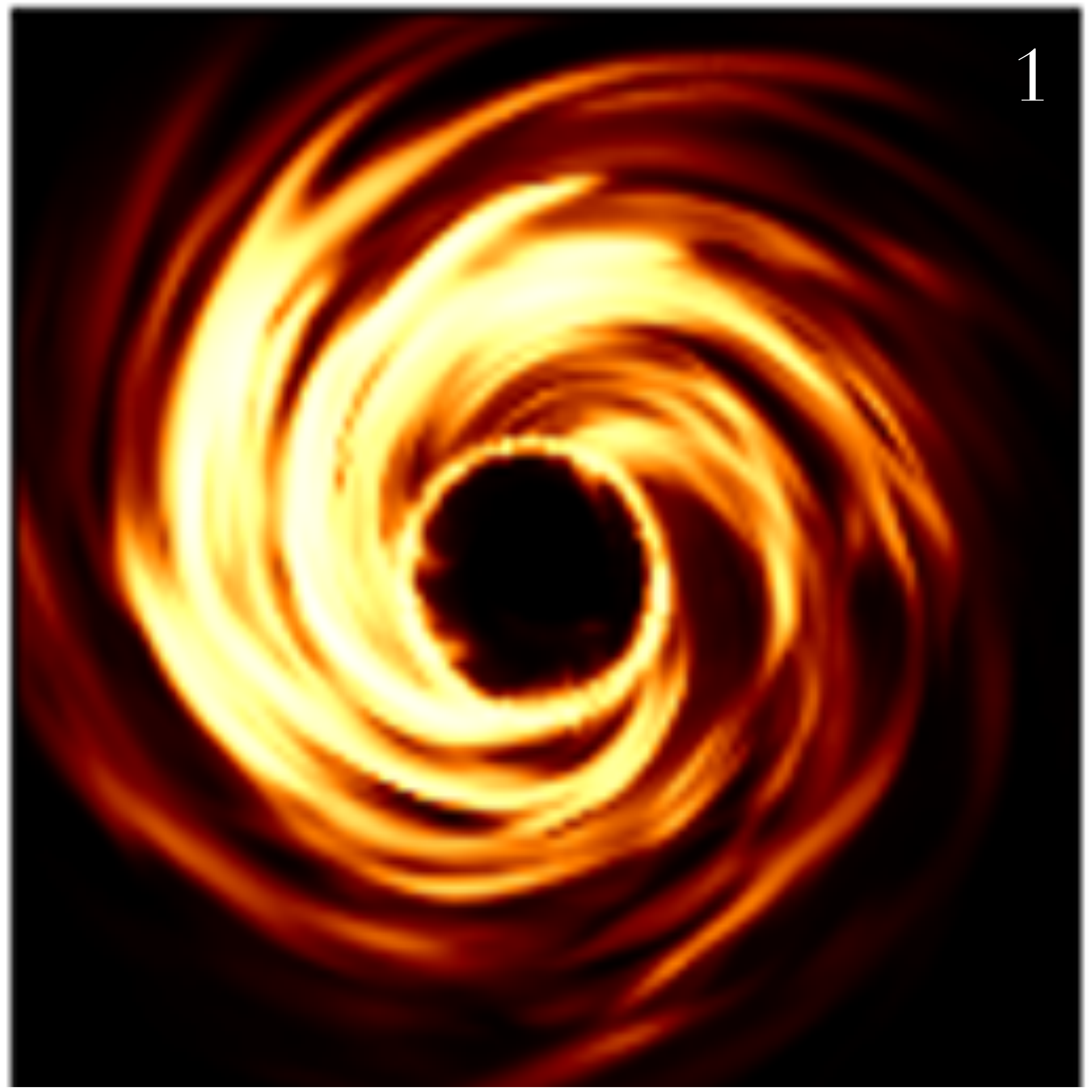
Dark Matter Field Simulations



Core Francisco Park , Victoria Ono, Yueying Ni, Carolina Cuesta-Lázaro, Francisco Villaescusa-Navarro

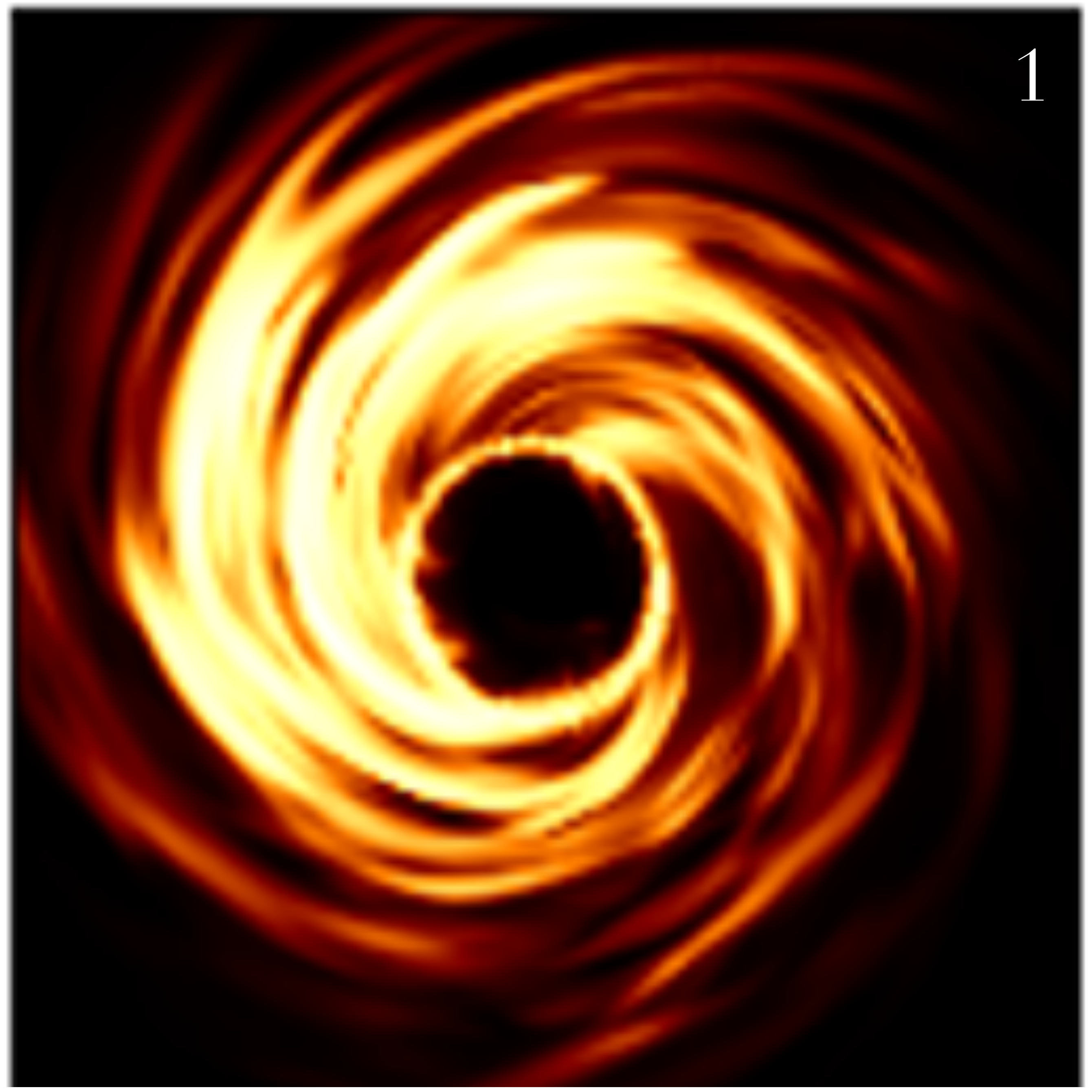
Modelos de Difusión

Supermassive BHs



Modelos de Difusión

Supermassive BHs



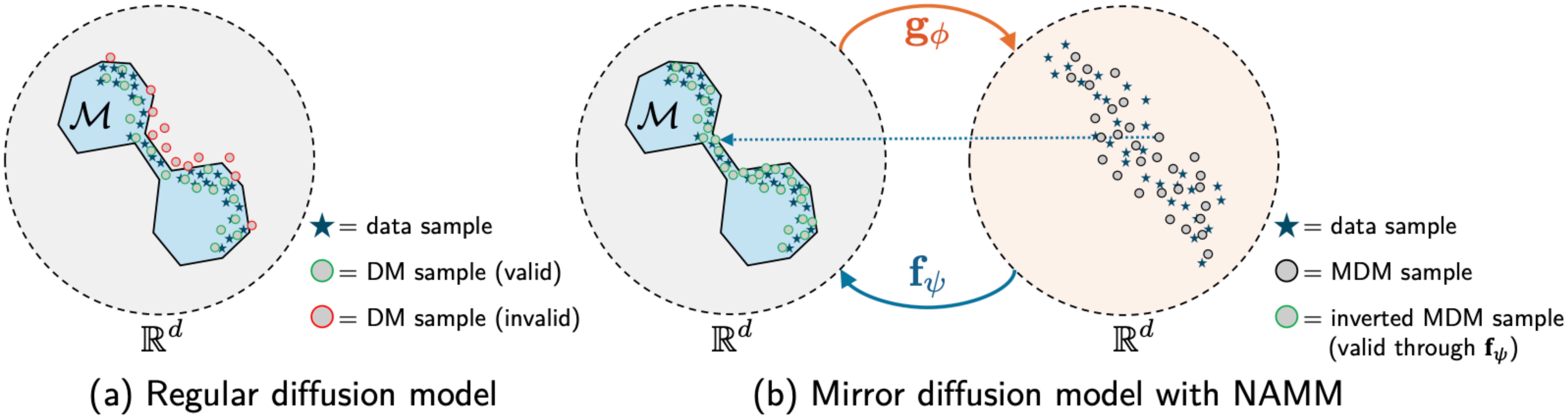
GRMHD

GenAI

GRMHD

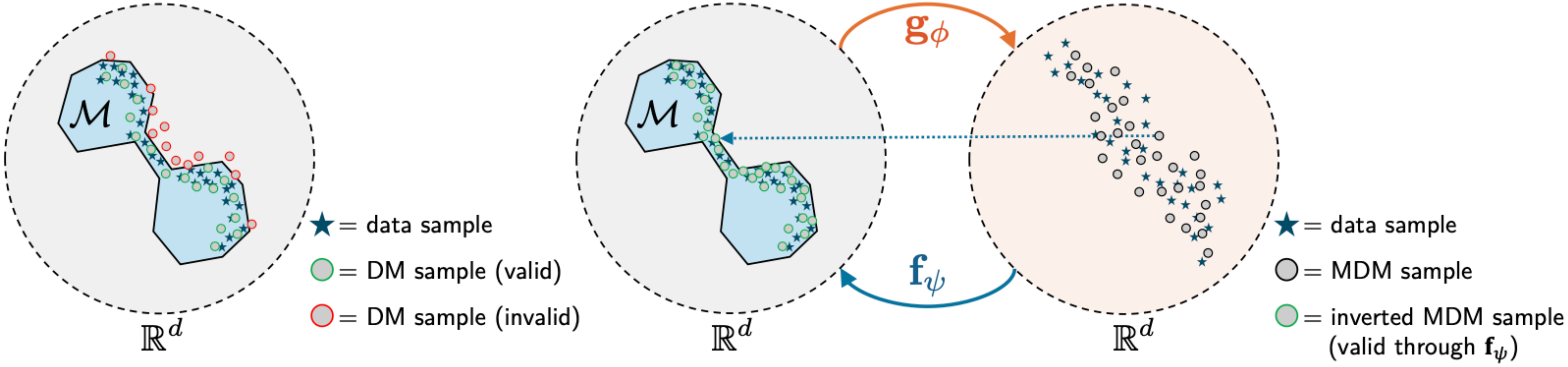
What does realistic mean?

Modelos de Difusión Constrained



Berthy Feng, Ricardo Baptista and Katie Bouman NeurIPS 2024

Constrained Diffusion Models



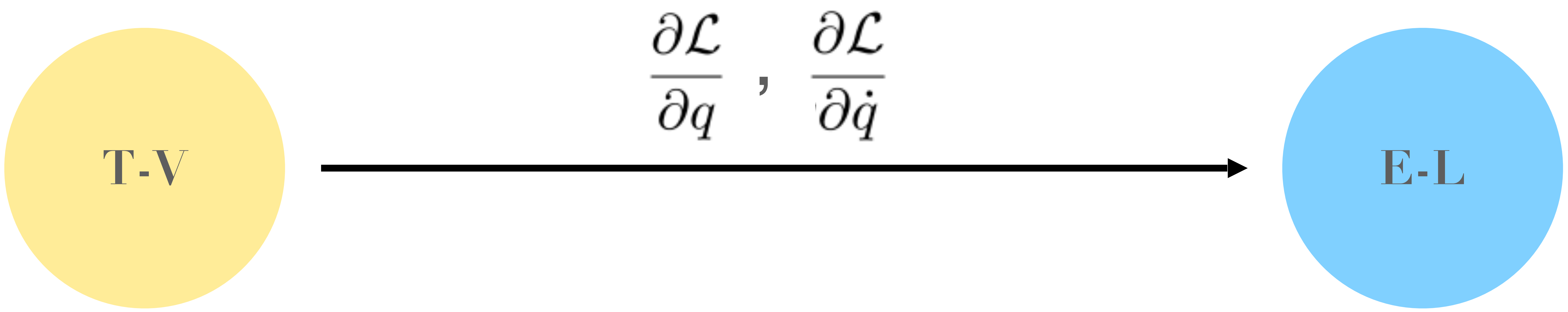
(a) Regular diffusion model

(b) Mirror diffusion model with NAMM

Berthy Feng, Ricardo Baptista and Katie Bouman NeurIPS 2024

Lagrangian Formulation

Most physical systems can be described by a **Lagrangian** (**$\mathcal{L}$** = **T - V**), from which we derive the **Euler-Lagrange** equations of motion and, through symmetries, **conservation laws**.



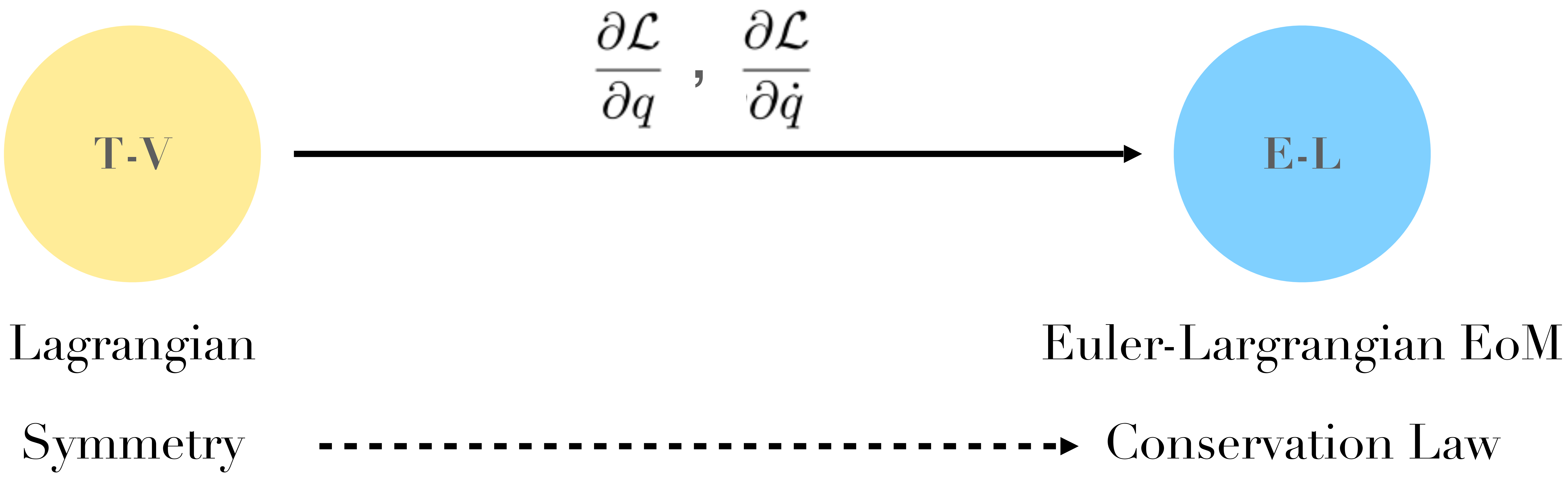
Lagrangian

Euler-Largrangian EoM

$$\ddot{q} = \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \left(\frac{\partial \mathcal{L}}{\partial q} - \dot{q}\frac{\partial^2 \mathcal{L}}{\partial q \partial \dot{q}}\right)$$

Lagrangian Formulation: Generalized Coordinates

Most physical systems can be described by a **Lagrangian** ($\mathcal{L}$ = **T - V**), from which we derive the **Euler-Lagrange** equations of motion and, through symmetries, **conservation laws**.



(e.g., time invariance ⇒ energy conservation)

Hamiltonian Formulation: Cannonical Coordinates

Most physical systems can be described by a **Lagrangian** ($\boldsymbol{\mathcal{L}}$ = **T - V**), from which we derive the **Euler-Lagrange** equations of motion and, through symmetries, **conservation laws**.

$$\textbf{T+V} \xrightarrow{\dfrac{\partial H}{\partial q},\ \dfrac{\partial H}{\partial p}} \textbf{H-E}$$

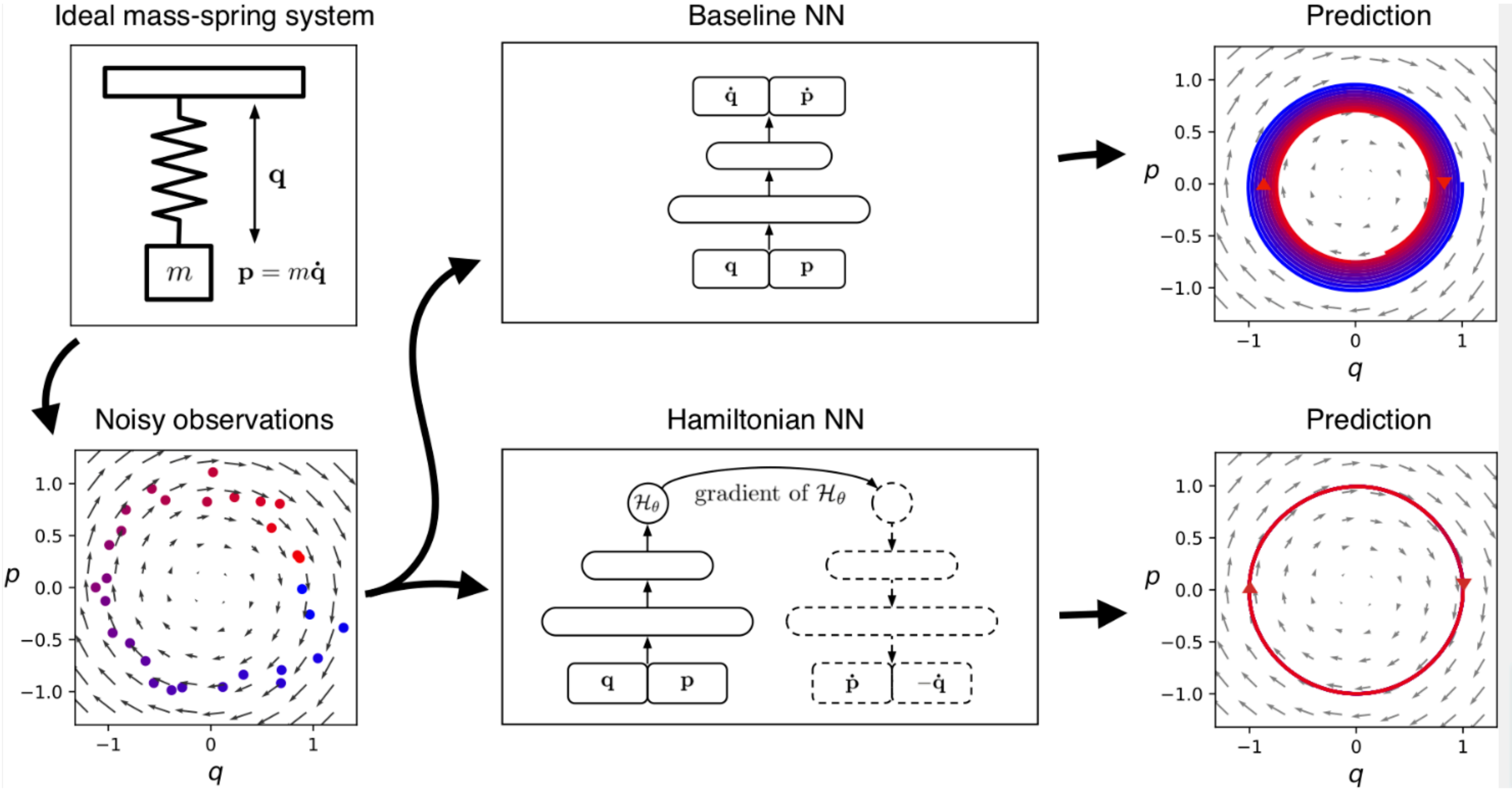
Hamiltonian

Hamilton's Equations

Symmetry $\dashrightarrow$ Conservation Law

(e.g., time invariance ⇒ energy conservation)

Hamiltonian Neural Networks

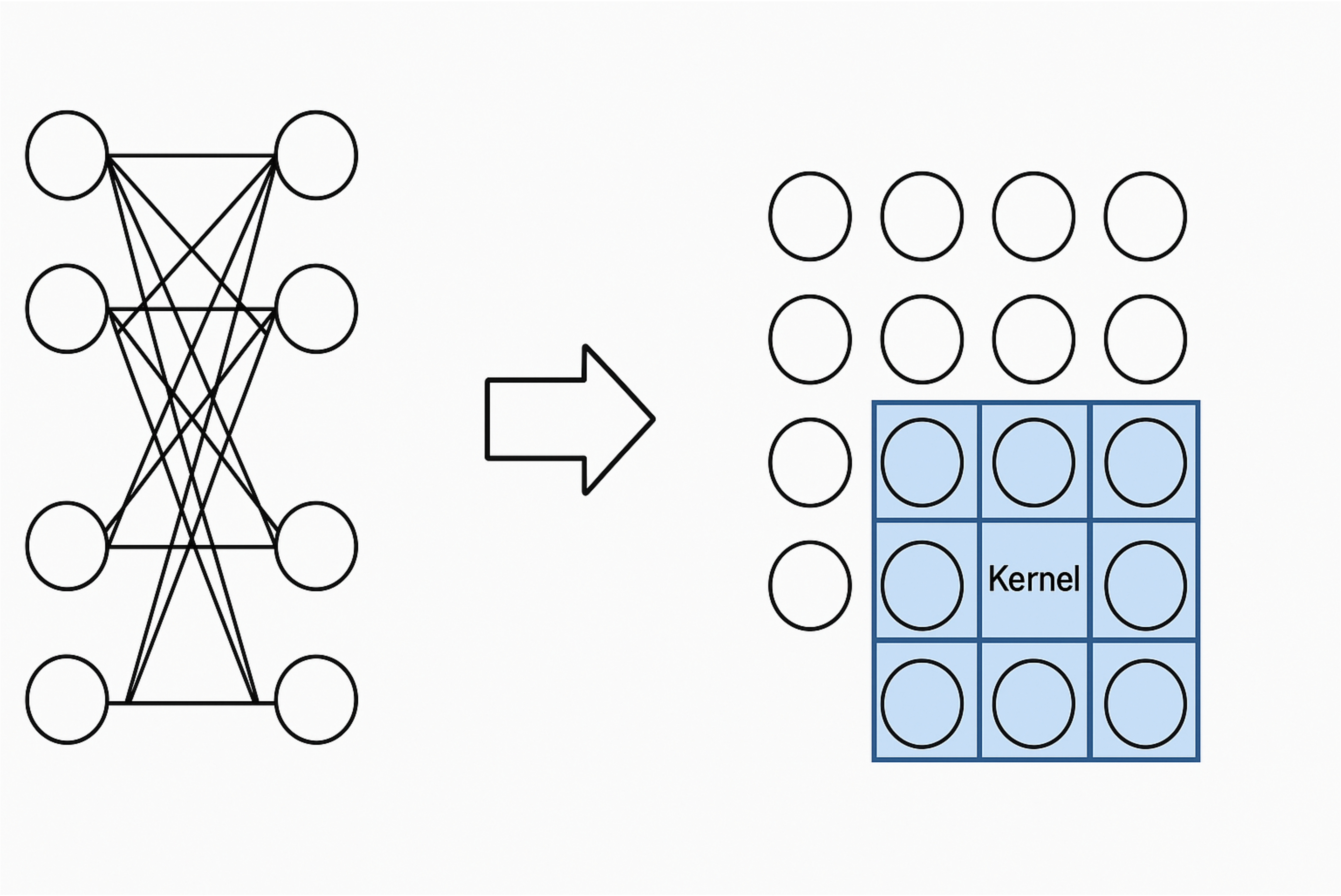


Greydanus et al., 2019

Computer Vision: Introducing Symmetry

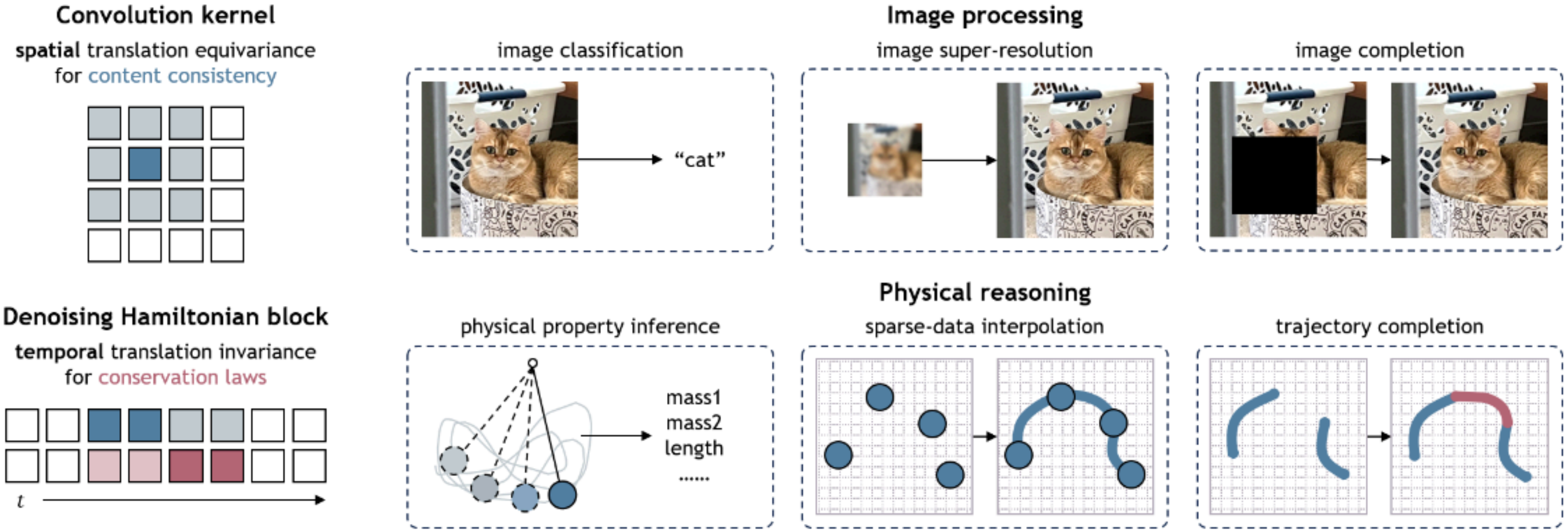
Multi Layer Perceptron

Convolutional Neural Networks



Denoising Hamiltonian Network for Physical Reasoning

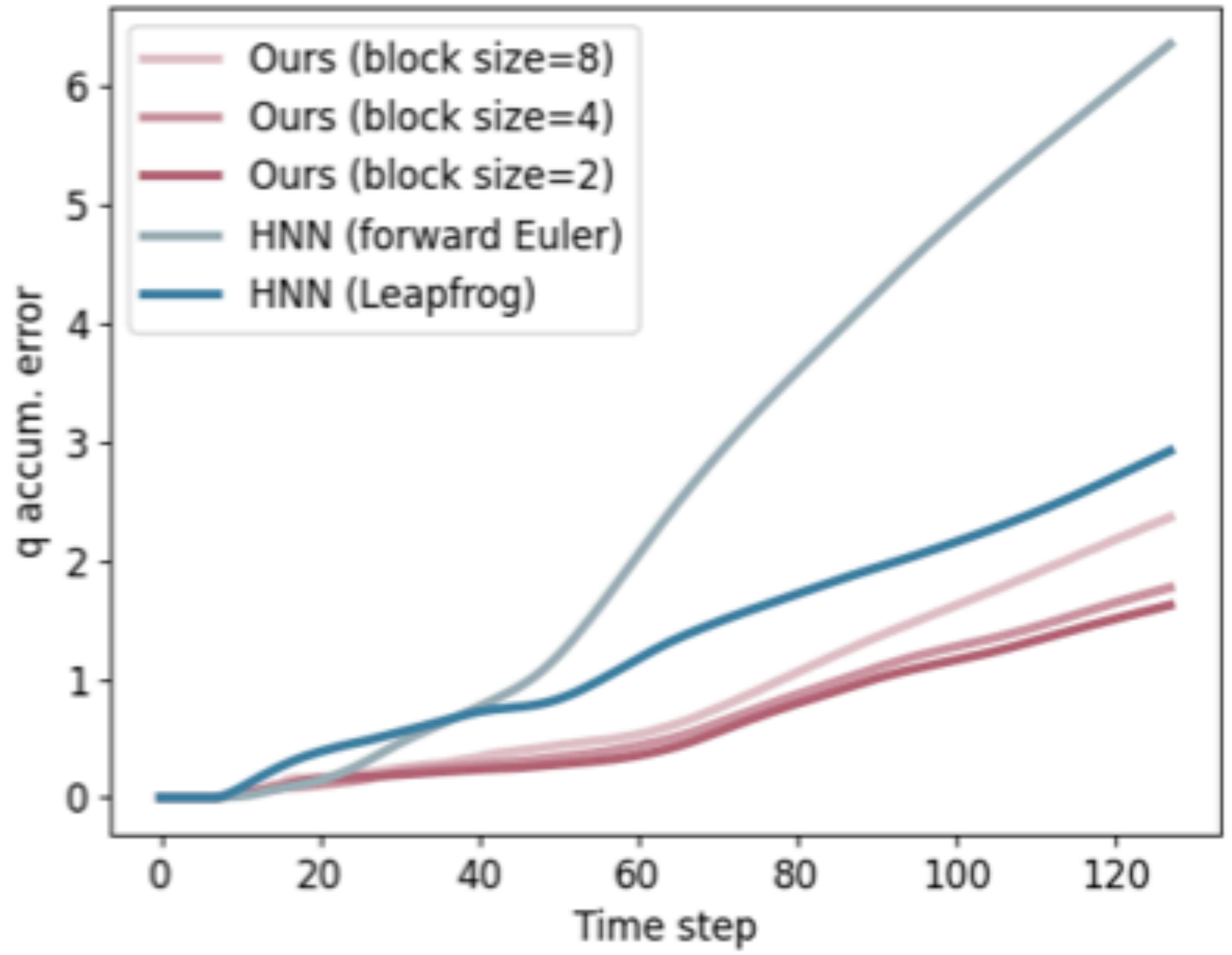
Denoising Hamiltonian Network for Physical Reasoning

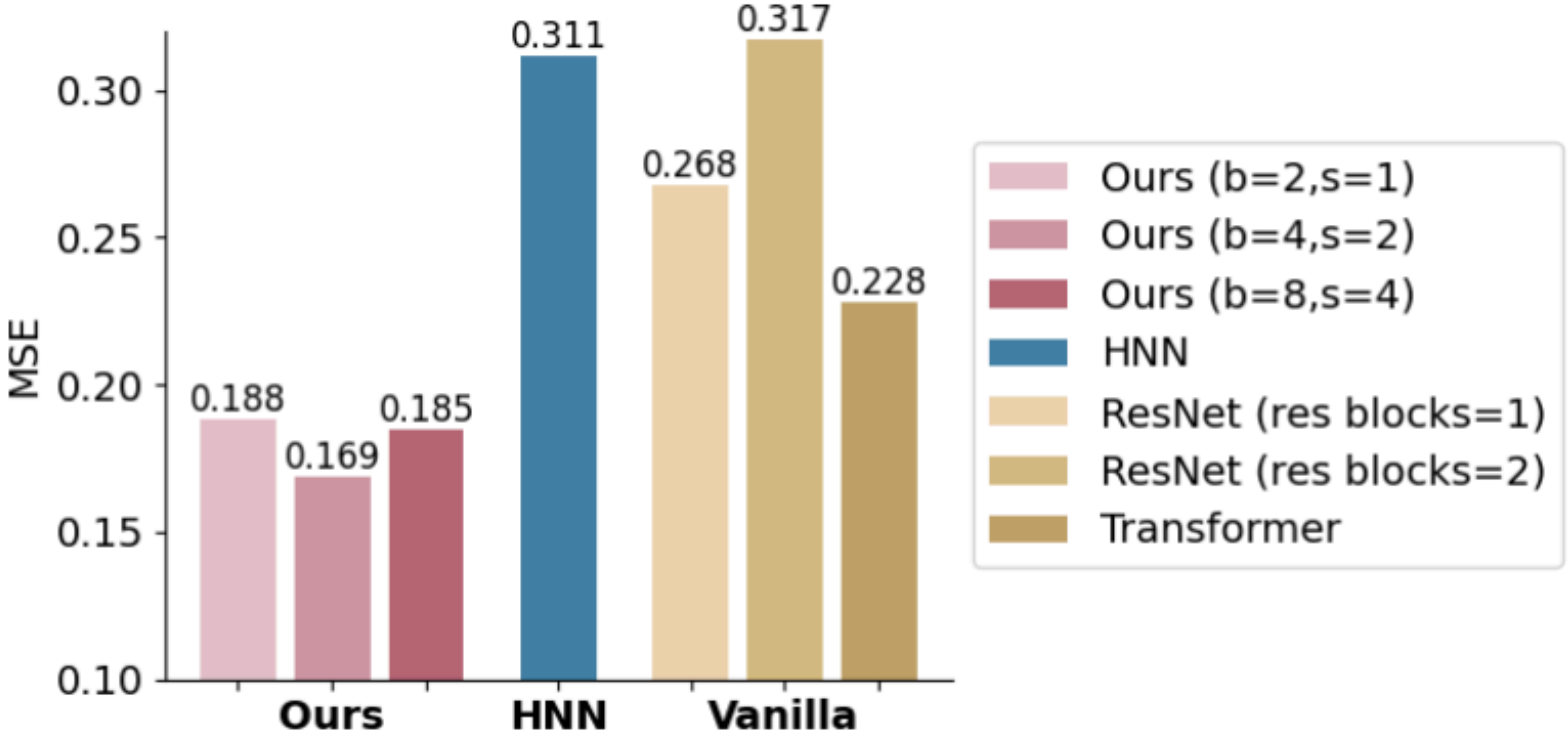


Congyue Deng, Brandon Feng, CG, Alan Garbarz, Bill Freeman, Leo Guibas, Kaiming He, Arxiv: 2503.07596

Denoising Hamiltonian Network for Physical Reasoning

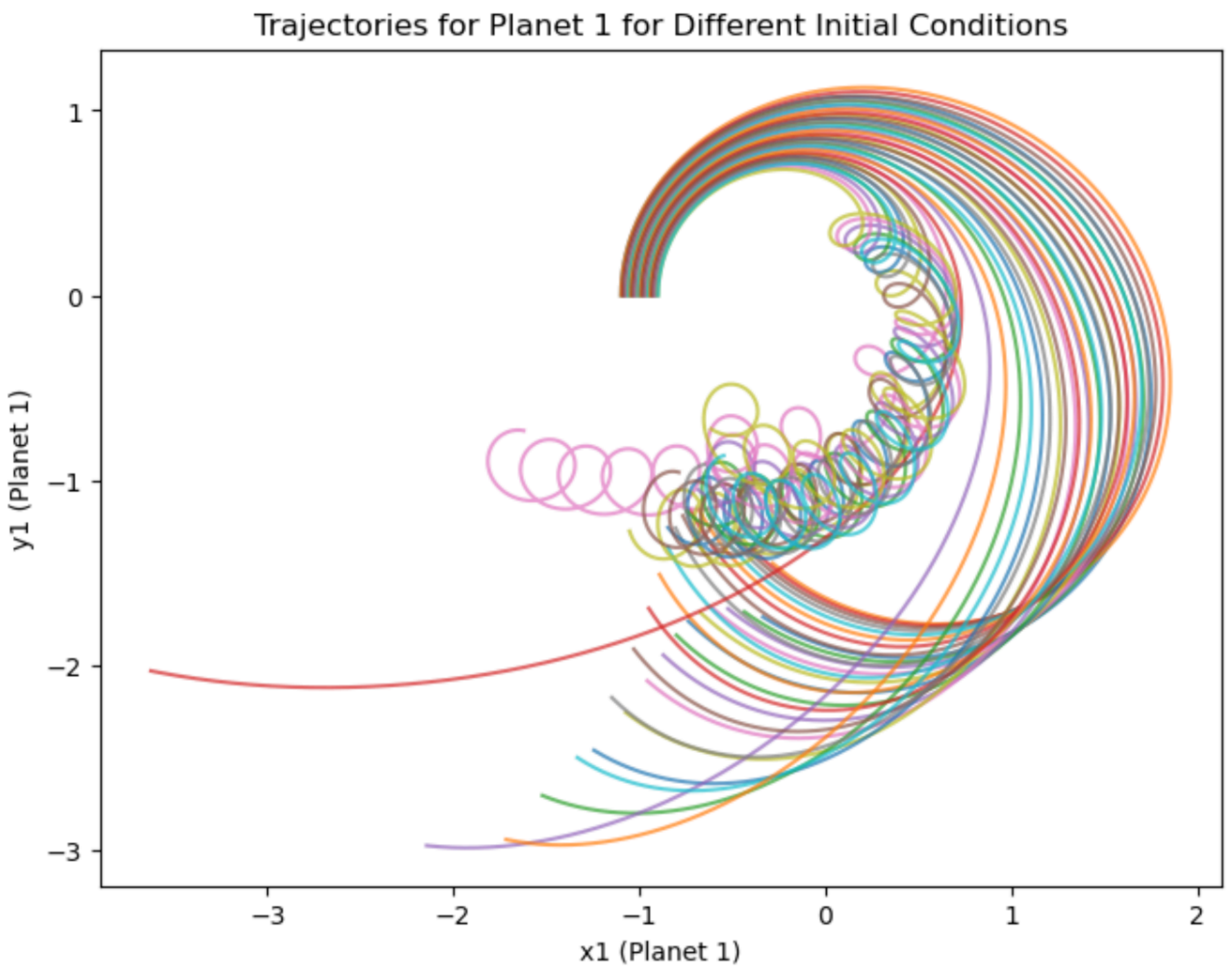
Double Pendulum:

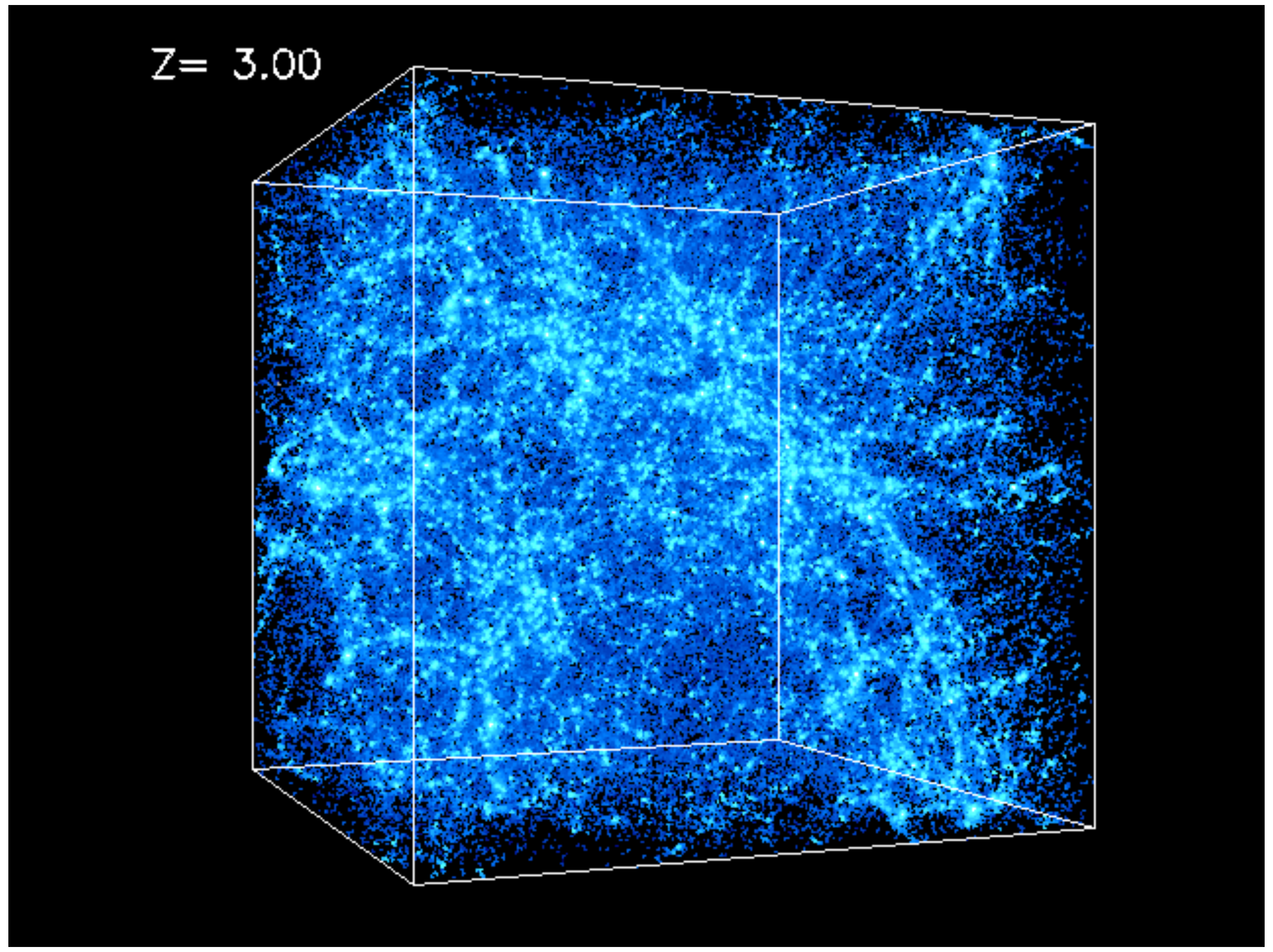




Congyue Deng, Brandon Feng, CG, Alan Garbarz, Bill Freeman, Kaiming He, ArXiv: 2503.07596

Next Up: Three to N-body Problems





Fin de Curso - Gracias Totales!